

# SIEMENS

## SINUMERIK

### SINUMERIK 840D/840Di/810D Grundfunktionen: PLC- Grundprogramm powerline (P3 pl)

Funktionshandbuch

<u>Kurzbeschreibung</u>	1
<u>Ausführliche Beschreibung</u>	2
<u>Randbedingungen</u>	3
<u>Beispiele</u>	4
<u>Datenlisten</u>	5

#### Gültig für

##### *Steuerung*

SINUMERIK 840D powerline/840DE powerline  
SINUMERIK 840Di powerline/840DiE powerline  
SINUMERIK 810D powerline/810DE powerline

##### *Software*

NCU Systemsoftware für 840D/840DE 7.4  
NCU Systemsoftware für 840Di/840DiE 3.3  
NCU Systemsoftware für 810D/810DE 7.4

##### *Version*

11/2006

6FC5397-0BP10-2AA0

## Sicherheitshinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.



---

### Gefahr

bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---



---

### Warnung

bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---



---

### Vorsicht

mit Warndreieck bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---

---

### Vorsicht

ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---

---

### Achtung

bedeutet, dass ein unerwünschtes Ergebnis oder Zustand eintreten kann, wenn der entsprechende Hinweis nicht beachtet wird.

---

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

---

## Qualifiziertes Personal

Das zugehörige Gerät/System darf nur in Verbindung mit dieser Dokumentation eingerichtet und betrieben werden. Inbetriebsetzung und Betrieb eines Gerätes/Systems dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieser Dokumentation sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

## Bestimmungsgemäßer Gebrauch

Beachten Sie Folgendes:



---

### Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden. Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

---

## Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

## Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

# Inhaltsverzeichnis

<b>1</b>	<b>Kurzbeschreibung</b> .....	<b>1-1</b>
<b>2</b>	<b>Ausführliche Beschreibung</b> .....	<b>2-1</b>
2.1	Eckdaten der PLC-CPU's für 810D, 840D und 840Di .....	2-1
2.2	Ressourcen (Timer, FC, FB, DB, Peripherie) reservieren .....	2-8
2.3	Inbetriebnahme Hardwarekonfiguration der PLC-CPU's .....	2-9
2.4	Inbetriebnahme PLC-Programm .....	2-14
2.4.1	Installation des Grundprogramms für 810D, 840D .....	2-14
2.4.2	Anwendung des Grundprogramms .....	2-16
2.4.3	Versionskennzeichnungen .....	2-17
2.4.4	Maschinenprogramm .....	2-17
2.4.5	Datensicherung .....	2-18
2.4.6	PLC Serien Inbetriebnahme, PLC Archive: .....	2-18
2.4.7	Softwarehochrüstung .....	2-20
2.4.8	Peripheriebaugruppen (FM-, CP-Baugruppen).....	2-21
2.4.9	Fehler-Beseitigung .....	2-22
2.5	Ankopplung der PLC-CPU's an 810D, 840D .....	2-23
2.5.1	Allgemein .....	2-23
2.5.2	Eigenschaften der PLC CPU's.....	2-23
2.5.3	Nahtstelle bei 810D und 840D mit integrierter PLC .....	2-24
2.5.4	Diagnosepuffer der PLC .....	2-26
2.6	Struktur der Nahtstelle .....	2-28
2.6.1	Allgemein .....	2-28
2.6.2	Nahtstelle PLC/NCK .....	2-28
2.6.3	Nahtstelle PLC/HMI .....	2-34
2.6.4	Nahtstelle PLC/MSTT/BHG .....	2-37
2.7	Struktur und Funktionen des Grundprogramms .....	2-41
2.7.1	Allgemein .....	2-41
2.7.2	Anlauf und Synchronisation NCK-PLC .....	2-43
2.7.3	Zyklischer Betrieb (OB1).....	2-43
2.7.4	Zeitalarm Bearbeitung (OB 35).....	2-46
2.7.5	Prozessalarm Bearbeitung (OB 40).....	2-46
2.7.6	Verhalten bei NC-Ausfall.....	2-46
2.7.7	Funktionen des Grundprogramms mit Aufruf vom Anwenderprogramm .....	2-48
2.7.8	Symbolische Programmierung des Anwenderprogramms mit Nahtstellen-DB .....	2-51
2.7.9	M-Dekodierung nach Liste .....	2-52
2.7.10	PLC-Maschinendaten .....	2-57
2.7.11	Projektierung von Maschinensteuertafel, Bedienhandgerät .....	2-61
2.8	SPL für Safety Integrated.....	2-69
2.9	Belegungsübersicht .....	2-70
2.9.1	Belegung: NC/PLC-Nahtstelle .....	2-70
2.9.2	Belegung: FB/FC .....	2-70
2.9.3	Belegung: DB .....	2-71

2.9.4	Belegung: Timer.....	2-72
2.10	Speicherbedarf des PLC-Grundprogramms für 840D .....	2-73
2.11	Rahmenbedingungen und NC-VAR-Selektor .....	2-76
2.11.1	Rahmenbedingungen.....	2-76
2.11.1.1	Programmier- und Parametrierwerkzeuge.....	2-76
2.11.1.2	Notwendige SIMATIC-Dokumentation .....	2-78
2.11.1.3	Relevante SINUMERIK-Dokumente .....	2-78
2.11.2	NC-VAR-Selector .....	2-78
2.11.2.1	Übersicht .....	2-78
2.11.2.2	Funktionsbeschreibung.....	2-81
2.11.2.3	Inbetriebnahme, Installation.....	2-90
2.12	Bausteinbeschreibungen.....	2-91
2.12.1	FB 1: RUN_UP Grundprogramm, Anlaufteil.....	2-91
2.12.2	FB 2: GET NC-Variable lesen .....	2-100
2.12.3	FB 3: PUT NC-Variable schreiben .....	2-108
2.12.4	FB 4: PI_SERV Allgemeine PI-Dienste.....	2-116
2.12.4.1	Überblick verfügbarer PI-Dienste.....	2-119
2.12.4.2	Allgemeine PI-Dienste.....	2-120
2.12.4.3	Werkzeugverwaltungsdienste .....	2-125
2.12.5	FB 5: GETGUD GUD-Variable lesen .....	2-141
2.12.6	FB 7: PI_SERV2 Allgemeine PI Dienste.....	2-147
2.12.7	FB 9: M zu N Bedieneinheitenumschaltung.....	2-151
2.12.8	FB 10: Sicherheits-Relais (SI-Relais) .....	2-157
2.12.9	FB 11: Bremsentest .....	2-161
2.12.10	FB 29: Diagnose Signalrekorder und Datentrigger.....	2-167
2.12.11	FC 2: GP_HP Grundprogramm, zyklischer Teil .....	2-171
2.12.12	FC 3: GP_PRAL Grundprogramm, alarmgesteuerter Teil .....	2-173
2.12.13	FC 7: TM_REV Transfer-Baustein für Werkzeugwechsel mit Revolver .....	2-176
2.12.14	FC 8: TM_TRANS Transfer-Baustein für Werkzeugverwaltung .....	2-180
2.12.15	FC 9: ASUP Start von asynchronen Unterprogrammen .....	2-188
2.12.16	FC 10: AL_MSG Fehler- und Betriebsmeldungen .....	2-191
2.12.17	FC 12: AUXFU Aufrufchnittstelle für Anwender bei Hilfsfunktionen .....	2-193
2.12.18	FC 13: BHGDisp Displaysteuerung für Bedienhandgerät.....	2-194
2.12.19	FC 15: POS_AX Positionierung von Linear- und Rundachsen.....	2-198
2.12.20	FC 16: PART_AX Positionierung von Teilungsachsen .....	2-202
2.12.21	FC 17: YDelta Stern-/Dreieck Umschaltung .....	2-206
2.12.22	FC 18: SpinCtrl Spindelsteuerung .....	2-210
2.12.23	FC 19: MCP_IFM Übertragung der MSTT-Signale an die Nahtstelle.....	2-221
2.12.24	FC 21: Transfer Datenaustausch PLC-NCK.....	2-229
2.12.25	FC 22: TM_DIR Richtungsauswahl für Werkzeugverwaltung.....	2-238
2.12.26	FC 24: MCP_IFM2 Übertragung der MSTT-Signale an die Nahtstelle.....	2-241
2.12.27	FC 25: MCP_IFT Übertragung der MSTT-/BT-Signale an die Nahtstelle.....	2-245
2.12.28	FC 26: HPU_MCP Übertragung der PHG / HT6-Signale an die Nahtstelle.....	2-248
2.12.28.1	FC 26: HPU_MCP Übertragung der PHG / HT6-Signale an die Nahtstelle.....	2-248
2.12.28.2	Anwahlsignale der MSTT auf die Anwender-Nahtstelle .....	2-251
2.12.28.3	Rückmeldungen der Anwender-Nahtstelle zur Ansteuerung von Anzeigen.....	2-253
2.12.29	FC 19, FC 24, FC 25, FC 26 Beschreibung Quellcode .....	2-254
2.13	Signal-/Datenbeschreibungen.....	2-256
2.13.1	Nahtstellensignale NCK/PLC, MMC/PLC, MSTT/PLC .....	2-256
2.13.2	Dekodierte M-Signale.....	2-256
2.13.3	G-Funktionen .....	2-258
2.13.4	Meldesignale im DB2 .....	2-260
2.14	Programmiertipps mit STEP 7.....	2-263

---

2.14.1	Allgemeines.....	2-263
2.14.2	Daten kopieren.....	2-263
2.14.3	ANY und POINTER.....	2-264
2.14.3.1	Variable POINTER bzw. ANY für Übergabe an FC oder FB.....	2-264
2.14.3.2	Allgemeines.....	2-266
2.14.3.3	Verwendung von POINTER und ANY im FC, wenn POINTER bzw. ANY als Parameter vorliegt.....	2-266
2.14.3.4	Verwendung von POINTER und ANY im FB, wenn POINTER bzw. ANY als Parameter vorliegt.....	2-268
2.14.4	Multi-Instanz DB.....	2-269
2.14.5	Strings.....	2-271
2.14.6	Ermittlung von Offsetadressen auf Datenbaustein-Strukturen.....	2-272
<b>3</b>	<b>Randbedingungen .....</b>	<b>3-1</b>
<b>4</b>	<b>Beispiele .....</b>	<b>4-1</b>
<b>5</b>	<b>Datenlisten.....</b>	<b>5-1</b>
5.1	Maschinendaten.....	5-1
5.1.1	NC-spezifische Maschinendaten .....	5-1
5.1.2	Kanal-spezifische Maschinendaten .....	5-1
	<b>Index.....</b>	<b>Index-1</b>



# Kurzbeschreibung

## Allgemeines

Das PLC-Grundprogramm organisiert den Austausch von Signalen und Daten zwischen dem PLC-Anwenderprogramm und dem NCK-, HMI- und MSTT-Bereich. Bei den Signalen und Daten ist dabei zwischen folgenden Gruppen zu unterscheiden:

- Zyklischer Signalaustausch
- Ereignisgesteuerter Signalaustausch
- Meldungen

## Zyklischer Signalaustausch

Die Signale, die zyklisch ausgetauscht werden, bestehen im Wesentlichen aus Bitfeldern.

- Sie enthalten **Befehle**, die von der PLC an den NCK übergeben werden (z.B. Start, Stopp etc.) und **Status-Informationen** des NCK (z.B. Programm läuft, Programm unterbrochen etc.).
- Die Bitfelder sind gegliedert in Signale für:
  - BAGs
  - Kanäle
  - Achsen/Spindeln
  - allgemeine NCK-Signale

Der zyklische Signalaustausch wird vom Grundprogramm am Zyklusanfang der PLC (OB1) durchgeführt. Dadurch ist sichergestellt, dass z.B. die Signale vom NCK über einen Zyklus konstant bleiben.

## Ereignisgesteuerter Signalaustausch NCK → PLC

PLC-Funktionen, die abhängig vom Werkstückprogramm ausgeführt werden müssen, werden über Hilfsfunktionen im Werkstückprogramm angestoßen. Gelangt ein Satz mit Hilfsfunktionen zur Ausführung, so hängt es von der Art der Hilfsfunktion ab, ob der NCK die Ausführung dieser Funktion abwarten muss (z.B. Werkzeugwechsel) oder ob diese Funktion begleitend zur Werkstückbearbeitung zur Ausführung kommt (z.B. Werkzeugbereitstellung bei Fräsmaschinen mit Kettenmagazinen).

Damit die NC-Bearbeitung möglichst wenig beeinflusst wird, muss die Datenübergabe möglichst schnell und trotzdem sicher erfolgen. Sie erfolgt deshalb alarm- und quittungsgesteuert. Das Grundprogramm wertet die Signale und Daten aus, quittiert dies zum NCK und überträgt die Daten am Zyklusanfang zur Anwendernahtstelle. Erfordern die Daten keine Anwenderquittung, so wird die NC-Bearbeitung dadurch nicht beeinflusst.

### **Ereignisgesteuerter Signalaustausch PLC → NCK**

Immer dann, wenn die PLC an den NCK einen Auftrag übergibt (z.B. Verfahren einer Hilfsachse), findet ein "ereignisgesteuerter Signalaustausch PLC → NCK" statt. Auch hier erfolgt die Datenübergabe quittungsgesteuert. Vom Anwenderprogramm aus wird ein derartiger Signalaustausch über einen FB bzw. FC angestoßen.

Die zugehörigen FBs (Funktionsbausteine) und FCs (Function Calls) werden zusammen mit dem Grundprogramm ausgeliefert.

### **Meldungen**

Die Erfassung und Aufbereitung der Anwendermeldungen erfolgt durch das Grundprogramm. Über ein vereinbartes Bitfeld werden die Meldesignale an das Grundprogramm übergeben. Dort werden diese Signale ausgewertet und bei Auftreten der Meldeereignisse in den PLC-Diagnosepuffer eingetragen. Ist eine Bedieneinheit vorhanden, werden die Meldungen auf der Bedienoberfläche angezeigt.

---

#### **Hinweis**

Die Funktion der Maschine wird maßgebend vom PLC-Programm bestimmt. Jedes im Arbeitsspeicher vorhandene PLC-Anwender-Programm kann mit dem Programmiergerät geändert werden.

---



## Ausführliche Beschreibung

### 2.1 Eckdaten der PLC-CPU's für 810D, 840D und 840Di

Die nachfolgenden Tabellen zeigen den Leistungsumfang der PLC-CPU's und den Umfang des PLC-Grundprogramms bezogen auf die verschiedenen Steuerungstypen.

#### Steuerungstyp: 810D bzw. 840D

##### CPU-Eckdaten

	810D / 840D	810D / 840D	810D / 840D
PLC-CPU	integrierte PLC CPU314	integrierte PLC CPU315-2DP	integrierte PLC CPU315-2DP Master/Slave
MLFB		6ES7 315-2AF00-0AB0	6ES7 315-2AF01-0AB0
Speicher für Anwender- und Grundprogramm	64, 96, 128 kByte	64, 96, 128, 160, 192, 224, 256, 288 kByte (Optionsabhängig)	96, 160, 224, 288, 352, 416, 480 kByte (Optionsabhängig)
Datenbaustein-Speicher	Wie Anwenderspeicher	Wie Anwenderspeicher	Bis 96 kByte
Speichermodul	nein	nein	nein
Merker (Bits)	2048	2048	2048/ 4096 ab PLC Betriebssystem 03.10.13
Zeiten	128	128	128
Zähler	64	64	64
Taktmerker	8	8	8
Programm-/Datenbausteine			
OB	1, 10, 20, 35, 40, 80-82, 85, 87, 100,	1, 10, 20, 35, 40, 80-82, 85-87, 100,	1, 10, 20, 35, 40, 80-82, 85-87, 100,
FB	121-122	121-122	121-122
FC	1-127	1-127	0-255
DB	1-127	1-127	0-255
	1-127	1-127	1-399
max. Datenbausteinlänge	16 kByte	16 kByte	16 kByte
max. Bausteinlänge FC, FB	16 kByte	24 kByte	24 kByte
Eingänge/Ausgänge (Adressiervolumen)			
- digital	768	1024/1024	1024/1024
- analog	64	64	64

## Ausführliche Beschreibung

### 2.1 Eckdaten der PLC-CPU's für 810D, 840D und 840Di

	810D / 840D	810D / 840D	810D / 840D
Eingänge/Ausgänge 1) (Adressierung)	Zeile 0 ist für die Peripherie nicht verfügbar:	durch freie Projektierung der Peripherie:	durch freie Projektierung der Peripherie:
- digital	ab E/A-Byte 32	ab E/A-Byte 0	ab E/A-Byte 0
- analog	ab PE/PA-Byte 384	ab PE/PA-Byte 272	ab PE/PA-Byte 272
Bearbeitungszeit			
- Bitbefehle (E/A)	0,3 ms/kA	0,3 ms/kA	0,3 ms/kA
- Wortbefehle	1-4 ms/kA	1-4 ms/kA	1-4 ms/kA
PDIAG (Alarm S,SQ)	nein	nein	ja
Profibus	entfällt	Master	Master/Slave
Anzahl Profibus Slaves		Min. 16, max. 64 SDB 2000 ≤ 8 kByte	Min. 16, max. 64 SDB 2000 ≤ 32 kByte
programmierbare Baustein- kommunikation PBK	nein	nein	ja
Konsist. Daten an Norm- slave über SFC 14, 15	entfällt	26	26
1) Zeile 0 ist in der NC integriert. Für Peripherie stehen Zeile 1 bis 3 zur Verfügung			

### Peripherie-Ausbau

	810D / 840D	810D / 840D	810D / 840D
PLC-CPU	integrierte PLC CPU314	integrierte PLC CPU315-2DP	integrierte PLC CPU315-2DP Master/Slave
MLFB		6ES7 315-2AF00-0AB0	6ES7 315-2AF01-0AB0
E-/A-Module	24	24	24
Profibus DP-Module	entfällt	ja	ja
Schnittstellen (MPI)	1	1	1

### Steuerungstypen: 840Di, 810D und 840D

#### CPU-Eckdaten

	840Di	810D	840D
PLC-CPU	integrierte PLC 315-2DP Master / Slave	integrierte PLC 315-2DP Master / Slave	integrierte PLC 314C-2DP Master / Slave
MLFB	6ES7 315-2AF03-0AB0	6ES7 315-2AF03-0AB0	6FC5 314-6CF00-0AB0
Speicher für Anwender- und Grundprogramm	64, 96, 128, 160, 192, 224, 256, kByte	64, 96, 128, 160, 192, 224, 288 kByte	96, 160, 224, 352, 416, 480 kByte (Options-abhängig)
Datenbaustein-Speicher	Wie Anwenderspeicher	Wie Anwenderspeicher	Bis 96 kByte
Speichermodul	nein	nein	nein
Merker (Bits)	4096	4096	4096
Zeiten	128	128	256
Zähler	64	64	256
Taktmerker	8	8	8

	840Di	810D	840D
Programm/Datenbausteine			
OB	1, 10, 20, 35, 40, 80-82, 85-87, 100, 121-122	1, 10, 20, 35, 40, 80-82, 85-87, 100, 121-122	1, 10, 20, 35, 40, 80-82, 85-87, 100, 121-122
FB	0-255	0-255	0-255
FC	0-255	0-255	0-255
DB	1-399	1-399	1-399
max. Daten-Bausteinlänge	16 kByte	16 kByte	16 kByte
max Bausteinlänge FC, FB	24 kByte	24 kByte	24 kByte
Eingänge/Ausgänge (Adressiervolumen)			
- digital	1024/1024	1024/1024	1024/1024
- analog	64	64	64
Eingänge/Ausgänge 1) (Adressierung)	durch freie Projektierung der Peripherie:	durch freie Projektierung der Peripherie:	durch freie Projektierung der Peripherie:
- digital	ab E/A-Byte 0	ab E/A-Byte 0	ab E/A-Byte 0
- analog	ab PE/PA-Byte 272 (nur Profibus)	ab PE/PA-Byte 272	ab PE/PA-Byte 272
Bearbeitungszeit			
- Bitbefehle (E/A)	0,3 ms/kA	0,3 ms/kA	0,1 ms/kA
- Wortbefehle	1-4 ms/kA	1-4 ms/kA	0,25-1,2 ms/kA
PDIAG (Alarm S, SQ)	Ja	Ja	Ja
Profibus	Master	Master / Slave	Master / Slave
Anzahl Profibus Slaves	Max. 64 SDB 2000 ≤ 32 kByte	Max. 64 SDB 2000 ≤ 32 kByte	Max. 32 SDB 2000 ≤ 32 kByte
Max. Anzahl Profibus-Slots	256	256	256
programmierbare Baustein- kommunikation PBK	Ja	Ja	Ja
Konsist. Daten an Norm- slave über SFC 14, 15	26	26	32

1) Zeile 0 ist in der NC integriert. Für Peripherie stehen Zeile 1 bis 3 zur Verfügung

### Peripherie-Ausbau

	840Di	810D	840D
PLC-CPU	integrierte PLC 315-2DP Master / Slave	integrierte PLC 315-2DP Master / Slave	integrierte PLC 314C-2DP Master / Slave
MLFB	6ES7 315-2AF03-0AB0	6ES7 315-2AF03-0AB0	6FC5 314-6CF00-0AB0
E-/A-Module	Nur Profibus	24	24
Profibus DP-Module	Ja	Ja	Ja
Schnittstellen (MPI)	1	1	1

**Steuerungstypen: 840Di und 840D**

**CPU-Eckdaten**

	<b>840Di</b>	<b>840D</b>
PLC-CPU	integrierte PLC 317-2DP Master / Slave	integrierte PLC 317-2DP Master / Slave
MLFB	6FC5 317-2AJ10-0AB0	6FC5 317-2AJ10-1AB0
Speicher für Anwender- und Grundprogramm	128 bis 768 kByte	128 bis 768 kByte
Datenbaustein-Speicher	max. 256 kByte	max. 256 kByte
Speichermodul	nein	nein
Merker (Bits)	32768	32768
Zeiten	512	512
Zähler	512	512
Taktmerker	8	8
Programm-/ Datenbausteine:		
OB	10, 20-21, 32-35, 40, 55-57, 80, 82, 85-87, 100,	10, 20-21, 32-35, 40, 55-57, 80, 82, 85-87, 100,
FB	121-122	121-122
FC	0-2048	0-2048
DB	0-2048 1-2048	0-2048 1-2048
max. Daten-Bausteinlänge	32 kByte	32 kByte
max Bausteinlänge FC, FB	64 kByte	64 kByte
Eingänge/Ausgänge 1) (Adressiervolumen in Byte):		
- digital / - analog		
- Inkl. reservierter Bereich	4096/4096	4096/4096
- Prozessabbild	8192/8192	8192/8192
Achtung: Die Ein-/Ausgänge oberhalb 4096 sind für integrierte Antriebe reserviert.	256/256	256/256
Eingänge/Ausgänge 2) (Adressierung):	durch freie Projektierung der Peripherie:	durch freie Projektierung der Peripherie:
- digital	ab E/A-Byte 0	ab E/A-Byte 0
- analog	ab PE/PA-Byte 272 (nur Profibus)	ab PE/PA-Byte 272
Bearbeitungszeit:		
- Bitbefehle (E/A)	≤ 0,03 ms/kA	≤ 0,03 ms/kA
- Wortbefehle	0,1 ms/kA	0,1 ms/kA
PDIAG (Alarm S, SQ)	Ja	Ja
Profibus	Master / Slave	Master / Slave
Anzahl Profibus Slaves	max. 125	max. 125
Max. Anzahl Profibus-Slots	512	512
DP-Mastersystem Nr. DP	1	1

	840Di	840D
DP-Mastersystem Nr. MPI/DP	2	entfällt
programmierbare Baustein-kommunikation PBK	Ja	Ja
Konsist. Daten an Norm-slave über SFC 14, 15	128	128
1) Achtung: Die Ein-/Ausgänge oberhalb 4096 sind für integrierte Antriebe reserviert.		
2) Zeile 0 ist in der NC integriert. Für Peripherie stehen Zeile 1 bis 3 zur Verfügung		

### Peripherie-Ausbau

	840Di	840D
PLC-CPU	integrierte PLC 317-2DP Master / Slave	integrierte PLC 317-2DP Master / Slave
MLFB	6FC5 317-2AJ10-0AB0	6FC5 317-2AJ10-1AB0
E-/A-Module	Nur Profibus	24
Profibus DP-Module	1 (2)	1
Schnittstellen (MPI)	1 (0)	1

### Hinweis

#### Anzahl Profibus Slaves

Da der SDB 2000 und weitere SDBs durch das PLC-Betriebssystem im statischen RAM-Bereich, auf den auch der Profibus ASIC zugreifen kann, abgelegt werden muss, können weiterhin die Information des SDB2000 auch dem NCK und dem PLC Grundprogramm in aufbereiteter Form (CPI- Interface) übergeben werden.

Dies ist notwendig für die Ansteuerung von Antrieben und PROFIsafe Baugruppen auf dem Profibus. Für diese Datenstrukturen steht ein durch die PLC festgelegter Speicherbereich zur Verfügung. Die Begrenzung ist durch die maximale Slotanzahl gekennzeichnet. Dadurch können beim Laden auch SDB's abgelehnt werden, die eine kleinere Anzahl von Slaves beinhalten gegenüber den oben aufgeführten Zahlen. Ein Slot ist im Regelfalle in Modul des Slaves oder der Slave selbst. Nur bei einem Modul, welches gleichzeitig E und A Bereiche hat, zählt ein Modul als 2 Slots.

Eine exakte Angabe der Grösse des SDB 2000 ist deshalb nicht möglich.

Erst nach dem Laden des SDB-Containers in die CPU kann ausgesagt werden, ob die Konfiguration zulässig ist. Die in den vorher genannten Tabellen angegebenen Werte sind daher nur als Richtwerte zu verstehen.

Ist die Konfiguration unzulässig, wird beim Laden der SDBs eine Umlöschanforderung gesetzt. Nach dem Umlöschen kann im Diagnosepuffer die Ursache festgestellt werden.

### PLC Versionen

Ab Softwarestand 3.5 bei 840D sind bei der PLC 314 der Versionsstand 6 (Versions-Kennzeichnung 35.06.03) und bei der PLC 315-2DP der Versionsstand 3 (Versions-Kennzeichnung 35.03.03) bzw. höherer Versionsstand eingebaut.

Diese Versionsstände sind kompatibel mit den entsprechenden SIMATIC CPU300. Somit sind alle Baugruppen und Softwarepakete einsetzbar, die von SIMATIC für diese Versionsstände und diese CPUs freigegeben sind. Eine Ausnahme bilden Baugruppen, die nur in Zeile 0 gesteckt werden dürfen (weitere Ausnahmen sind die Baugruppen FM-NC und FM 357).

Versions-Kennzeichnung: XX.YY.ZZ

- XX: PLC-Versionsstand der SIMATIC-CPU
- YY: Hochzählung der Firmwareübernahme
- ZZ: Interne Hochzählung

#### Beispiel

PLC 315-2DP mit MLFB 6ES7 315-2AF00-0AB0:	04.02.14
PLC 315-2DP mit MLFB 6ES7 315-2AF01-0AB0:	03.10.23
PLC 314:	07.02.12

### HMI-Versionsbild

In der letzten Zeile des HMI-Versionsbildes wird das PLC-Modul, die Version des PLC-Betriebssystems und die Baugruppenkennung angezeigt.

#### Beispiel

PLC-Modul	PLC-Betriebssystems-Version	Baugruppenkennung
S7 PLC_315-2DP System	03.10.23	1200

### Baugruppenkennungen

Die folgende Tabelle zeigt den Zusammenhang zwischen Baugruppenkennung und dem entsprechenden PLC-Modul, dem einsetzbaren PLC-Betriebssystem und dessen aktuellen SW-Stand:

Baugruppenkennung	PLC Modul	einsetzbare PLC Betriebssysteme (entsprechende SIMATIC-MLFB)	SW-Stand PLC-Betriebssystem
0208	PLC 314	6ES7 314-1AE0-0AB0	07.02.12
1008	PLC 315-2DP mit ASPC 2 Step C	6ES7 315-2AF00-0AB0	04.02.14
1100	PLC 315-2DP mit ASPC 2 Step D	6ES7 315-2AF01-0AB0	03.10.23
1200	PLC 315-2DP mit ASPC 2 Step E	6ES7 315-2AF01-0AB0	03.10.23
		6ES7 315-2AF03-0AB0 FW1.2	12.30.10
1400	PLC 314C-2DP mit IBC 16	6ES7 314-6CF00-0AB0 FW1.0.2	10.60.20
2200	PLC 317-2DP mit IBC 32	6ES7 317-2AJ10-0AB0 FW2.1	20.71.15
MCI 1 (840Di)	PLC 315-2DP mit ASPC 2 Step E	6ES7 315-2AF03-0AB0 FW1.0	04.20.36

Baugruppenkennung	PLC Modul	einsetzbare PLC Betriebssysteme (entsprechende SIMATIC-MLFB)	SW-Stand PLC-Betriebssystem
MCI 2 (840Di) 2100	PLC 317-2DP mit IBC32	6ES7 317-2AJ10-0AB0 FW2.1	20.70.17

## 810 D, 840D

Die folgenden Tabellen zeigen die Eckdaten bezüglich BTSS-Schnittstelle und PLC-Grundprogramm-Funktionalität für SINUMERIK 810D, 840D und 840Di:

### BTSS-Schnittstelle

	840Di	810D	840D
Anzahl	entfällt	entfällt	1

### PLC-Grundprogramm-Funktionen

	840Di	810D	840D
Achsen / Spindeln	1)	5	31
Kanäle	1)	2	10
BAGs	1)	1	10
Status- / Steuersignale	+	+	+
M-Decoder (M00-99)	+	+	+
G-Gruppen-Decoder	+	+	+
Hilfsfunktionsverteiler	+	+	+
Alarmgesteuert Ausgabe der Hilfsfunktionen	+	+	+
Achsen / Spindeln von PLC aus bewegen	+	+	+
ASUP-Schnittstelle	-	-	-
Fehler- / Betriebsmeldungen	+	+	+
MSTT- und BHG-Signale über NCK	+	+	+
Lesen / Schreiben von NC-Variablen	+	+	+
PI-Dienste	+	+	+
Werkzeugverwaltung	+	+	+
Stern / Dreieck Umschaltung	+	+	+
Display-Steuerung Bedienhandgerät	+	+	+

1) Abhängig vom gewählten System-Softwarepaket

## 2.2 Ressourcen (Timer, FC, FB, DB, Peripherie) reservieren

### Reservierte Komponenten

Folgende Komponenten sind für das Grundprogramm reserviert:

Komponente	Reservierter Bereich
Timer	T0 - T9
Funktionen (allgemein)	FC 0 - FC 29
Funktionen (bei ShopMill/ShopTurn)	FC 0 - FC 35
Funktionsbausteine	FB 0 - FB 29
Datenbausteinen (allgemein) <sup>1)</sup>	DB 1 - DB 62; DB 71 - DB 80
Datenbausteinen (bei ShopMill/ShopTurn) <sup>1)</sup>	DB 1 - DB 62; DB 71 - DB 89

1) Datenbausteine von nicht aktivierten Kanälen, Achsen/Spindeln, Werkzeugverwaltung sind frei für den Anwender.

### PLC 317-2DP

Bezüglich der PLC-CPU: PLC 317-2DP sind für SIEMENS-Anwendungen bezüglich FC, FB, DB und Peripheriebereiche weitere Nummernbänder reserviert.

#### FC, FB und DB

Komponente	Reservierter Bereich
Funktionen	FC 1000 - FC 1023
Funktionsbausteine	FB 1000 - FB 1023
Datenbausteinen	DB 1000 - DB 1099

#### Peripheriebereich

Komponente	Reservierter Bereich
Adressbereich	256 - 271 <sup>1)</sup>
Ein- / Ausgänge	ab 4096 <sup>2)</sup>

1) Reserviert für die NC-Baugruppe und zukünftige Erweiterungen

2) Reserviert für integrierte Antriebe. Diagnoseadressen für Baugruppen können aber , so wie es von STEP7 auch vorgeschlagen wird, im obersten Adressbereich platziert werden.



## 2.3 Inbetriebnahme Hardwarekonfiguration der PLC-CPU's

### Allgemeines Vorgehen

Für eine PLC-CPU ist, inklusive der zugehörigen Peripherie, mittels STEP 7 die Hardwarekonfiguration zu definieren.

Mit folgendem Vorgehen soll dieser Vorgang dargestellt werden:

1. Tool-Box auf PG/PC laden
2. Neues Projekt anlegen (Datei, neu, Projekt)
3. Einfügen, Hardware, SIMATIC 300 Station
4. SIMATIC 300-Station1 mit Maus anwählen
5. Rechte Maustaste, Objekt öffnen, jetzt wird das HWKonfig gestartet
6. Zielsystem, laden in PG, jetzt wird die Hardwarebestückung aus dem zentralen System zurückgelesen
7. Dezentrale Peripherie projektieren
8. PLC-Grundprogramm einfügen

Die Adressen für die Peripheriebaugruppen können bei Bedarf verändert werden (nur bei bestimmten PLC-CPU's zulässig, z. B. PLC 315-2DP).

Alternativ kann auch eine Handeingabe der gesamten Hardwareprojektierung vorgenommen werden (siehe auch entsprechende Dokumentation von STEP 7). Die unten stehenden Hinweise sind zu beachten.

### STEP7, Version 3

Ab STEP7 Version 3 muss die Hardwarekonfiguration der SINUMERIK Komponenten über die Einträge in SIMATIC\RACK 300 vorgenommen werden. Voraussetzung hierfür ist die Anwendung des Install- bzw. Setup-Programms des Grundprogramms auf den Toolbox-Disketten.

### STEP7 Version 5.1 SP2 und Toolbox 6.03.02

Ab STEP7 Version 5.1 SP2 und Toolbox 6.03.02 sind die SINUMERIK Komponenten unter SIMATIC 300\SINUMERIK zu finden. Die aktuelle Hardware-Ergänzung für STEP 7 ist auch unter eSupport zu finden.

Aktueller Pfad (13.02.2004): sinumerik\_software > 840d/810d/fm-nc > patches & fixes > plc > Hardware\_fuer\_STEP7 > Hardware upto NCU\*.5/CCU3/840Di\_MCI2 > V6.5.2.0

NCU	MLFB	Enthaltene vergleichbare SIMATIC CPU-MLFB	Auswahl aus STEP7 Hardware Katalog
CCU1 810D-CPU	6FC5 410-0AA00-0AA0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
CCU2 810D-CPU	6FC5 410-0AA01-0AA0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
CCU1 810DE-CPU	6FC5 410-0AY01-0AA0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
CCU2 810D-CPU	6FC5 410-0AX02-0AA0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314

NCU	MLFB	Enthaltene vergleichbare SIMATIC CPU-MLFB	Auswahl aus STEP7 Hardware Katalog
SINUMERIK 810DE-Light CCU1-Baugruppe mit Systemsoftware (Export)	6FC5 410-0AY00-0AA0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 810D CCU2 Baugruppe mit Systemsoftware (Standard)	6FC5 410-0AX02-1AA0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840DE NCU 561.2 ohne Systemsoftware	6FC5 356-0BB11-0AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 571 (Exportversion)	6FC5 357-0BA10-0AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 571 (Exportversion) mit Profibus-DP	6FC5 357-0BA11-0AE0	6ES7 315-2AF00-0AB0	840D mit PLC315-2AF00
SINUMERIK 840D NCU 571.2 (Exportversion) mit Profibus DP	6FC5 357-0BA11-1AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840DE NCU 571.2 ohne Systemsoftware	6FC5 357-0BB11-0AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 572	6FC5 357-0BA20-0AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 572	6FC5 357-0BA20-1AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 572	6FC5 357-0BA21-0AE0	6ES7 315-2AF00-0AB0	840D mit PLC315-2AF00
SINUMERIK 840D NCU 572	6FC5 357-0BA21-1AE0	6ES7 315-2AF00-0AB0	840D mit PLC315-2AF00
SINUMERIK 840D NCU 572.2 mit Profibus DP	6FC5 357-0BA21-1AE1	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D/DE NCU 572.2 ohne Systemsoftware	6FC5 357-0BB21-0AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D/DE NCU 572.3 ohne Systemsoftware	6FC5 357-0BB22-0AE0	6ES7 315-2AF01-0AB0	10D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 572 (Exportversion)	6FC5 357-0BY20-0AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 572 (Exportversion)	6FC5 357-0BY20-1AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 572 (Exportversion) mit Profibus DP	6FC5 357-0BY21-0AE0	6ES7 315-2AF00-0AB0	840D mit PLC315-2AF00
SINUMERIK 840D NCU 572 (Exportversion) mit Profibus DP	6FC5 357-0BY21-1AE0	6ES7 315-2AF00-0AB0	840D mit PLC315-2AF00
SINUMERIK 840D NCU 572.2 (Exportversion) mit Profibus DP	6FC5 357-0BY21-1AE1	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 572 mit Digitalisieren	6FC5 357-0BA24-0AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 572.2 mit Digitalisieren und Profibus DP	6FC5 357-0BA24-1AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D/DE NCU 572.2 ohne Systemsoftware	6FC5 357-0BB24-0AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01

2.3 Inbetriebnahme Hardwarekonfiguration der PLC-CPU's

NCU	MLFB	Enthaltene vergleichbare SIMATIC CPU-MLFB	Auswahl aus STEP7 Hardware Katalog
SINUMERIK 840D NCU 572 (Exportversion) mit Digitalisieren	6FC5 357-0BY24-0AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 572.2 (Exportversion) mit Digitalisieren und Profibus DP	6FC5 357-0BY24-1AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 573	6FC5 357-0BA30-0AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840DE NCU 573 (Exportversion)	6FC5 357-0BY30-0AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 573 mit Digitalisieren	6FC5 357-0BA31-0AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 573 (Exportversion) mit Digitalisieren	6FC5 357-0BY31-0AE0	6ES7 314-1AE01-0AB0	810D/840D mit PLC314
SINUMERIK 840D NCU 573 mit Profibus-DP	6FC5 357-0BA32-0AE1	6ES7 315-2AF00-0AB0	840D mit PLC315-2AF00
SINUMERIK 840D NCU 573 (Exportversion) mit Profibus-DP	6FC5 357-0BY32-0AE1	6ES7 315-2AF00-0AB0	840D mit PLC315-2AF00
SINUMERIK 840D NCU 573 mit Profibus-DP	6FC5 357-0BA33-0AE0	6ES7 315-2AF00-0AB0	840D mit PLC315-2AF00
SINUMERIK 840D NCU 573 (Exportversion) mit Profibus-DP	6FC5 357-0BY33-0AE0	6ES7 315-2AF00-0AB0	840D mit PLC315-2AF00
SINUMERIK 840D NCU 573.2 (Pentium Pro) bis 12 Achsen mit Profibus DP	6FC5 357-0BA32-1AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 573.2 (Pentium Pro) bis 31 Achsen mit Profibus DP	6FC5 357-0BA33-1AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D/DE NCU 573.2 ohne Systemsoftware	6FC5 357-0BB33-0AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D/DE NCU 573.2 Pentium II ohne Systemsoftware	6FC5 357-0BB33-0AE1	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 573.2 (Pentium Pro) für Digitalisieren mit Profibus DP	6FC5 357-0BA31-1AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D/DE NCU 573.2 ohne Systemsoftware	6FC5 357-0BB31-0AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 573.2 (Pentium Pro) (Exportversion) für 12 Achsen mit Profibus DP	6FC5 357-0BY32-1AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 573.2 (Pentium Pro) (Exportversion) für 31 Achsen mit Profibus DP	6FC5 357-0BY33-1AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01

NCU	MLFB	Enthaltene vergleichbare SIMATIC CPU-MLFB	Auswahl aus STEP7 Hardware Katalog
SINUMERIK 840D NCU 573.2 (Pentium Pro) (Exportversion) für Digitalisieren mit Profibus DP	6FC5 357-0BY31-1AE0	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840Di	6FC5 220-0AA00-1AA0	6ES7 315-2AF03-0AB0	810D/810Di mit PLC315-2AF03
SINUMERIK 840Di mit PK Bus		6ES7 315-2AF03-0AB0	810D/810Di mit PLC315-2AF03, PK-Bus
SINUMERIK 840D NCU 572.3	6FC5 357-0BB22-0AE0	Mit BESY 03.10.23: 6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 572.3	6FC5 357-0BB22-0AE0	Mit BESY 12.30.07: 6ES7 315-2AF03-0AB0	810D/840D mit PLC315-2AF03
SINUMERIK 840D NCU 573.3	6FC5 357-0BB33-0AE2	6ES7 315-2AF01-0AB0	810D/840D mit PLC315-2AF01
SINUMERIK 840D NCU 573.3	6FC5 357-0BB33-0AE2	Mit BESY 12.30.07: 6ES7 315-2AF03-0AB0	810D/840D mit PLC315-2AF03 (ab STEP7 V5.0 und Toolbox 05.03.05)
SINUMERIK 840D NCU 572.4	6FC5 357-0BB23-0AE0	6ES7 314-6CF00-0AB0	810D/840D mit PLC314C-2DP
SINUMERIK 840D NCU 573.4	6FC5 357-0BB34-0AE1	6ES7 314-6CF00-0AB0	810D/840D mit PLC314C-2DP
SINUMERIK 810D CCU3	6FC5 410-0AY03-0AA0	6ES7 315-2AF03-0AB0	810D/840D mit PLC315-2AF03
SINUMERIK 840D NCU 571.3	6FC5 357-0BB11-0AE1	6ES7 315-2AF03-0AB0	810D/840D mit PLC315-2AF03
SINUMERIK 840D NCU 561.3	6FC5356-0BB11-0AE1	6ES7 315-2AF03-0AB0	810D/840D mit PLC315-2AF03(ab STEP7 V5.0 und Toolbox 05.03.05)
SINUMERIK 840D NCU 571.4	6FC5357-0BB12-0AE0	6ES7 314-6CF00-0AB0	810D/840D mit PLC314C-2 DP (ab STEP7 V5.1 SP3 und Toolbox 06.03.02)
SINUMERIK 840D NCU 561.4	6FC5356-0BB12-0AE0	6ES7 314-6CF00-0AB0	810D/840D mit PLC314C-2 DP (ab STEP7 V5.1 SP3 und Toolbox 06.03.02)
SINUMERIK 840D NCU 573.5	6FC5357-0BB35-0A E0	6ES7 317-2AJ00-0AB0	810D/840D mit PLC317-2 DP (ab STEP7 V5.2 SP1 und Toolbox 06.05.02)
SINUMERIK 840Di mit MC12	6FC5 222-0AA02-1AA0	6ES7 317-2AJ00-0AB0	810Di mit PLC317-2 DP (ab STEP7 V5.2 SP1 und Toolbox 06.05.02)

**Hinweis**

Bei SINUMERIK 810D bzw. 840D ist die SIMATIC Zeile 0 in der NC enthalten. In dieser Zeile stecken folgende Komponenten:

- Steckplatz 2: die integrierte PLC (PLC 314 bzw. PLC 315-2DP)
- Steckplatz 3: eine IM 360
- Steckplatz 4: die FM NCU

Bei PLC 314 ist die FM NCU ab NC-Softwarestand 3.5 auch zu definieren, wenn weitere MPI (K-Bus) Teilnehmer in Zeile 1 bis Zeile 3 enthalten sind (z. B. FM-Baugruppen mit K-Bus Anschluss). Die Eigenschaften der FM NCU dürfen nicht verändert werden, da sonst Prozessalarme (z. B. Hilfsfunktionen) der NCU nicht mehr funktionieren.

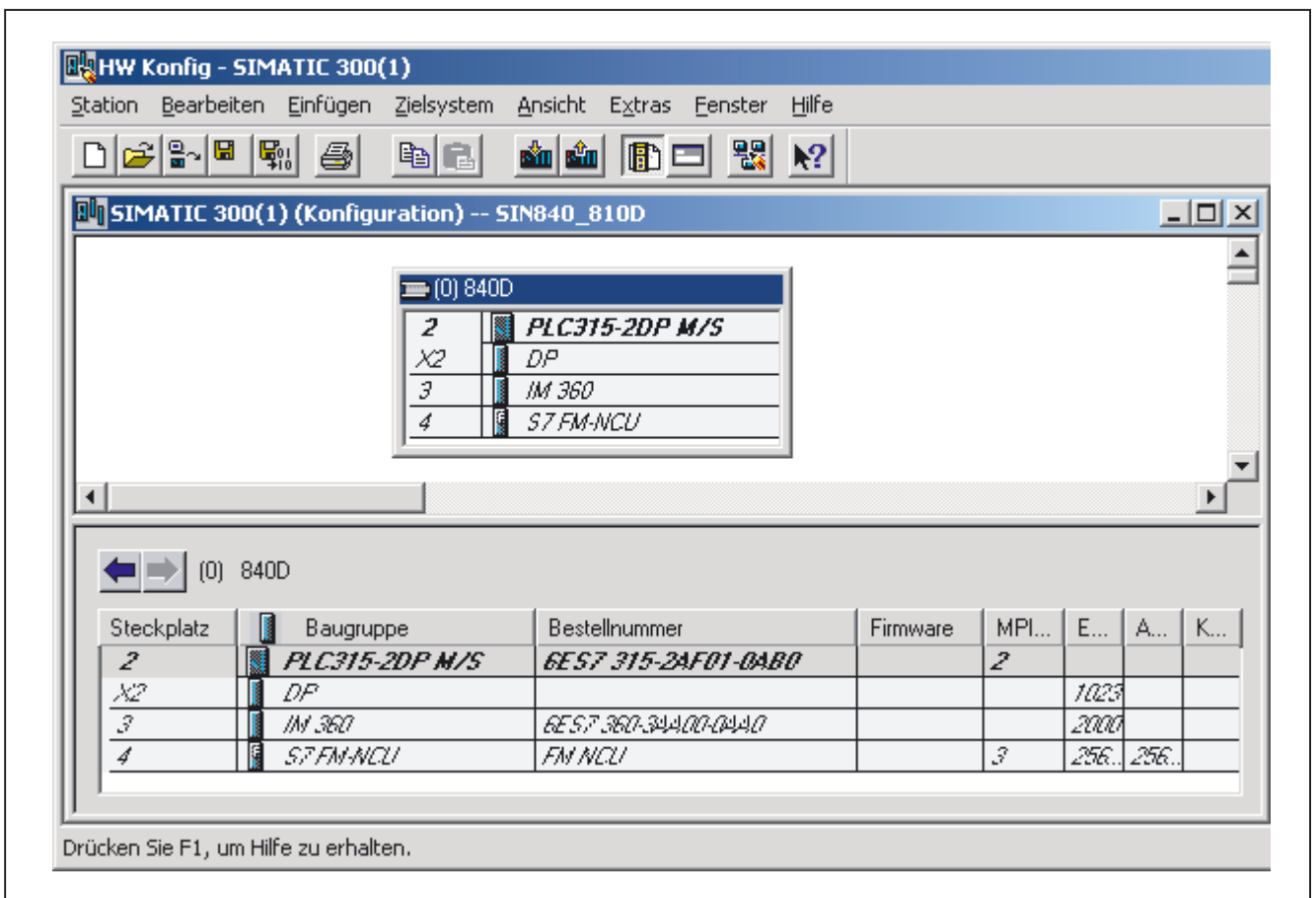


Bild 2-1 Hardwarekonfiguration der Zeile 0 bei 810D, 840D

## Maschinensteuertafel (MSTT) und Bedienhandgerät (BHG)

(gilt nur für SINUMERIK 810D bis SW 3.x)

Wird eine Projektierung der Maschinensteuertafel bzw. des Bedienhandgeräts vorgenommen, (Abweichung vom Standard), muss in das Maschinenprojekt eine weitere SIMATIC 300 Station je Bedienkomponente eingefügt werden. In dieser Station ist über die Hardware-Konfiguration (HW-Konfig) in der Zeile 0 auf Platz 2 eine beliebige CPU zu bestücken. Als MPI-Adresse ist die MPI-Adresse der Bedienkomponente einzustellen. Im SIMATIC-Manager kann dann MPI-Netz(1) markiert werden. Dann können über den Menüpunkt "Extras" die Globaldaten aktiviert werden. Weitere Vorgehensweise ist detailliert in der Inbetriebnahmeanleitung beschrieben.

## 2.4 Inbetriebnahme PLC-Programm

### 2.4.1 Installation des Grundprogramms für 810D, 840D

Vor der Erstinbetriebnahme der NC muss ein NC- und PLC-Urlöschen zum Initialisieren der jeweiligen Speicherbereiche durchgeführt werden. Dazu ist der Schalter S3 auf Stellung 1 und der Schalter S4 in Stellung 3 zu stellen und die Steuerung aus- und wieder einzuschalten (POWER ON-Reset).

#### Installation

Ab Toolbox-Softwarestand 6.1 wird die Installation über ein WINDOWS-konformes Setup-Programm für die Komponenten Grundprogramm, Hardwareauswahl in STEP7 (Optionspaket SINUMERIK 810D / 840D) und dem NC-Var Selektor durchgeführt. Hierzu ist das Programm "Setup.exe" im Basisverzeichnis der CD zu starten. Danach können die zu installierenden Komponenten ausgewählt werden. Nach der Installation ist die Grundprogramm-Bibliothek direkt von STEP 7 auswählbar (gp8x0d61, 61 ist hierbei die Haupt-Grundprogrammversion).

Die konkrete Version des Grundprogramms kann unter den Objekteigenschaften der Bibliothek bzw. des Programmordners im Kommentarfeld abgefragt werden.

Ab Softwarestand 3.7 bzw. 4.2 wird die Installation über das INSTALL.BAT (INSTALL1.BAT, INSTALL4.BAT) vorgenommen.

Dieses Programm installiert das Grundprogramm und Zusatzdateien für die jeweilige STEP7-Version. Bei der automatischen Installation werden auch ab STEP 7 Version 3 im HW-Katalog die TYP-, GSD- und Meta-Dateien aktualisiert und ergänzt.

Hiermit werden die Hardwarekomponenten der SINUMERIK für die Hardware-Projektierung unter STEP7 verfügbar. Ein Entpacken, wie unten erwähnt, ist somit nicht mehr erforderlich. Das Grundprogramm wird als **Projekt** für die STEP7 Version 1.x bzw. als **Bibliothek** für die STEP7 Version 2.x und deren Folgestände in gepackter Form geliefert.

#### Allgemeines

Die Quellprogramme der OBs inklusive Standard-Parametrierung, die Nahtstellensymbolik und DB-Vorlagen für Bedienhandgerät und M-Dekodierung sind im SIMATIC-Projekt bzw. SIMATIC-Bibliothek des Grundprogramms beigefügt.

Vor Installation des Grundprogramms muss STEP7 installiert sein.

### **STEP7 Version 1.x**

Das Grundprogramm befindet sich im Hauptverzeichnis der Diskette als gepackte Datei mit dem Namen GP840D.EXE (bzw. auch GP810D.EXE und GPFMNC.EXE). Das Grundprogramm (GP840D.exe) ist in das Hauptverzeichnis (Root) eines Laufwerks zu kopieren (z.B. c:\) und aufzurufen. Die für das Grundprogramm notwendige Projektstruktur wird automatisch erzeugt. Der Katalogname des Grundprogramms heißt GP840Dxy.S7A. Hierbei bedeutet xy die Version des Grundprogramms.

---

#### **Hinweis**

Bei STEP7 Version 1 muss der GP840Dxy.S7A - Katalog im Hauptverzeichnis stehen. Ein Katalog mit dem gleichen Namen GP840Dxy.S7A ist vorher zu löschen.

---

### **STEP7 Version 2.x, 3.x**

Das Grundprogramm befindet sich im Verzeichnis S7V2.840 bzw. S7V2.810 bzw. S7V2 der Grundprogramm-Diskette als gepackte Datei mit dem Namen "GP840D.EXE". Das Grundprogramm (GP840D.exe) ist in den Unterkatalog "S7LIBS" der STEP7 Version 2 (step7\_v2) bzw. deren Folgestände zu kopieren und aufzurufen. Die für das Grundprogramm notwendige Bibliotheksstruktur wird automatisch angelegt. Der Katalogname des Grundprogramms heißt GP840Dxy. Hierbei bedeutet xy die Version des Grundprogramms. Die Datei "MET.EXE" ist in den Basiskatalog von STEP7 zu kopieren und dort über das Dos-Fenster mit "MET.EXE -O" aufzurufen.

---

#### **Hinweis**

Der oben aufgeführte Name GP840D bezieht sich auf das Grundprogramm der SINUMERIK 840D. Bei der SINUMERIK 810D heißt das Grundprogramm GP810D, bzw. bei FMNC heißt das Grundprogramm GPFMNC. Ab Softwarestand 4.2 ist das Grundprogramm für 810D und 840D zusammengeführt. Der Name ist jetzt mit GP8x0D bezeichnet.

---

## 2.4.2 Anwendung des Grundprogramms

Für jede Anlage (Maschine) ist ein neues CPU-Programm in einem Projekt über die STEP7-Software anzulegen (z. B. "Drehma1").

### Anmerkung

Die Katalogstrukturen eines Projekts und die Vorgehensweise zur Erstellung von Projekten und Anwenderprogrammen sind in der zugehörigen SIMATIC-Dokumentation enthalten.

### STEP7 Version 1

Für das Maschinen-CPU-Programm ist über den Menüpunkt "Bearbeiten", "Konfiguration" eine Vernetzung zur PLC zu aktivieren. Dieses geschieht über das Menü "Dienste", "Parametrieren" und anschließender Auswahl der MPI-Parameter.

Standard:

- "vernetzt"
- "MPI Subnetznummer = 0"
- "CPU-MPI Adr = 2"

In das CPU-Programm für die maschinenspezifischen Programmdateien sind:

1. Die Bausteine des Grundprogramms (FCs, FBs, DBs, OBs, SFCs, SFBs und UDTs) zu kopieren (Menü "Datei", "Projekt verwalten" im Step7 Programmeditor).
2. Weiterhin ist die Datei GPOB840D.AWL (bzw. GPOB810D.AWL oder GPOBFMNC.AWL) und gegebenenfalls die weiteren AWL-Dateien aus dem Grundprogramm-katalog in dieses CPU-Programm zu kopieren. Die in dieser Datei vorhandenen OBs sind die Basis für das Anwenderprogramm mit den zugehörigen Grundprogramm-Aufrufen. Vorhandene Anwenderbausteine sind als AWL-Dateien in das neu angelegte CPU-Programm (Katalogname CPU1.S7D) hineinzukopieren und zu kompilieren.
3. Weiterhin ist es empfehlenswert, auch die symbolischen Namen aus dem Grundprogramm-Projekt mit dem Symboleditor zu übertragen.

### STEP7 Version 2

Das Kopieren der Grundprogrammbausteine erfolgt über den SIMATIC-Manager mit "Datei" / "Öffnen" / "Bibliothek".

Aus der Bibliothek müssen folgende Teile kopiert werden:

AP-off: FCs, FBs, DBs, OBs, SFC, SFB, UDT und der SDB-Container

Im SDB-Container ist für 810D enthalten:

- die Source\_files (SO):
  - GPOB810D bzw. GPOB840D
  - gegebenenfalls MDECLIST, BHG\_DB und weitere
- die Symboltabelle (SY)

---

### Hinweis

Der SDB-Container existiert auch nur für diese Steuerungsvarianten.

---



## Kompatibilität zu STEP7

Es sind keine Abhängigkeiten des Grundprogramms (auch ältere Versionen des Grundprogramms) zu den derzeit gültigen STEP 7-Versionen vorhanden.

### 2.4.3 Versionskennzeichnungen

#### Grundprogramm

Die Version des Grundprogramms, inklusive Steuerungstyp, wird im Versionsbild der HMI-Software angezeigt.

#### Anwenderprogramm

Der Anwender kann seine eigene PLC-Anwenderprogramm Versionskennzeichnung in einem beliebigen Datenbaustein, ein Datum vom Datentyp String mit max. 54 Zeichen, definieren. Als Inhalt kann ein beliebiger Text stehen. Die Parametrierung auf diesen String erfolgt über einen Zeiger am FB 1. Hierzu muss der Datenbaustein symbolisch definiert sein.

**Literatur:**

/FB1/ Funktionshandbuch Grundfunktionen; PLC – Grundprogramm (P3);  
Kapitel: Bausteinbeschreibung

### 2.4.4 Maschinenprogramm

Das Maschinenprogramm wird durch den Maschinenhersteller erstellt unter Zuhilfenahme der Bibliotheksroutinen des Grundprogramms. Im Maschinenprogramm sind die logischen Verknüpfungen und Abläufe der Maschine enthalten. Weiterhin werden die Nahtstellensignale zur NC bedient. Komplexere Kommunikationsfunktionen zur NC (z.B. NC-Daten lesen, schreiben, Werkzeugverwaltungs-Quittungen usw. werden über FC, FB des Grundprogramms angestoßen und ausgeführt). Das Maschinenprogramm kann in unterschiedlichen Erstellungssprachen (z.B. AWL, KOP, FUP, S7 HIGRAPH, S7GRAPH, SCL) erstellt werden. Das komplette Maschinenprogramm muss in der richtigen Reihenfolge erstellt und kompiliert sein. Das bedeutet, dass Bausteine, die von anderen Bausteinen aufgerufen werden, generell vor diesen Bausteinen kompiliert werden müssen. Wenn diese aufgerufenen Bausteine nachträglich im Zuge der Programmentwicklung in der Schnittstelle (VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT, VAR) geändert werden, dann sind auch der rufende Baustein und alle damit verbundenen Bausteine anschließend zu kompilieren. Diese Vorgehensweise gilt sinngemäß auch für Instanz-Datenbausteine für FBs. Wenn diese Reihenfolge nicht eingehalten wird, treten ansonsten Zeitstempel-Konflikte bei der Rückübersetzung in STEP7 auf. Somit ist die Rückübersetzbarkeit der Bausteine nicht sichergestellt und bereitet unter anderem auch Probleme bei der Funktion "Status Baustein". Es ist außerdem empfehlenswert, die im Kontaktplan oder in Einzelanweisung (inkrementeller Modus) erstellten Bausteine in ASCII-AWL durch den STEP7 Editor zu generieren.

### 2.4.5 Datensicherung

Die PLC-CPU speichert keine symbolischen Namen mit ab, sondern nur die Datentypbeschreibungen der Bausteinparameter:

- VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT, VAR und
- die Datentypen der globalen Datenbausteine.

---

#### Hinweis

Ohne das zugehörige Projekt für diese Maschine ist keine vernünftige Rückübersetzung möglich. Dies betrifft insbesondere z.B. die Funktion Status Baustein oder nachträglich notwendige Änderungen der PLC-CPU-Programme. Es ist deshalb erforderlich, das sich in der PLC-CPU befindende STEP7-Projekt an der Maschine unbedingt aufzubewahren. Für einen Servicefall ist dies eine große Hilfe und erspart einen unnötigen Zeitaufwand das ursprüngliche zugehörige Projekt wieder herzustellen.

---

Wenn das STEP7-Projekt vorhanden ist und nach den genannten Regeln erstellt wurde, kann an dieser Maschine in der PLC-CPU symbolisch gearbeitet werden. Gegebenfalls sind auch die Quell-Programme der Maschine als .awl-Dateien mit zu hinterlegen für eine eventuell notwendige Hochrüstung.

Als Minimum sollten die Quellprogramme aller Organisationsbausteine und auch aller Instanz-Datenbausteine vorhanden sein.

### 2.4.6 PLC Serien Inbetriebnahme, PLC Archive:

#### Serienarchiv erzeugen

Nach dem Laden der Bausteine in die PLC CPU kann über die HMI-Bedienoberfläche ein Serienarchiv erzeugt werden zur Datensicherung an der Maschine. Die Datensicherung soll direkt nach dem Laden der Bausteine im PLC Stop Zustand erfolgen um Konsistenz der Daten zu erreichen. Diese Datensicherung ersetzt nicht die Sicherung des SIMATIC-Projektes, da im Serien-Archiv nur Binärdaten abgespeichert werden. Hier stehen z.B. keine symbolischen Informationen zur Verfügung. Weiterhin werden keine CPU-DBs (SFC 22 DBs) und in der CPU erzeugte SDBs gesichert.

Ab der Toolbox 06.03.03 und der STEP 7 Version 5.1 kann das PLC-Serienarchiv direkt aus dem passenden SIMATIC-Projekt erzeugt werden.

Hierzu ist in STEP 7 unter dem Menüpunkt "Extras" → "Einstellungen" der Tabulator "Archivieren" auszuwählen. Dort ist ein Eintrag "SINUMERIK (\*.arc)" enthalten, der ausgewählt werden muss zum Erzeugen eines Serien-Inbetriebnahme-Files. Nach Auswahl dieses Archivs wird der Menüpunkt "Datei" → "Archivieren" angewählt. Nach entsprechender Auswahl wird das Serien-Archiv erzeugt. Falls das Projekt mehrere Programme enthält, kann der Programmpfad ausgewählt werden. Für den ausgewählten Programmpfad wird das Serienarchiv angelegt. Es werden alle Bausteine in das Archiv aufgenommen, die in dem Programmpfad enthalten sind, ausser CPU-DBs (SFC 22 DBs).

Die Erzeugung des Serienarchivs ist auch automatisierbar (vergleichbar mit der Kommandoschnittstelle von STEP 7, ab Version 5.1). Diese Erzeugung stellt eine Erweiterung der Kommandoschnittstelle dar.

## Funktionen

Folgende Funktionen stehen in dieser Erweiterung zur Verfügung:

Die Funktionen (hier mit VB-Script dargestellt) sind erst nach Aufruf von Serverinstanzierungen und Magic-Aufruf verfügbar:

Funktionen:

Const S7BlockContainer = 1138689, S7PlanContainer = 17829889	
Const S7SourceContainer = 1122308	
set S7 = CreateObject("Simatic.Simatic.1")	
rem Kommandoschnittstelle von STEP7 instanzieren	
Set S7Ext = CreateObject("SimaticExt.S7ContainerExt")	
Call S7Ext.Magic("")	

Function **Magic**(bstrVal As String) As Long

Über Aufruf wird Zugang zu bestimmten Funktionen erreicht. Die Funktion muss nach Serverinstanzierung einmalig aufgerufen werden. Der Wert von bstrVal kann leer sein. Hiermit wird die korrekte Step7 Version und Path Angabe in Autoexec geprüft. Bei Rückgabe von 0 sind die Funktionen freigeschaltet.

Rückgabe (-1) = falsche Step7 Version

Rückgabe (-2) = kein Eintrag in Autoexec.bat

Function **Magic**(bstrVal As String) As Long

Function **MakeSerienIB**(FileName As String, Option As Long, Container As S7Container) As Long

Parameter "Option":

0:           Normaler SerienIB File mit Urlöschen.

Bit 0 = 1:    SerienIB File ohne Urlöschen. Wenn SDBs im Projekt sind, ist diese Option nicht wirksam.  
Es wird dann immer ein Urlöschen durchgeführt.

Bit 1 = 1:    SerienIB File mit PLC Neustart

Rückgabe-Wert:

0           = OK

-1          = Funktion nicht verfügbar, Funktion Magic vorher aufrufen

-2          = Filename nicht erzeugbar

-4          = Parameter Container ungültig oder BausteinContainer leer

-5          = interner Fehler (Speicheranforderung durch Windows abgelehnt)

-6          = interner Fehler (Problem im STEP7 Projekt)

-7          = Schreibfehler beim Erzeugen des SerienIB Files (z. B. Diskette voll)

## Nutzung im Script

```
If S7Ext.Magic("") < 0 Then
    Wscript.Quit(1)
End If
Set Proj1 = s7.Projects("neu")
set S7Prog = Nothing
Set s7prog = Proj1.Programs.Item(1) 'wenn es nur ein Programm gibt
For i = 1 to S7Prog.Next.Count
    Set Cont = S7Prog.Next.Item(i)
    Bausteincontainer prüfen
    If (Cont.ConcreteType = S7BlockContainer) Then
        Exit For
    End if
Next
Fehler = S7Ext.MakeSerienIB("f:\dh\arc.dir\PLC.arc", 0, Cont)'Jetzt Fehlerauswertung
```

## 2.4.7 Softwarehochrüstung

### Software-Hochrüstung

Bei jeder PLC- bzw. NCK-Software-Hochrüstung ist ein Urzustand der PLC herzustellen. Dieser Urzustand kann über das Urlöschen der PLC erfolgen. Bei diesem Urlöschen werden alle vorhandenen Bausteine gelöscht.

Im Regelfall ist bei einem neuen NC-Softwarestand das neue Grundprogramm mit einzubinden. Hierzu ist die Übertragung der Grundprogrammbausteine in das Anwenderprojekt erforderlich. Hierbei sollten keine OB 1, OB 40, OB 100, der FC 12 und auch der DB 4 nicht mit übertragen werden, wenn diese Bausteine sich schon im Anwenderprojekt befinden. Diese vorgenannten Bausteine sind evtl. vom Anwender verändert worden. Das neue Grundprogramm ist mit dem Anwenderprogramm zusammenzubinden.

Hierzu ist folgende Vorgehensweise zu beachten:

1. Vor dem Kopieren des Grundprogramms von allen Anwender-Bausteinen die Text- bzw. Quell-Datei generieren.
2. Dann die neuen Grundprogrammbausteine in dieses Maschinenprojekt kopieren (Beschreibung siehe Kapitel "Anwendung des Grundprogramms")
3. Danach sind alle Anwenderprogramme \*.awl in korrekter Reihenfolge neu zu übersetzen! (siehe auch Kapitel "Maschinenprogramm"). Dieses neu kompilierte Maschinenprogramm ist dann mit STEP7 in die PLC-CPU zu laden.

Im Normalfall reicht jedoch ein neues Übersetzen der Organisationsbausteine (OB) und der Instanz-Datenbausteine des Maschinenprogramms aus. D.h. es brauchen auch nur Quellen für die Organisationsbausteine und die Instanz-Datenbausteine erzeugt werden (vor der Hochrüstung).

## Urlöschen

Das Urlöschen der PLC ist in der Inbetriebnahmeanleitung beschrieben. Durch dieses Urlöschen wird jedoch nicht der Diagnosepuffer und auch nicht die Teilnehmeradresse am MPI-Bus gelöscht. Eine weitere Vorgehensweise des Urlöschens ist unten beschrieben. Dieses Urlöschen ist dann anzuwenden, wenn die normale Urlöschroutine versagt.

Die Vorgehensweise hierzu ist:

Nr.	Handlung	Wirkung
1	Steuerung ist ausgeschaltet	
2	PLC-Schalter Stellung 3 (MRES) und Steuerung wieder einschalten oder Hardware-Reset.	LED mit der Beschriftung PS blinkt langsam.
3	PLC-IBN-Schalter in Stellung 2 (STOP) einschalten, dann wieder in die Stellung 3 schalten.	Es erfolgt ein schnelleres Blinken der LED PS.
4	PLC-IBN-Schalter in Stellung 2 oder 0 schalten.	

## NC-Variablen

Für jeden NC-Softwarestand (auch ältere Versionen) kann der neueste NC-VAR-Selektor verwendet werden. Für ältere NC-Softwarestände können die Variablen auch aus der neuesten Gesamtliste selektiert werden. Der Informationsinhalt im DB 120 (Default DB für Variablen) ist nicht abhängig vom Softwarestand. D.h. selektierte Variablen in einem älteren Softwarestand müssen bei einer Softwarehochrüstung nicht neu selektiert werden.

### 2.4.8 Peripheriebaugruppen (FM-, CP-Baugruppen)

Für die komplexeren Peripherie-Baugruppen werden in der Regel Zusatzpakete zu STEP7 benötigt. In diesen Zusatzpaketen werden zum Teil in einer STEP7 Bibliothek Unterstützungsbausteine (FC, FB) mitgeliefert. Diese Bausteine beinhalten Funktionen zum Betrieb der jeweiligen Baugruppe. Diese Funktionen werden vom Anwenderprogramm parametrisiert und aufgerufen. In vielen Fällen sind die FC-Nummern der Hantierungsbausteine für die CP-, FM-Baugruppe im Nummernbereich des Grundprogramms von 810D, 840D vorgesehen.

Was kann man bei diesen Konflikten tun?

Die Baustein-Nummern des Grundprogramms müssen unverändert bleiben. Die Baustein-Nummern der benötigten Hantierungsbausteine können auf freie Nummern über STEP7 umbenannt werden. Diese neuen Bausteine (neue FC-Nummern) werden dann im Anwenderprogramm mit der für die Funktion notwendigen Parametrierung aufgerufen.

### 2.4.9 Fehler-Beseitigung

Dieser Abschnitt soll Hinweise zu Problemfällen und deren Beseitigung bzw. auch Ursachen geben, bevor ein Hardwaretausch erfolgt.

Fehler, Ursache/Beschreibung und Abhilfe			
Ifd. Nr. Fehlerhinweis	Fehler	Ursache/Beschreibung	Abhilfe
1	Keine Verbindung über MPI zur PLC.	Das MPI-Kabel ist nicht gesteckt oder defekt. Möglicherweise ist auch die STEP 7 Software für die MPI-Karte nicht richtig konfiguriert.	Versuch: Mit dem PG im STEP7 Editor über die Verbindung "Direct_PLC" ankoppeln. Hierbei müssen einige Teilnehmeradressen eingeblendet werden. Fehlen die, dann MPI-Kabel defekt / nicht gesteckt.
2	Trotz Urlöschen der PLC ist kein Zugriff zur PLC möglich.	Ein Systemdatenbaustein SDB 0 wurde mit einer geänderten MPI-Adresse geladen. Hierdurch tritt ein MPI-Buskonflikt wegen doppelt vergebener Adressen auf.	Alle MPI-Kabel zu anderen Komponenten trennen. Mit dem PG die Verbindung "Direct_PLC" aufbauen. Durch Taufe der MPI-Adresse die MPI-Adresse korrigieren.
3	Alle 4 LEDs der PLC blinken (DI-Katastrophe)	Ein Systemfehler in der PLC ist aufgetreten. <b>Maßnahmen:</b> Zur genaueren Analyse des Systemfehlers ist der Diagnosepuffer der PLC auszulesen. Hierzu ist die PLC zuerst in den Stoppzustand zu bringen (z.B. Schalter "PLC" auf Stellung 2. Anschließend ist ein Hardware-RESET erforderlich. Danach kann mit STEP7 der Diagnosepuffer ausgelesen werden. Die Informationen aus dem Diagnosepuffer sind der Hotline / Entwicklung zu melden. Wenn nach dem Hardware-RESET zusätzlich ein Urlöschen angefordert wird, ist erst das Urlöschen durchzuführen. Anschließend kann dann im Stoppzustand der Diagnosepuffer gelesen werden.	Nach RESET bzw. nach Neuladen des PLC-Programms läuft das System unter Umständen normal weiter. In jedem Fall ist der Diagnosepuffer der Entwicklung zur Verfügung zu stellen.

## 2.5 Ankopplung der PLC-CPU's an 810D, 840D

### 2.5.1 Allgemein

Als PLC wird für alle Systeme die AS 300 Familie verwendet. Der Unterschied bei den verschiedenen NCU Varianten besteht im wesentlichen in der Art der Ankopplung. Bei 840D und 810D ist die PLC 314 CPU (Anwender-Speicherausbau bis 128 kByte) bzw. PLC 315-2 DP (Anwender-Speicherausbau bis 288 kByte) als Submodul in die NC-Einheit integriert. Die PLC / CPU 315-2 DP unterstützt zusätzlich dezentrale Peripherie am Profibus (L2DP). Die jeweiligen Leistungsdaten der PLC-CPU's sind obiger Tabelle bzw. dem Katalog FT70 zu entnehmen.

### 2.5.2 Eigenschaften der PLC CPU's

Die SINUMERIK 810D / 840D / 840Di PLC-CPU's sind abgeleitet von Standard SIMATIC CPU's der S7-300 Familie. Im Regelfall besitzen sie dadurch gleiche Funktionalität. Die Abweichungen sind in obiger Tabelle dargestellt. Durch das teilweise unterschiedliche Speicherkonzept gegenüber der S7 CPU sind bestimmte Funktionen nicht vorhanden (z.B. Bausteine auf Memory Card, Projekt auf Memory Card)

---

#### **Hinweis**

Bei den aktuellen SIMATIC CPU's wird bei Auslösen des PLC Stops über Softwarebedienung kein automatisches Anstarten der PLC nach Spannungsausfall und -Wiederkehr durchgeführt. Hierbei verbleibt die PLC aus Sicherheitsgründen im Stoppzustand mit entsprechendem Diagnoseeintrag. Ein Starten der PLC ist nur möglich über Softwarebedienung "Neustart ausführen" oder durch Betätigen des Schalters in "Stop" und anschließenden "RUN". Dieses Verhalten ist auch in den aktuellen Versionen der SINUMERIK PLC's enthalten.

---

### 2.5.3 Nahtstelle bei 810D und 840D mit integrierter PLC

#### Physikalische Nahtstellen

Als integrierte PLC ergibt sich bei 810D und 840D die Möglichkeit, den Austausch der Signale zwischen NCK und PLC direkt über ein Dual-Port-RAM vorzunehmen.

#### Austausch mit Bedientafelfront und MSTT

Der Datenaustausch mit der Bedientafelfront (BT), Maschinensteuertafel (MSTT) und Bedienhandgerät (BHG) erfolgt bei der 840D in der Regel über die Bedientafelfront-Schnittstelle (BTSS), wobei das COM-Modul den Datentransport durchführt.

Alle aufgeführten Geräte lassen sich bei der 840D aber auch an der MPI-Schnittstelle betreiben. Bei der 810D erfolgt der Datenaustausch mit der Bedientafelfront (BT), Maschinensteuertafel (MSTT) und Bedienhandgerät (BHG) nur über die MPI-Schnittstelle.

Das Programmiergerät wird über die MPI-Schnittstelle (Multi-Point-Interface) direkt an die PLC angeschlossen.

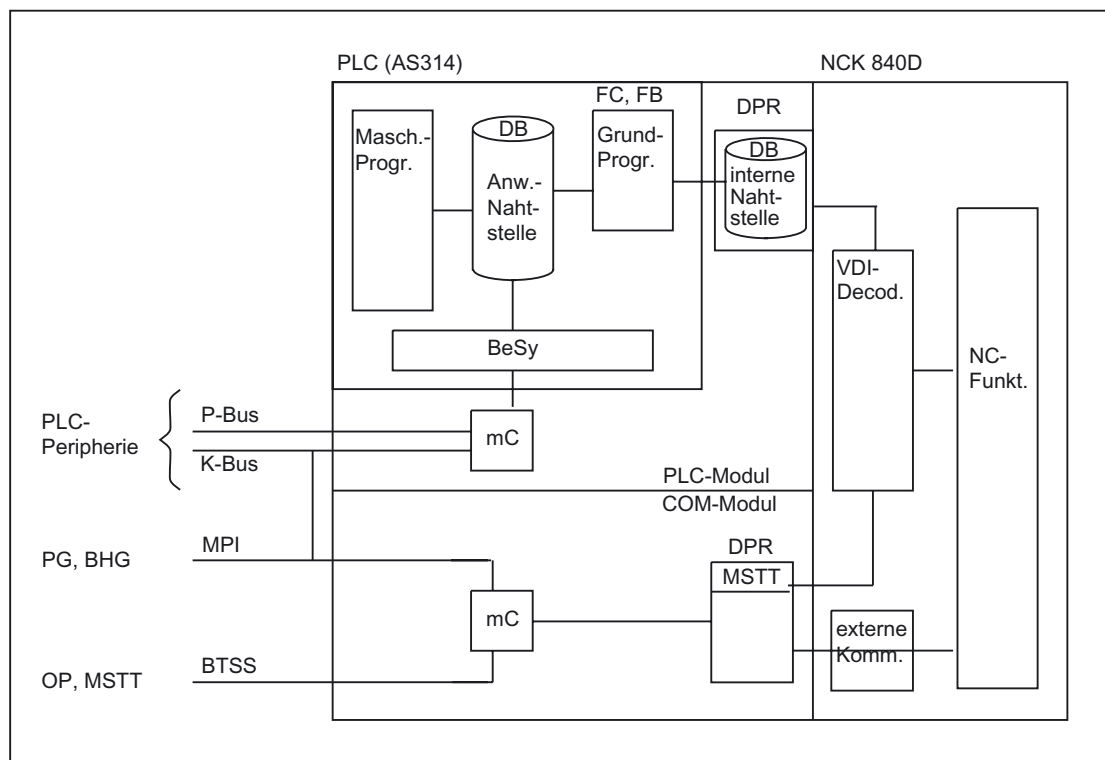


Bild 2-2 NCK/PLC-Kopplung bei 810D, 840D (integrierte PLC)



## Nahtstelle NCK/PLC

Der Datenaustausch NCK/PLC wird auf PLC-Seite vom Grundprogramm organisiert. Die von der NC im internen Dual-Port-RAM (DPR) abgelegten **Statusinformationen** (wie z. B. "Programm läuft") werden vom Grundprogramm am Zyklusanfang (OB 1) in Datenbausteine kopiert, auf die der Anwender dann zugreifen kann (Anwendernahtstelle). Die vom Anwender in die Nahtstellen-DBs eingetragenen **Steuersignale** an die NC (z. B. NC-Start) werden ebenso am Zyklusanfang an den NC übertragen.

Die abhängig vom Werkstückprogramm an die PLC übergebenen **Hilfsfunktionen** werden zunächst alarmgesteuert vom Grundprogramm ausgewertet und dann am Anfang vom OB 1 an die Anwendernahtstelle übergeben. Sind im betreffenden NC-Satz Hilfsfunktionen enthalten, die eine Unterbrechung der NC-Bearbeitung erfordern (z. B. M06 für Werkzeugwechsel), wird die NC-Satz-Decodierung in der NC vom Grundprogramm zunächst für eine PLC-Zykluszeit angehalten. Der Anwender kann dann über das Nahtstellensignal "Einlesesperre" die Decodierung so lange anhalten, bis z.B. der Werkzeugwechsel abgeschlossen ist. Sind dagegen im betreffenden NC-Satz nur Hilfsfunktionen enthalten, die keine Unterbrechung der Decodierung erfordern (z. B. M08 für "Kühlmittel Ein"), so wird die Übergabe dieser "schnellen" Hilfsfunktionen direkt im OB 40 quittiert, so dass die Decodierung von der Übergabe an die PLC nur unwesentlich beeinflusst wird.

Die vom NCK übergebenen **G-Funktionen** werden ebenfalls alarmgesteuert ausgewertet und quittiert, dann jedoch direkt an die Anwendernahtstelle übergeben. Wird eine G-Funktion an mehreren Stellen im PLC-Programm ausgewertet, können sich Unterschiede in der Information der G-Funktion innerhalb eines PLC-Zyklus ergeben.

Bei **NC-Aktionen**, die von der PLC aus angestoßen und parametrierbar sind (z. B. das Verfahren von konkurrierenden Achsen), erfolgt der Anstoß und die Parametrierung nicht über Nahtstellen-DBs, sondern über FCs und FBs. Die zu den NC-Aktionen gehörenden FCs werden zusammen mit dem Grundprogramm ausgeliefert. Die jeweils benötigten FCs und FBs müssen vom Anwender geladen und in dem PLC-Programm des Maschinenherstellers (Maschinenprogramm) aufgerufen werden. Eine Übersicht der Bausteine FC, FB und DB getrennt nach Basisfunktion und erweiterter Funktion finden Sie in:

### Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; PLC-Grundprogramm (P3);  
"Inbetriebnahme PLC-Programm"

## Nahtstelle OP/PLC

Der Datenaustausch OP/PLC erfolgt über den seriellen Bus OP/NC, COM-Modul und K-Bus. Das COM-Modul überträgt die Daten unverändert vom einem Bus-Segment in das andere. Er setzt lediglich die Baud-Rate um. Das OP ist immer der aktive Partner (Client) und die PLC immer der passive Partner (Server). Die vom OP angeforderten bzw. übergebenen Daten werden vom PLC-Betriebssystem aus dem Nahtstellenbereich OP/PLC ausgelesen bzw. in ihn eingetragen (Zeitpunkt: Zykluskontrollpunkt). Aus PLC-Anwendersicht verhalten sie sich wie E-/A-Signale.

## Nahtstelle MSTT/PLC, Nahtstelle BHG/PLC (nur 840D)

Der Datenaustausch MSTT/PLC, BHG/PLC erfolgt über den seriellen Bus MSTT, BHG/NC, COM-Modul und NCK. Vom NCK werden die MSTT, BHG-Signale in das interne NC-/NCK-DPR (Dual-Port-RAM) abgelegt bzw. von ihm abgeholt. Auf PLC-Seite übernimmt das Grundprogramm den Austausch mit der Anwendernahtstelle. Über die Parameter des Grundprogramms werden die Operandenbereiche (z. B. E/A) und die Anfangsadressen festgelegt.

## Nahtstelle MSTT/PLC, Nahtstelle BHG/PLC (810D)

Der Datenaustausch MTT/PLC, BHG/PLC erfolgt über die MPI-Nahtstelle der PLC. Es wird hierfür der Dienst Kommunikation mit Globaldaten (GD)<sup>1)</sup> verwendet (Siehe hierzu STEP 7 Benutzerhandbuch). Das PLC-Betriebssystem übernimmt den Transfer der Signale von und zur Anwendernahtstelle. Über das STEP 7-Projektierwerkzeug **Communication-Configuration** werden neben den GD-Parametern auch die Operandenbereiche (z.B. E/A) und deren Anfangsadressen festgelegt. Ab Softwarestand 2.2 ist der Datenaustausch wie bei 840D möglich.

<sup>1)</sup> IK (GD) = Implizite Kommunikation (Globale Daten)

## 2.5.4 Diagnosepuffer der PLC

### Allgemeines

Im Diagnosepuffer der PLC (auslesbar mit STEP 7) werden Diagnoseinformationen des PLC-Betriebssystems angezeigt. Darüber hinaus werden durch das Grundprogramm und die Funktion: "Alarmer / Meldungen" über den FC10 Einträge in den Diagnosepuffer vorgenommen. Diese Alarmer und Meldungen werden im Diagnosepuffer als Ereignis mit "Ereignis-ID: <ID>" ohne erklärenden Text angezeigt.

### Bedeutung der angezeigten Daten

Im Folgenden wird anhand eines beispielhaften Ereignisses erläutert, welche Informationen angezeigt werden und wie diese zu interpretieren sind.

Diagnosepuffer (Ausschnitt, relevante Daten markiert):

Details zum Ereignis: 1 von 10	Ereignis-ID: 16# B046
Kein Eintrag in Textdatenbasis. Hex-Werte werden angezeigt.	
Ereignis-ID:	16# <b>B046</b>
OB:	16# 01
PK:	16# 01
DatID 1/ 2:	16# 59 C9
Zusatzinfo 1 / 2 / 3:	16# 0200 0000 0020
gehendes Ereignis:	36:02:459 08.04.03

### Typ des Ereignisses

In den beiden höchstwertigen Stellen der Ereignis-ID: 16# **B046** ist der Typ des Ereignisses verschlüsselt:

A1 :	Alarm gesetzt
A0 :	Alarm gelöscht
B1 :	Meldung gesetzt
B0 :	Meldung gelöscht

### **Alarm- / Meldungsnummer**

In den beiden niederwertigsten Stellen der Ereignis-ID: 16# B046 sind die höchstwertigen Dezimalstellen der Alarm- bzw. Meldungsnummer verschlüsselt. Zur Ermittlung der vollständigen Alarm- bzw. Meldungsnummer sind zusätzlich die beiden niederwertigsten Stellen der Zusatzinfo 1 / 2 / 3 zu berücksichtigen:

- Zusatzinfo 1: 16# 0200 0000 0020
- Zusatzinfo 2: 16# 0200 0000 0020

Alle Werte sind dezimal umzurechnen und zur Alarm- bzw. Meldungsnummer aneinander zu reihen:

Ereignis-ID:	46 <sub>hex</sub>	=	70 <sub>dec</sub>
Zusatzinfo 1:	00 <sub>hex</sub>	=	00 <sub>dec</sub>
Zusatzinfo 2:	20 <sub>hex</sub>	=	32 <sub>dec</sub>
Alarmnummer (BM) <sup>1)</sup>		=	<b>70 00 32</b>

1) Gesteuert durch DB2.DBX184.0  
BM = Betriebsmeldung

Somit bedeutet das Ereignis mit der Ereignis-ID: 16# B046 und den genannten Zusatzinformationen: "Meldung 700032 gelöscht".

### **Hinweise**

- Die Bedeutung der Meldungsnummer ist durch den Maschinenhersteller festgelegt.
- Ereignissen mit der Ereignis-ID 16# xx28 oder 16# xx29 werden vom Grundprogramm erzeugt.
- Die im Diagnosepuffer stehenden Meldungen können über die Bedienoberfläche mit zugehörigen Meldetexten abgelesen werden.

## 2.6 Struktur der Nahtstelle

### 2.6.1 Allgemein

#### Nahtstelle Datenbausteine

Die PLC-Anwendernahtstelle ist bei 840D und 810D bis auf das Mengengerüst identisch. Aufgrund der Vielzahl der Signale ist die Abbildung in Nahtstellen-DBs notwendig. Aus PLC-Programmsicht sind dies globale DBs. Das Grundprogramm erzeugt diese DBs beim Systemanlauf anhand von aktuellen NC-Maschinendaten (Anzahl der Kanäle, Achsen etc.). Dies hat den Vorteil, dass nur soviel PLC-RAM belegt wird, wie für die aktuelle Maschinenkonfiguration benötigt wird.

### 2.6.2 Nahtstelle PLC/NCK

#### Allgemeines

Die Nahtstelle PLC/NCK wird einerseits durch eine Datenschnittstelle und andererseits durch eine Funktionsschnittstelle gebildet. In der Datenschnittstelle sind Status- und Steuersignale, Hilfs- und G-Funktionen enthalten, während über die Funktionsschnittstelle Aufträge von der PLC an den NCK übergeben werden.

#### Datenschnittstelle

Die Datenschnittstelle wird in folgende Gruppen unterteilt:

- NCK-spezifische Signale
- BAG-spezifische Signale
- Kanal-spezifische Signale
- Achs-/Spindel-/Antriebsspezifische Signale

#### Funktionsschnittstelle

Die Funktionsschnittstelle wird durch FBs und FCs gebildet. Das folgende Bild zeigt die generelle Struktur der Nahtstelle zwischen PLC und NCK.

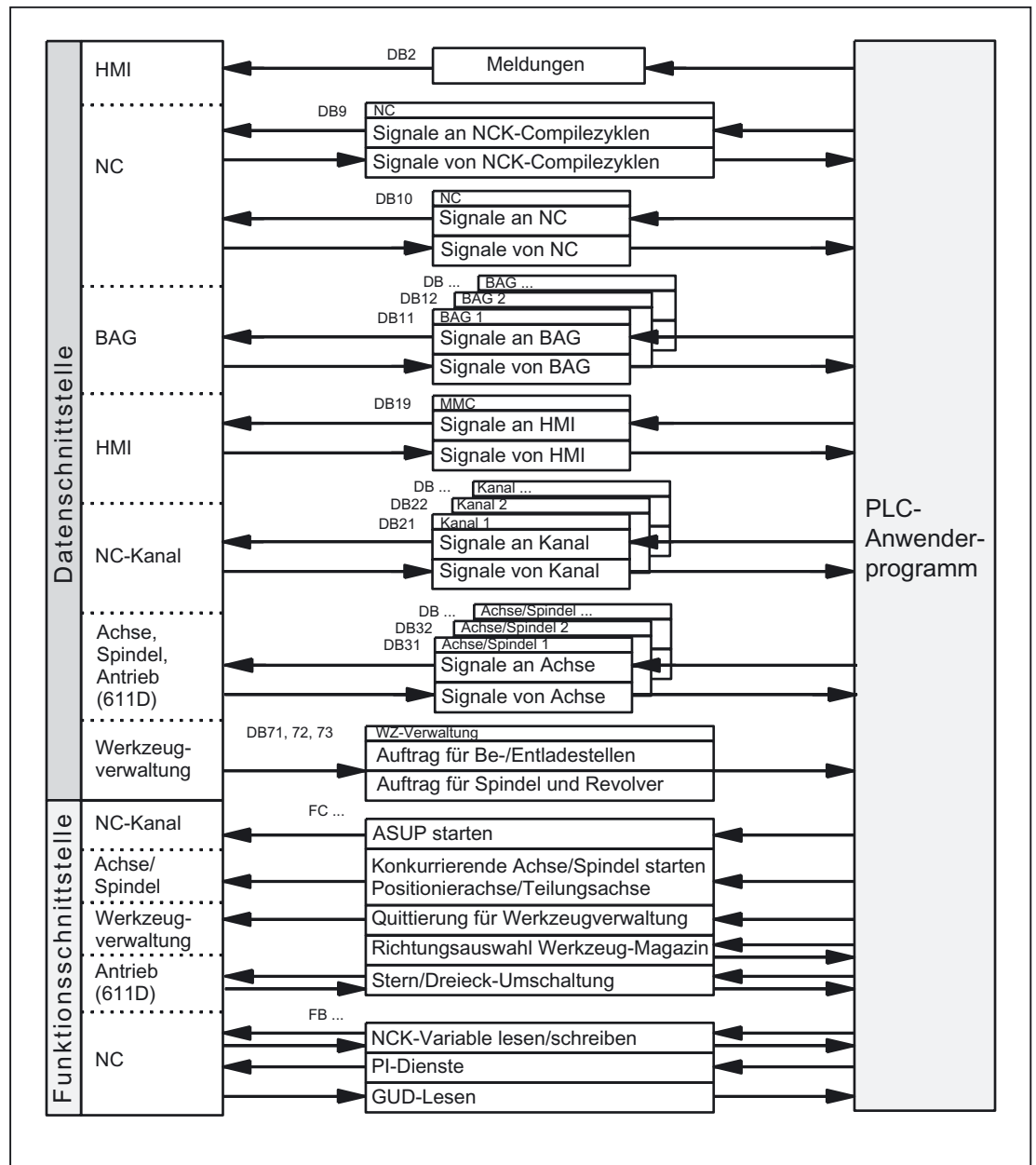


Bild 2-3 Nahtstelle PLC/NCK

Prinzip

### Compilezyklen-Signale

Neben den standardmäßig vorhandenen Signalen zwischen PLC und NCK wird bei Bedarf ein Nahtstellen-DB für Compile-Zyklen erzeugt (DB9). Die zugehörigen Signale, die abhängig von den jeweiligen Compilezyklen sind, werden zyklisch zu Beginn des OB1 übertragen.

**Signale PLC/NC**

In die Gruppe der Signale von PLC an NC fallen:

- Signale für die Beeinflussung der digitalen und analogen E-/A-Signale der NC
- Schlüsselschalter- und Not-Aus-Signale

**Signale NC/PLC**

In die Gruppe der Signale von NC an PLC fallen:

- Istwerte der digitalen und analogen E-/A-Signale der NC
- Bereitschafts- und Statussignale der NC

Weiterhin sind hier auch die Handradanwahlsignale und die Kanal-Statussignale von HMI abgelegt.

Die Signale zur Handradanwahl werden vom Grundprogramm decodiert und in die maschinen- bzw. kanalachsenspezifische Nahtstelle eingetragen.

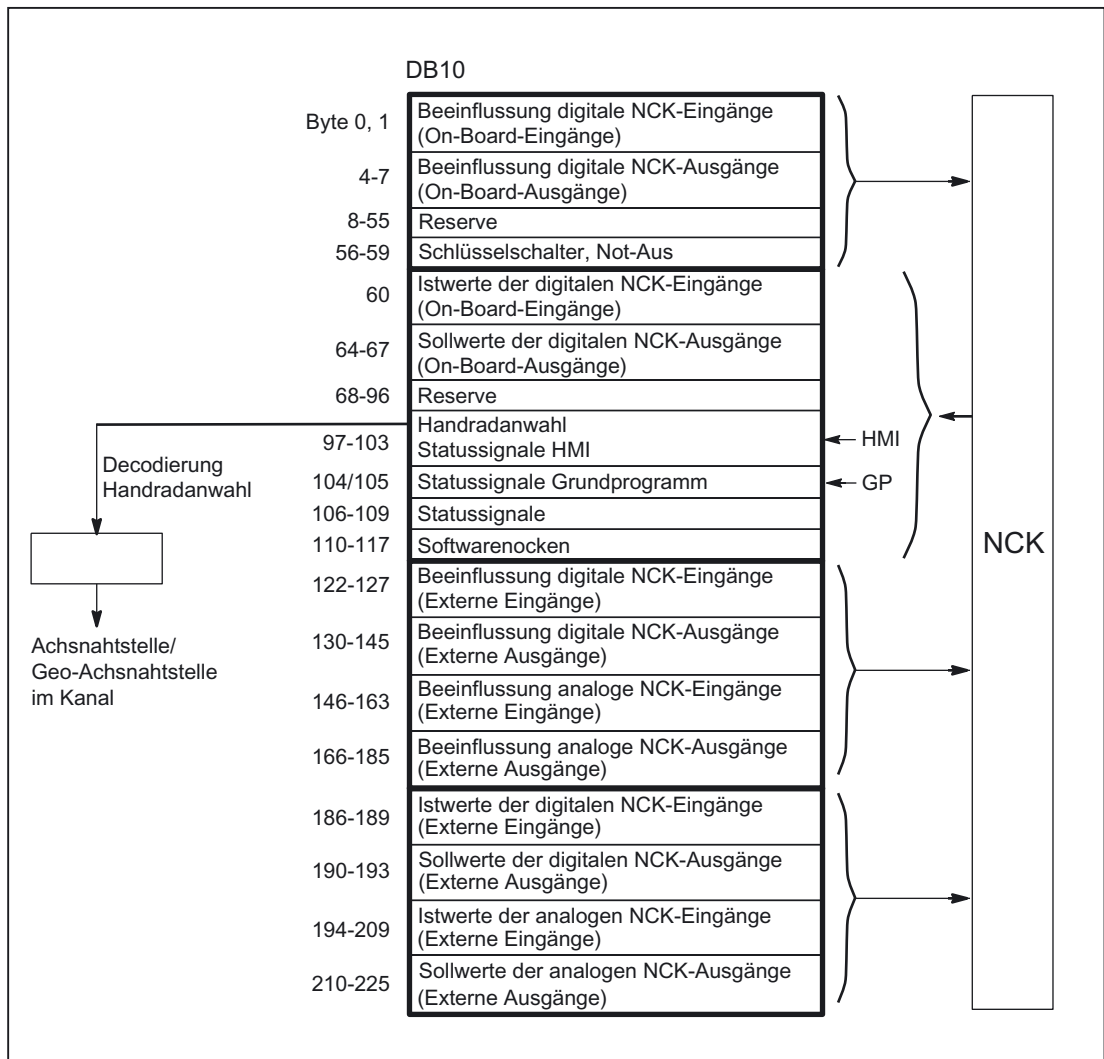


Bild 2-4 Nahtstelle PLC/NC

## Digitale /Analoge Ein-/Ausgänge des NCK

Bei den digitalen und analogen Ein- und Ausgängen des NCK ist Folgendes zu beachten:

### Eingänge:

- Alle Eingangssignale bzw. Eingangswerte des NCK werden auch an die PLC übergeben.
- Die Weitergabe der Signale an das NC-Teileprogramm kann von der PLC unterdrückt werden. Stattdessen kann ein Signal bzw. Wert von der PLC vorgegeben werden.
- Es kann auch dann ein Signal bzw. Wert von der PLC an den NCK übergeben werden, wenn auf NCK-Seite für diesen Kanal keine Hardware vorhanden ist.

### Ausgänge:

- Alle auszugebenden Signale bzw. Werte werden auch an die PLC übergeben.
- Es können auch dann Signale bzw. Werte vom NCK an die PLC übergeben werden, wenn auf NCK-Seite für diesen Kanal keine Hardware vorhanden ist.
- Die vom NCK vorgegebenen Werte können von der PLC überschrieben werden.
- Von der PLC aus können auch direkt Signale bzw. Werte über die NCK-Peripherie ausgegeben werden.

---

### Hinweis

Bei der Realisierung von digitaler und analoger NCK-Peripherie sind die Informationen in folgender Literatur zu berücksichtigen:

### Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen;  
Digitale und analoge NCK-Peripherie (A4)

---

## Signale PLC/BAG

Zur Betriebsartengruppe (BAG) des NCK werden die von der Maschinensteuertafel oder von der HMI-Software vorgegebenen Betriebsartensignale übertragen. Diese gelten bei der 810D und bei der 840D für alle NC-Kanäle der BAG. Im NCK können bei 840D-Systemen optional mehrere BAGs definiert sein.

Von der BAG wird deren aktueller Zustand an die PLC gemeldet.

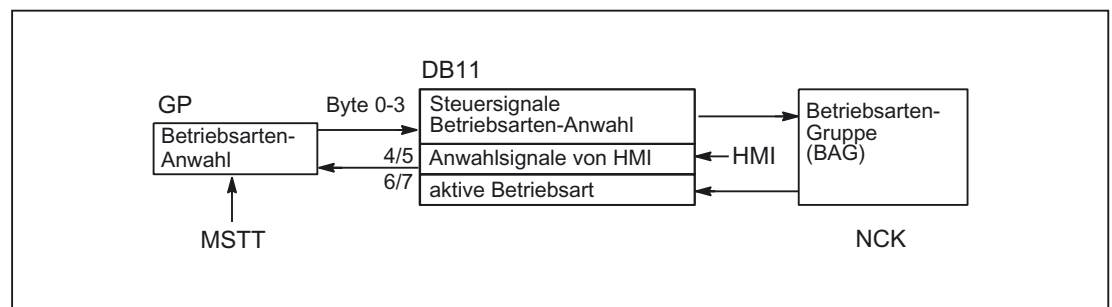


Bild 2-5 Signale PLC/BAG (z.B. BAG1)

## Signale PLC/NCK-Kanäle

Bei der Nahtstelle sind folgende Signalgruppen zu betrachten:

- Steuer-/Status-Signale
- Hilfs-/G-Funktionen
- Signale der Werkzeugverwaltung
- NCK-Funktionen.

Die Steuer-/Status-Signale werden zyklisch am Anfang von OB1 übertragen. Auch die von der HMI-Software in die kanalspezifische Nahtstelle eingetragenen Signale (der Eintrag der HMI-Signale erfolgt durch das Betriebssystem der PLC) werden zu diesem Zeitpunkt übertragen, wenn diese Signale nicht über die MSTT, sondern über die Bedientafelfront vorgegeben werden.

Die **Hilfs- und G-Funktionen** werden auf zweierlei Weise in die Nahtstellen-DBs eingetragen. Zunächst werden sie zusammen mit den Änderungssignalen eingetragen.

- Die **M-Signale** M00 - M99 \* (sie werden vom NCK mit der erweiterten Adresse 0 übergeben) werden zusätzlich decodiert und die zugehörigen Nahtstellenbits für eine Zyklusdauer gesetzt.

\* Die M-Signale M0 - M99 werden vom NCK mit der erweiterten Adresse 0 übergeben

- Bei den **G-Funktionen** wird zusätzlich die Gruppe decodiert und die in der jeweiligen Gruppe aktive G-Funktion in den Nahtstellen-DB eingetragen.
- Die **S-Werte** werden zusätzlich zusammen mit den zugehörigen M-Signalen (M03, M04, M05) in die spindelspezifische Nahtstelle eingetragen. Ebenso werden achsspezifische Vorschübe in die entsprechende achsspezifische Nahtstelle eingetragen.

Bei aktivierter **Werkzeugverwaltung** im NCK wird die Belegung von Spindel bzw. Revolver und der Be-/Entladestellen in eigene Nahtstellen-DBs eingetragen (DB71 - 73).

Die **NCK-Funktionen** werden über PLC-Funktionsaufrufe angestoßen und parametrierbar. Es stehen z.B. folgende Funktionsaufrufe zur Verfügung:

- Positionieren einer Linear- bzw. Rundachse
- Positionieren einer Teilungsachse
- Starten eines vorbereiteten asynchronen Unterprogramms (ASUP)
- Lesen/schreiben von NC-Variablen
- Magazinbelegung aktualisieren

Die vorgenannten Funktionen sind zum Teil in eigenen Funktionsdokumentationen beschrieben.



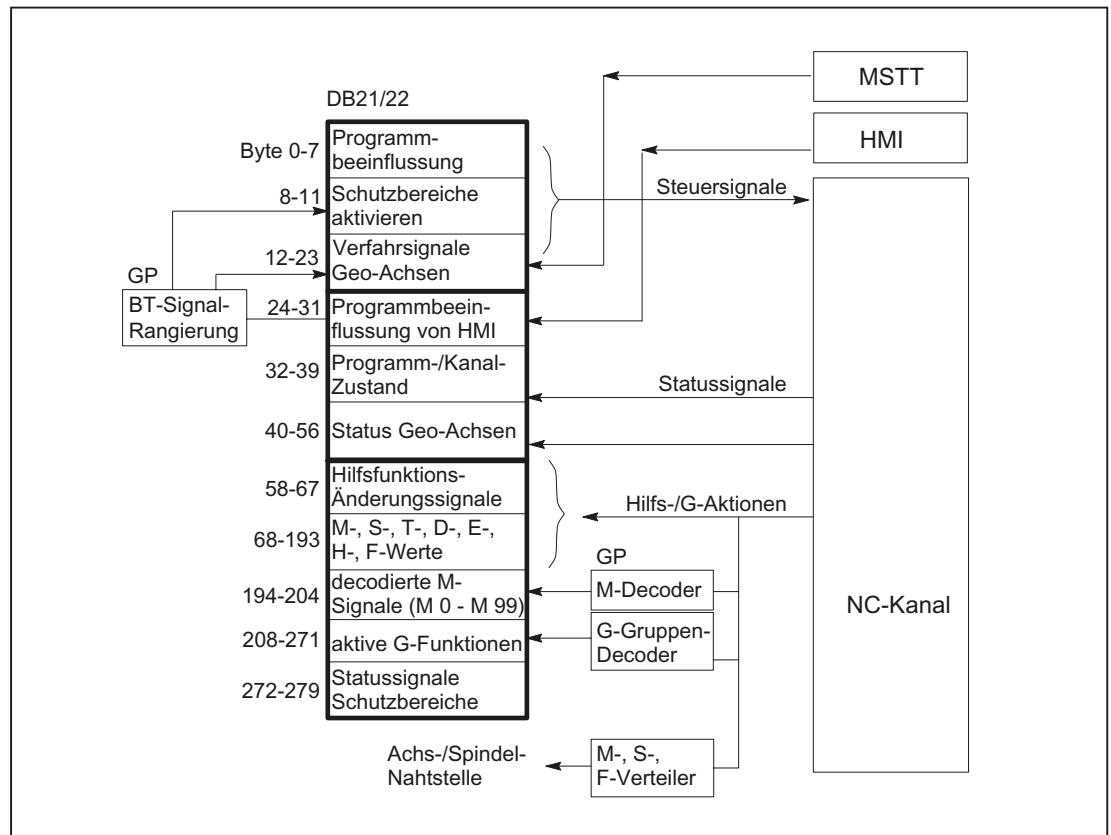


Bild 2-6 Nahtstelle PLC/NC-Kanal

### Signale PLC/Achsen, Spindel, Antrieb

Die achs- und spindelspezifischen Signale sind in folgende Gruppen aufgeteilt:

- Gemeinsame Achs-/Spindelsignale
- Achssignale
- Spindelsignale
- Antriebssignale

Die Signale werden bis auf die im Folgenden beschriebenen Ausnahmen zyklisch am Anfang des OB1 übertragen.

Zu den Ausnahmen gehören:

#### **INC-Mode von HMI, Axialer F-Wert, M-/S-Wert.**

Ein axialer F-Wert wird über den M-, S-, F-Verteiler des Grundprogramms dann eingetragen, wenn er im Zuge der NC-Programmbearbeitung an die PLC übertragen wird.

M- und S-Wert werden ebenfalls dann über den M-, S-, F-Verteiler des Grundprogramms eingetragen, wenn ein S-Wert zusammen mit dem zugehörigen M-Wert (M03, M04, M05) zur Abarbeitung kommt.

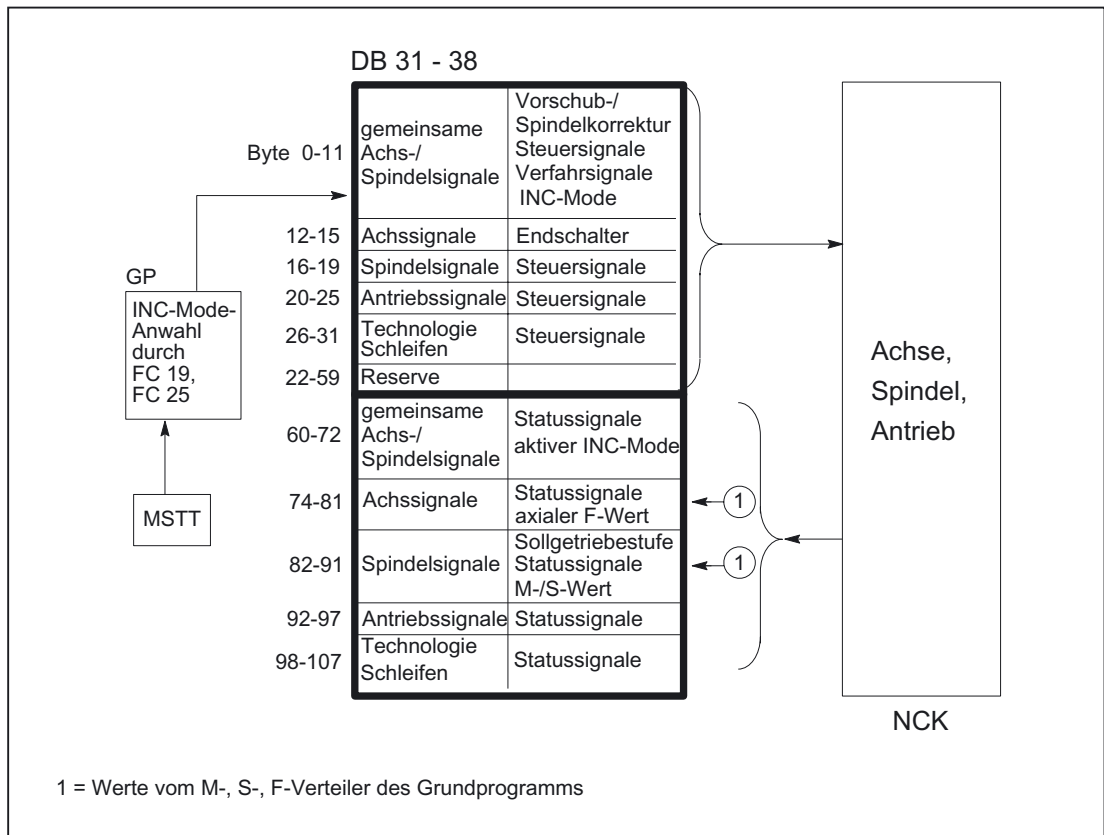


Bild 2-7 Nahtstelle PLC/Achsen, Spindeln, Antriebe

### 2.6.3 Nahtstelle PLC/HMI

#### Allgemeines

Bei der Nahtstelle PLC/HMI müssen folgende Funktionskomplexe betrachtet werden:

- Steuersignale
- Maschinenbedienung
- PLC-Meldungen
- PLC-Status-Anzeige

#### Steuersignale

Bei den Steuersignalen handelt es sich um Signale, die u.a. von der Maschinensteuertafel vorgegeben werden und von der HMI-Software berücksichtigt werden müssen. Zu diesen Signalen gehören z.B. Istwerte im MKS oder WKS anzeigen, Tastensperre usw. Diese Signale werden über einen eigenen HMI-Nahtstellen-DB (DB19) ausgetauscht.

## Maschinenbedienung

Alle Bedienhandlungen, die zu Aktionen an der Maschine führen, werden von der PLC kontrolliert. Im Normalfall werden diese über die Maschinensteuertafel (MSTT) vorgenommen. Es ist jedoch auch möglich, einen Teil der Bedienhandlungen wahlweise von der Bedienoberfläche aus vorzunehmen (wie z.B. Betriebsartenwahl, INC-Mode-Anwahl).

Die von HMI kommenden Bediensignale werden vom PLC-Betriebssystem direkt in die Nahtstellen-DBs eingetragen. Im Standardfall werden diese Bediensignale so vom Grundprogramm rangiert, dass - sofern eine gleichartige Bedienmöglichkeit von der MSTT besteht - die Bedienung wahlweise von der Bedienoberfläche oder von der MSTT vorgenommen werden kann. Bei Bedarf kann die Bedienung über HMI durch einen Parameter "MMCToIF" des FB1 vom Anwender ausgeschaltet werden.

## PLC-Meldungen

Basis der Meldefunktionen sind die in das Betriebssystem der AS 300 integrierten Systemdiagnose-Funktionen, die folgende Eigenschaften besitzen:

- In einer **Diagnose-Zustandsliste** werden vom PLC-Betriebssystem alle wichtigen Systemzustände sowie Zustandsübergänge eingetragen. Weiterhin werden Kommunikationsereignisse und (Peripherie-) Baugruppen-Diagnosedaten (bei Baugruppen mit Diagnosefunktionen) eingetragen.
- Zusätzlich werden die Diagnose-Ereignisse, die zum Systemstop führen, in einen **Diagnosepuffer** (Ringpuffer) in der zeitlichen Reihenfolge des Auftretens mit Uhrzeitstempel eingetragen.
- Die in den Diagnosepuffer eingetragenen Ereignisse werden automatisch an Bedien-/Beobachtungssysteme über die MPI bzw. auch über die BTSS durch das COM-Modul versandt, sofern diese sich als empfangsbereit angemeldet haben (Meldedienst). Die Übertragung an den angemeldeten Teilnehmer ist eine Funktion des Betriebssystems der PLC. Der Empfang und die Interpretation der Meldungen werden durch die HMI-Software sichergestellt.
- Vom PLC-Anwenderprogramm können über einen SFC (System-Function-Call) ebenfalls Meldungen in den Diagnosepuffer eingetragen werden.
- Die Ereignisse werden codiert in den Diagnosepuffer eingetragen. Die zugehörigen Texte müssen in der Bedientafel vorgehalten werden.

Zusammen mit dem Grundprogramm wird ein FC zur Meldeerfassung (FC10) bereitgestellt, der die zu meldenden Ereignisse - in Signalgruppen eingeteilt - erfasst und über den Diagnosepuffer der HMI meldet.

Die Struktur der Meldeerfassung ist in Bild "Erfassung und Meldung von PLC-Ereignissen" dargestellt.

Sie weist folgende Merkmale auf:

- Die Bitfelder für Ereignisse, die die VDI-Schnittstelle betreffen, sind zusammen mit den Bitfeldern für die Anwendermeldungen in dem DB2 zusammengefasst.
- Die Bitfelder werden mehrfach durch den FC10 ausgewertet.

### – Auswertung 1; Erfassung von **Sammelsignalen**

Je Signalgruppe wird ein Sammelsignal erzeugt, wenn mindestens ein Bit Signal "1" führt. Dieses Signal wird im Normalfall auf das zugehörige Sperrsignal der VDI-Nahtstelle (bei Baugruppen mit Diagnosefunktionen) geführt. Die Sammelsignale werden zyklisch komplett erfasst.

- **Auswertung 2;**Erfassung von
- **Fehlermeldungen**  
Es gibt eine feste Vorgabe, welche Signale einer Gruppe bei ihrem Wechsel von "0" nach "1" eine Fehlermeldung erzeugen.
- **Auswertung 3;**Erfassung von
- **Betriebsmeldungen**  
Es gibt eine feste Vorgabe, welche Signale einer Gruppe eine Betriebsmeldung erzeugen.
- Der Umfang der Anwender-Bitfelder (Bereich User) ist standardmäßig auf 10 Bereiche zu je 8 Byte festgelegt, kann jedoch über Grundprogramm-Parameter am FB1 auf die Bedürfnisse des Maschinenherstellers angepasst werden.

## Quittungskonzept

Bei den Fehler- und Betriebsmeldungen werden folgende Quittungskonzepte verfolgt:

Die **Betriebsmeldungen** zeichnen sich dadurch aus, dass mit ihnen normale Betriebszustände der Maschine als Information für den Bediener angezeigt werden sollen. Deshalb entfällt bei ihnen die Notwendigkeit von Quittungssignalen. Bei ihnen wird sowohl das Kommen als auch das Gehen des Ereignisses erfasst und ein Eintrag in den Diagnosepuffer vorgenommen. Die HMI-Software erzeugt anhand der Kennungen "BM gekommen" und "BM gegangen" ein aktuelles Abbild der anstehenden Betriebsmeldungen.

Mit **Fehlermeldungen** werden Fehlerzustände an der Maschine angezeigt, die im Normalfall zum Maschinenstillstand führen. Treten mehrere Fehler "gleichzeitig" auf, ist es für die Fehlersuche wichtig, die Reihenfolge des Auftretens zu kennen. Dies wird einerseits durch die Reihenfolge des Eintrags in den Diagnosepuffer angezeigt und andererseits durch den Uhrzeitstempel, den jeder Eintrag erhält.

Verschwindet die Fehlerursache, so wird die zugehörige Fehlermeldung erst dann gelöscht, wenn eine Anwenderquittung vorliegt (z.B. Kundentaste an der MSTT). Als Reaktion auf dieses Signal untersucht der FC "Meldungserfassung", welche der bereits gemeldeten Fehler verschwunden sind und trägt diese mit der Kennung "Fehler gegangen" in den Diagnosepuffer ein. Somit kann die HMI-Software auch bei den Fehlermeldungen ein aktuelles Abbild der anstehenden Meldungen erzeugen. Bei den noch anstehenden Meldungen bleibt die Uhrzeit ihres Entstehens erhalten (im Gegensatz zu einer Neuabfrage).

## Anwenderprogramm

Das Anwender-PLC-Programm muss im zyklischen Programmteil lediglich den Grundprogrammbaustein FC10 mit entsprechender Parametrierung aufrufen und die Bitfelder im DB2 setzen bzw. rücksetzen. Alles Weitere wird durch das Grundprogramm und der HMI-Software erledigt.

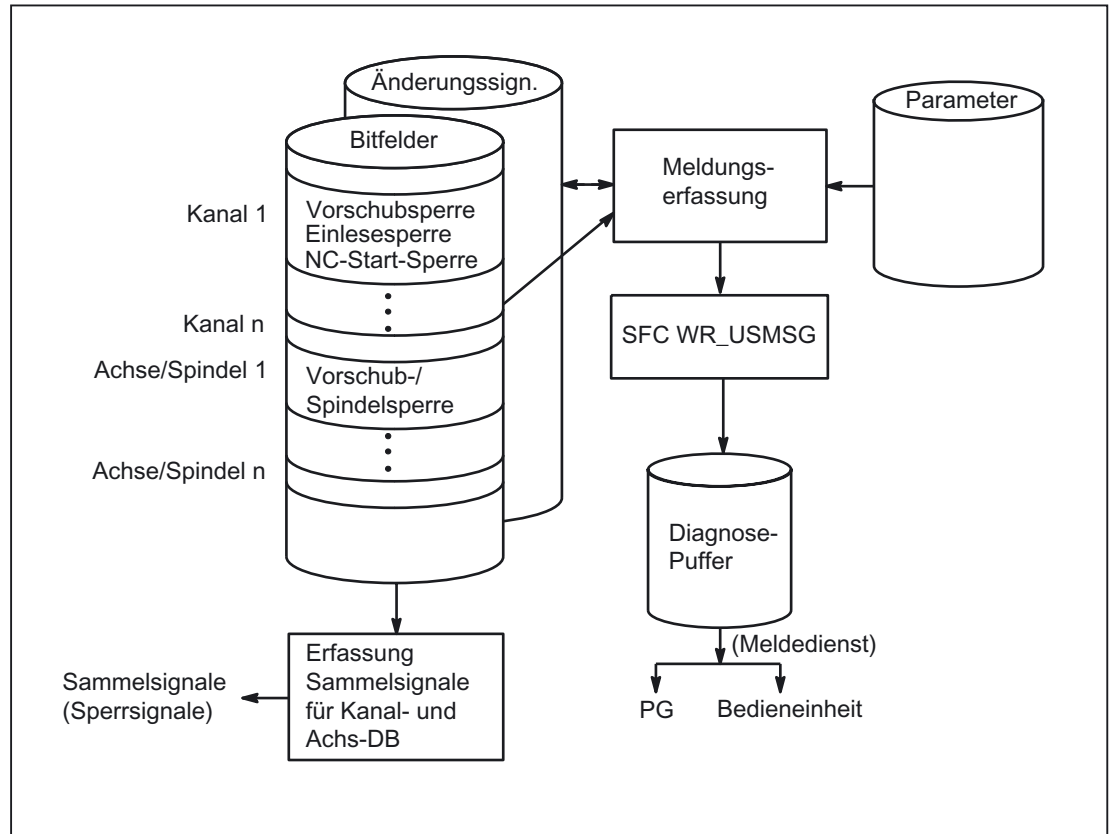


Bild 2-8 Erfassung und Meldung von PLC-Ereignissen

## 2.6.4 Nahtstelle PLC/MSTT/BHG

### Allgemeines

Die Maschinensteuertafel (MSTT) ist bei SINUMERIK 840D, 810D über den Bus angekoppelt, der auch das OP mit der NC verbindet. Dies hat den Vorteil, dass nur eine Busleitung zur Bedieneinheit verlegt werden muss. Das Bedienhandgerät (BHG) kann sowohl an der MPI-Schnittstelle der PLC angekoppelt werden als auch an der BTSS-Schnittstelle (nur 840D). Da der OP-Bus bei 840D eine höhere Baudrate erlaubt, ergeben sich somit zwei verschiedene Bustopologien.

**Topologie 840D**

Bei 840D wird die Maschinensteuertafel an das BTSS-Busselement (Übertragungsrate 1,5 MBaud) als aktiver Globaldaten-Teilnehmer angeschlossen. Sollen weitere Tasten und Anzeigen für Kunden-Bedientafelfront angeschlossen werden, so kann dies über ein weiteres Tastatur-Interface (MSTT ohne Bedienteil) erfolgen. Je Tastatur-Interface können über Flachbandkabel 64 Taster, Schalter etc. und 64 Anzeigeelemente angeschlossen werden.

Die von der MSTT kommenden Signale werden vom COM-Modul ins DPR (Dual-Port-RAM) zur NC kopiert. Die NC wiederum übergibt sie zur PLC (VDI-Task). Das Grundprogramm der PLC trägt die ankommenden Signale in das Eingangsabbild ein. Im Normalfall werden die NC-relevanten Signale vom Grundprogramm auf die VDI-Nahtstelle verteilt. Bei Bedarf kann dies vom Anwender modifiziert werden.

Die Signale von der PLC zur MSTT (Anzeigen) nehmen den umgekehrten Weg.

Die Signale des Bedienhandgerätes (BHG) werden entweder über die BTSS nach dem gleichen Verfahren wie bei der MSTT oder über den Dienst GD (Globale Daten) der MPI-Schnittstelle übertragen. Das PLC-Betriebssystem trägt die Daten des BHG z.B. ins Eingangsabbild ein und überträgt umgekehrt die Anzeigewerte z.B. vom Ausgangsabbild zum BHG. Die entsprechenden Parameter werden über den Systemdatenbaustein SDB 210 vorgegeben, der mit dem STEP 7-Werkzeug Communication-Configuration erzeugt wird.

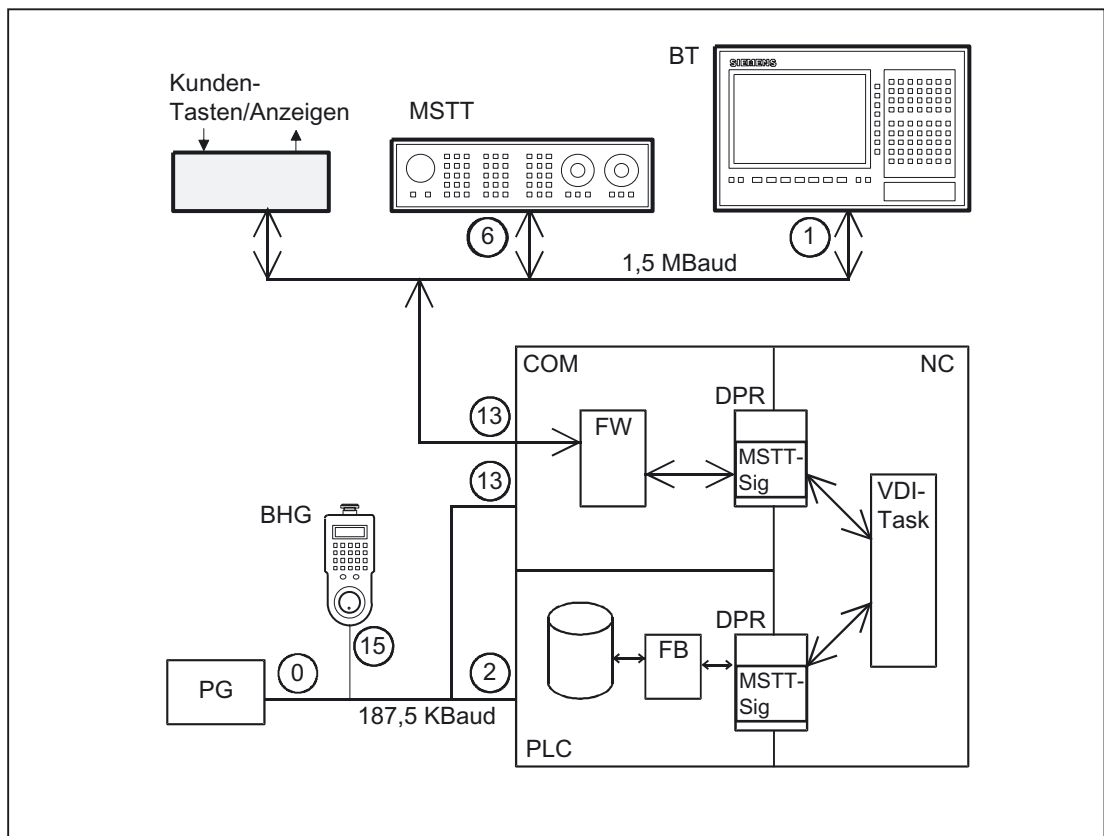


Bild 2-9 Ankopplung der Maschinensteuertafel bei 810D

## Topologie 810D

Bei 810D werden Maschinensteuertafel, Bedienhandgerät und OP an die MPI (Multipoint-Interface) der AS 300 angeschlossen. Die Übertragungsrate beträgt bei dieser Konfiguration 187,5 kBaud. Die ankommenden Signale werden am Zykluskontrollpunkt vom PLC-Betriebssystem direkt in die Anwender-Nahtstelle kopiert (z.B. Eingangs-Abbild). Der Transfer zur VDI-Nahtstelle wird ebenso wie bei 840D durch das Anwenderprogramm bzw. durch eine Standardroutine des Grundprogramms vorgenommen.

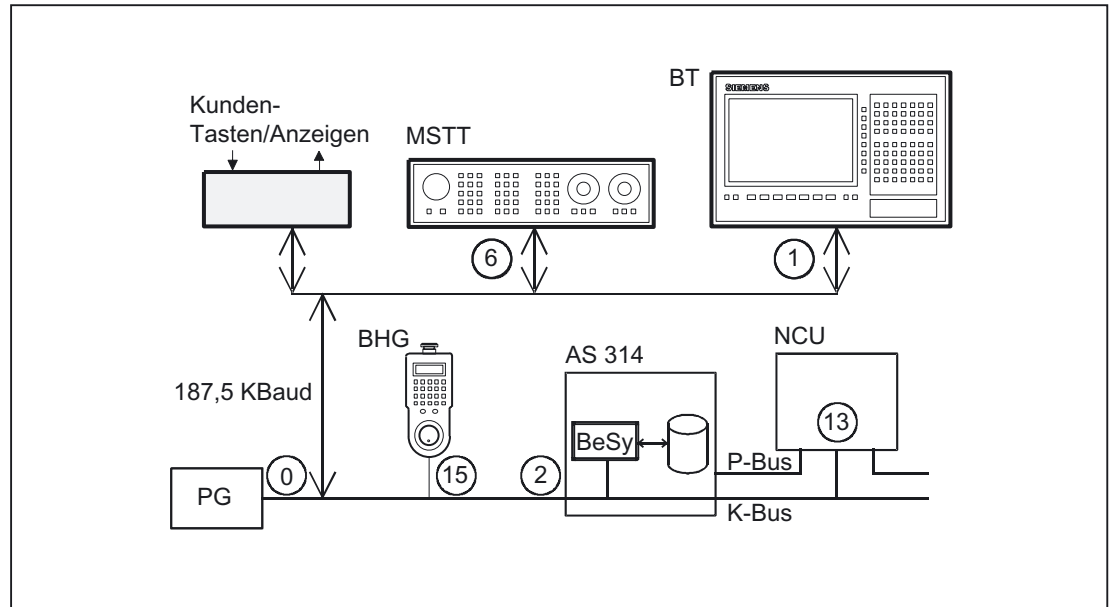


Bild 2-10 Ankopplung der Maschinensteuertafel bei 810D

## Busadressen

In den Bildern "Ankopplung der Maschinensteuertafel bei 810D" sind die Default-Busadressen für die Standardkonfigurationen eingetragen. Neben den Busadressen muss für den Dienst GD (Globale Daten) eine GD-Kreis-Nr. definiert werden.

Bei der Vergabe der Busadressen (Teilnehmer-Nr.) ist Folgendes zu beachten:

### Busadressen 840D

Bei 840D müssen die beiden Bussegmente getrennt betrachtet werden:

Bedientafelfront-Bussegment:		
Busteilnehmer	zul. Einstellbereich	Standard-Einstellung
Bedientafelfront (OP)	1 - 31	1
MSTT/Tastatur-Interface	15	(Einstellung über DIP-Fix)
COM-Modul	31	13
PG/PC (z.B. für IBS)	fest	0

PLC-Bussegment:			
Busteilnehmer	Einstellbereich	Standard-Einstellung	Anmerkung
PLC	31	2	
COM-Modul	abhängig von PLC-Adresse fest	3	
PG/PC (z.B. für IBS)	fest	0	

## MSTT-Nahtstelle in der PLC

Die Signale der Maschinensteuertafel werden standardmäßig über den E-/A-Bereich geführt. Es ist dabei zwischen den NC- und den maschinenspezifischen Signalen zu unterscheiden. Die NC-spezifischen Tastensignale werden standardmäßig vom FC 19 auf die jeweilige BAG-, NCK-, Achs- und Spindel-spezifische Nahtstelle verteilt. Umgekehrt werden die zugehörigen Statussignale auf die MSTT-Nahtstelle rangiert.

---

### Hinweis

Der FC 19 muss im PLC-Anwenderprogramm aufgerufen werden.

---

Die Kundentasten, mit denen diverse Maschinenfunktionen ausgelöst werden können, müssen direkt vom Anwenderprogramm ausgewertet werden. Ebenso sind von diesem auch die zugehörigen Statussignale auf den Ausgabebereich für die LEDs zu rangieren.



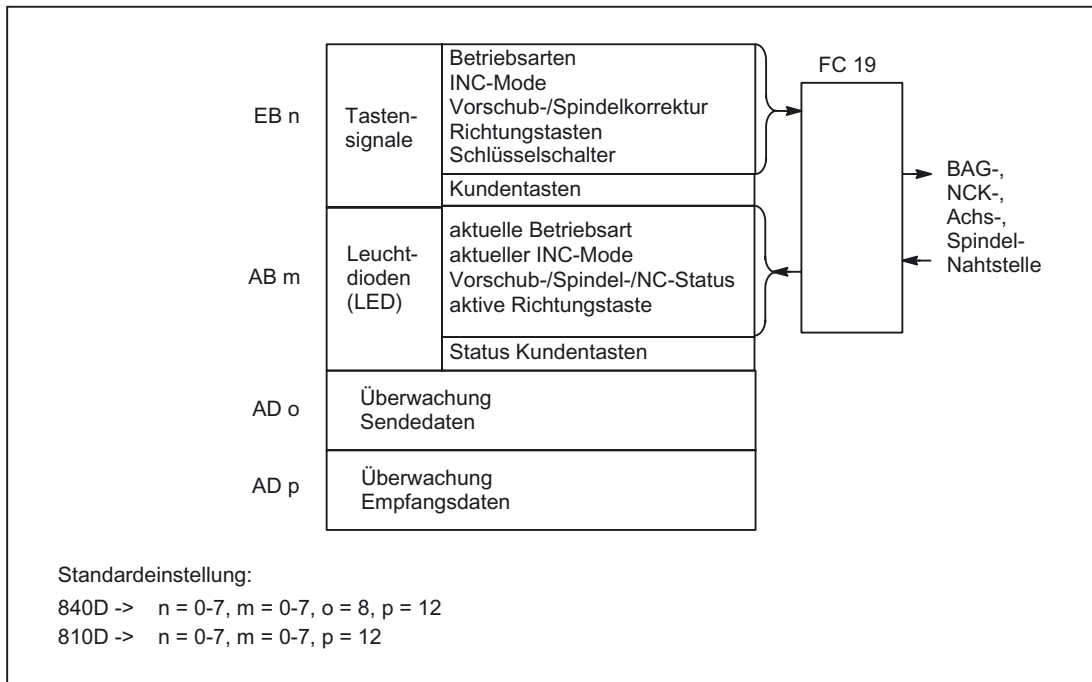


Bild 2-11 Nahtstelle von und zur Maschinensteuertafel

## 2.7 Struktur und Funktionen des Grundprogramms

### 2.7.1 Allgemein

#### Allgemeines

Das Programm ist modular aufgebaut, d.h. es ist nach NCK-Funktionen strukturiert.

Im Betriebssystem wird unterschieden:

- Anlauf und Synchronisation (OB 100)
- zyklischer Betrieb (OB 1)
- Prozessalarm-Bearbeitung (OB 40)

Im OBs 1, 40 und 100 muss der jeweilige Teil des Grundprogramms - wie im folgenden Bild dargestellt - vom Anwender aufgerufen werden.

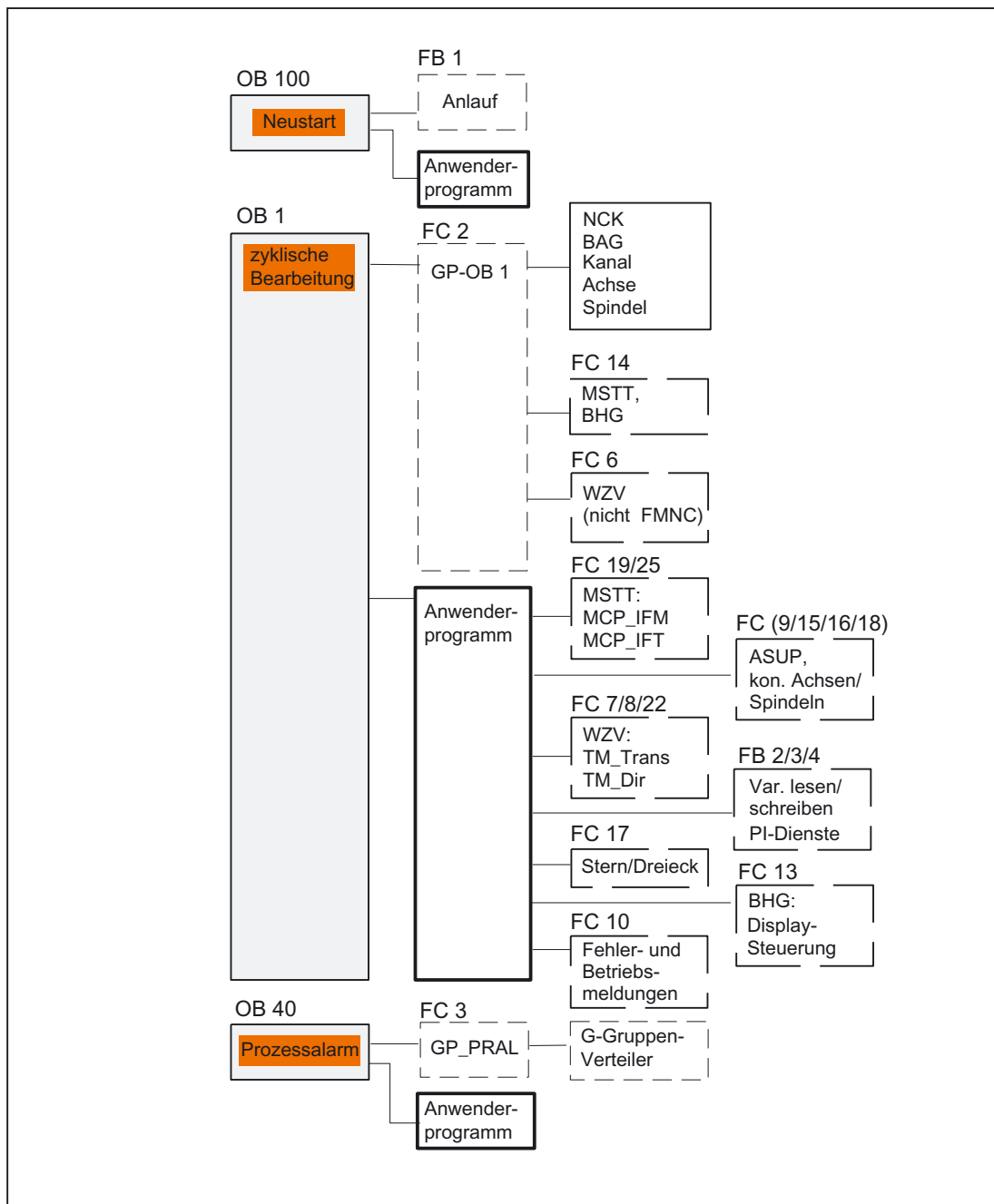


Bild 2-12 Struktur des PLC-Programmes

## 2.7.2 Anlauf und Synchronisation NCK-PLC

### Laden des Grundprogrammes

Das Laden des Grundprogrammes mit dem S7 Tool muss im Stopp-Zustand der PLC erfolgen. Es wird so sichergestellt, dass alle Bausteine des Grundprogrammes beim nächsten Anlauf richtig initialisiert werden. Es kann sonst zu undefinierten Zuständen der PLC kommen. (z.B. Blinken aller PLC-LEDs).

### Anlauf

Im Anlauf erfolgt die Synchronisation zwischen NCK und PLC. Es werden die System- und Anwenderdatenbausteine auf Vollständigkeit und die wichtigsten GP-Parameter auf Plausibilität geprüft. Im Fehlerfall übergibt das Grundprogramm eine Fehlerkennung in den Diagnosepuffer und setzt die PLC in den STOP-Zustand.

Das Betriebssystem durchläuft nach der Systeminitialisierung den Organisationsbaustein OB 100 und beginnt den zyklischen Betrieb immer am Anfang des OB 1.

### Synchronisation

Beim Hochlauf synchronisiert sich die PLC mit HMI und NCK.

### Lebenszeichen

Nach ordnungsgemäßigem Anlauf und erstem vollständigen OB1-Zyklus (Grundstellungszyklus) tauschen PLC und NCK fortlaufend Lebenszeichen aus. Bleibt das Lebenszeichen des NCK aus, wird die Nahtstelle PLC/NCK neutralisiert und das Signal "NCK-CPU-ready" in DB 10 auf Null gesetzt.

## 2.7.3 Zyklischer Betrieb (OB1)

### Allgemeines

Die komplette Bearbeitung der NCK-PLC-Schnittstelle erfolgt ausschließlich im zyklischen Betrieb. Das Grundprogramm läuft - zeitlich gesehen - vor der Bearbeitung des Anwenderprogramms. Um die Grundprogrammlaufzeit gering zu halten, werden nur die Steuer-/ Statussignale zyklisch übertragen, die Hilfs- und G-Funktionsübergabe wird nur auf Anforderung bearbeitet.

Folgende Funktionen werden im zyklischen Teil des Grundprogramms ausgeführt:

- Übertragung der Steuer-/Statussignale,
- Verteilung der Hilfs- und G-Funktionen,
- M-Decodierung (M00 - M99),
- M-, S-, F-Verteilung,
- Übertragung der MSTT-Signale über NCK (nur bei 840D),
- Erfassung und Aufbereitung der Anwender-Fehler- und Betriebsmeldungen.

### Steuer-/ Statussignale

Gemeinsames Merkmal der Steuer- und Statussignale ist, dass es sich um Bitfelder handelt. Das Grundprogramm aktualisiert sie am Anfang des OB 1.

Die Signale sind in folgende Gruppen eingeteilt:

- Allgemeine Signale
- BAG-spezifische Signale wie z.B. Betriebsarten
- Kanalspezifische Signale wie z.B. Programm- und Vorschub-Beeinflussung

Achs- und Spindel-spezifische Signale wie z.B. Vorschub-Sperre

### Hilfs- und G-Funktionen

Die Hilfs- und G-Funktionen weisen folgende Merkmale auf:

- Sie werden jeweils satzsynchron zur PLC übertragen (bezogen auf einen Teileprogrammsatz).
- Die Übergabe erfolgt quittungsgesteuert.
- Die Quittungszeiten beeinflussen unmittelbar die Satzausführungszeit von NC-Sätzen mit quittierungspflichtigen Hilfsfunktionen.

Der Wertebereich ist in folgender Tabelle dargestellt:

Funktion	Struktur		Wertebereich		Datentyp	
	1. Wert	2. Wert	1. Wert	2. Wert	1. Wert	2. Wert
G-Funktion		G-Funktion		255 <sup>1)</sup>		Byte
M-Wort	M-Gruppe	M-Wort	99	99.999.999	Word	DWord
S-Wort	Spindel-Nr.	S-Wort	6	Gleitpunkt <sup>2)</sup>	Word	DWord
T-Wort	Magazin-Nr.	T-Wort	99	65535	Word	Word
D-Wort	-	D-Wort	99	255	Byte	Byte
H-Wort	H-Gruppe	H-Wort	99	Gleitpunkt	Word	DWord
F-Wort	Achs-Nr.	F-Wort	18	Gleitpunkt	Word	DWord

<sup>1)</sup> relative Nummer, die je G-Gruppe übergeben wird

<sup>2)</sup> entsprechend STEP 7-Format (24 Bit Mantisse, 8 Bit Exponent)

Über den Hilfs-/G-Funktionsverteiler werden die vom NCK übergebenen M-, S-, T-, H-, D-, F-Werte zusammen mit den zugehörigen Änderungssignalen auf die Nahtstelle **KANAL-DB** (siehe Dokumentation Listen der SINUMERIK 840D, 810D) ausgegeben. Dabei werden jeweils die beiden Werte der Hilfsfunktion in die entsprechenden Datenworte übertragen. Das dazugehörige Änderungssignal wird für einen PLC-Zyklus auf 1 aktiv geschaltet. Mit dem Rücksetzen des Änderungssignals erfolgt die Quittierung an den NCK. Die Quittierung von schnellen Hilfsfunktionen erfolgt gleich nachdem das Grundprogramm die Hilfsfunktion erkannt hat durch das Grundprogramm.

Zusätzlich zur Verteilung der Hilfs- und G-Funktionen erfolgt für ausgewählte Signale eine weitere Aufbereitung wie im Folgenden ausgeführt.

### M-Decoder

Mit M-Funktionen können sowohl Schaltbefehle als auch Festpunkt-Werte übergeben werden. Für die Standard-M-Funktionen (Bereich M00 - M99) werden ausdecodierte dynamische Signale auf die Nahtstelle **KANAL-DB** ausgegeben (Signaldauer = 1 Zykluszeit).

### G-Gruppen-Decoder

Bei den vom NCK übergebenen G-Funktionen werden die zugehörigen Gruppen ausdecodiert und die jeweils aktuelle G-Nummer in das entsprechende Nahtstellenbyte des KANAL-DB eingetragen, d.h. in den Kanal-DBs sind alle aktiven G-Funktionen eingetragen. Die eingetragenen G-Funktionen bleiben auch nach NC-Programm-Ende oder Programm-Abbruch erhalten.

---

#### Hinweis

Bei Systemanlauf wird in alle G-Gruppenbytes der Wert "0" eingetragen.

---

### M-, S-, F-Verteiler

Mit dem M-, S-, F-Verteiler werden spindelspezifische M-Worte  $M(1..6)=[3,4,5]$ , S-Worte und F-Worte für axiale Vorschübe in die zugehörigen **Spindel- und Achs-DBs** eingetragen. Kriterium für die Verteilung ist die erweiterte Adresse, die bei M-Worten, S-Worten und axialen F-Worten an die PLC übergeben wird.

### MSTT-Signal-Übertragung

Bei 840D werden die MSTT-Signale über den seriellen Bus (MPI) zum NCK übertragen und von diesem an die PLC übergeben. Ein FC des Grundprogramms überträgt die Signale an die durch Grundprogramm-Parameter festgelegte Nahtstelle von Ein- und Ausgängen. Die Statussignale zur Ansteuerung der LEDs an der MSTT nehmen den umgekehrten Weg.

### Anwendermeldungen

Die Erfassung und Aufbereitung der Anwender-Fehler- und Betriebsmeldungen erfolgt durch einen FC des Grundprogrammes.

## 2.7.4 Zeitalarm Bearbeitung (OB 35)

### Allgemeines

Für die Zeitalarmbearbeitung ist der **OB 35** vom Anwender zu programmieren. Das Zeitraster des **OB 35** ist im Standard auf 100 ms eingestellt. Durch das STEP7-Tool "S7-Konfiguration" kann jedoch ein anderes Zeitraster gewählt werden. Allerdings darf eine Zeiteinstellung des OB 35 nicht kleiner als etwa 15 ms ohne zusätzliche Maßnahmen verwendet werden, da es sonst zum Stopp der PLC-CPU kommt. Der Stopp wird verursacht durch das Lesen der Systemzustandsliste (SZL) beim HMI-Hochlauf. Dieses Lesen bedeutet eine Blockierung der Ablaufebenen-Steuerung (AES) für ca. 8 bis 12 ms. Dadurch wird der OB 35 bei kleineren Zeiteinstellungen nicht mehr korrekt bearbeitet. Sind jedoch kleinere Zeiteinstellungen für den OB 35 notwendig, dann kann durch die Programmierung des OB 80 mit mindestens dem Programmbefehl "BE" der Stopp verhindert werden.

## 2.7.5 Prozessalarm Bearbeitung (OB 40)

### Allgemeines

Ein Prozessalarm **OB 40** (Interrupt) kann z.B. durch entsprechend projektierte Peripherie oder durch bestimmte NC-Funktionen ausgelöst werden. Wegen der unterschiedlichen Herkunft des Interrupts muss das PLC-Anwenderprogramm im OB 40 zuerst die Interruptursache interpretieren. Die Interruptursache ist in den Lokaldaten des OB 40 enthalten.

(Siehe hierzu auch SIMATIC STEP 7 Beschreibung oder Online-Help von STEP 7).

## 2.7.6 Verhalten bei NC-Ausfall

### Allgemeines

Während des zyklischen Betriebs erfolgt eine ständige Überwachung der NC-Bereitschaft durch das PLC-Grundprogramm mittels Abfrage eines Lebenszeichens. Reagiert der NCK nicht mehr, so wird die Nahtstelle NCK/PLC neutralisiert und das NST **NCK-CPU-Ready** im Bereich der **Signale von NC (DB 10.DBX 104.7)** zurückgesetzt. Ferner werden die Signale, die vom NCK an die PLC übergeben werden, in einen Grundzustand gesetzt.

Die PLC selbst bleibt aktiv, so dass weiterhin Maschinenfunktionen von ihr gesteuert werden können.

## Signale NCK ! PLC

Bei den Signalen, die von NCK an PLC übergeben werden, müssen folgende Gruppen unterschieden werden:

- Statussignale von NCK, Kanälen, Achsen und Spindeln
- Änderungssignale der Hilfsfunktionen
- Werte der Hilfsfunktionen
- Werte der G-Funktionen

### **Statussignale:**

Die Statussignale von NCK, den Kanälen, Achsen und Spindeln werden zurückgesetzt.

### **Änderungssignale der Hilfsfunktionen:**

Die Änderungssignale der Hilfsfunktionen werden ebenfalls zurückgesetzt.

### **Werte der Hilfsfunktionen:**

Die Werte der Hilfsfunktionen bleiben erhalten, so dass rekonstruierbar ist, welche Funktionen als letzte vom NCK angestoßen wurden.

### **Werte der G-Funktionen:**

Die Werte der G-Funktionen werden zurückgesetzt (d.h. es wird jeweils der Wert 0 eingetragen).

## Signale PLC → NCK

Bei den Signalen, die von der PLC an den NCK übergeben werden, wird zwischen Steuersignalen und Aufträgen, die per FCs an den NCK übergeben werden, unterschieden.

### **Steuersignale:**

Die Steuersignale von PLC an NCK werden eingefroren; die zyklische Aktualisierung durch das PLC-Grundprogramm wird unterbrochen.

### **Aufträge von PLC an NCK:**

Die FCs und FBs, mit denen Aufträge an den NCK übergeben werden, dürfen durch das PLC-Anwenderprogramm nicht mehr bearbeitet werden, da unter Umständen fehlerhafte Rückmeldungen auftreten können. Beim Hochlauf der Steuerung muss im Anwenderprogramm auch solange gewartet werden mit einer Auftragsaktivierung (z.B. NCK-Daten lesen), bis das Signal **NCK-CPU ready** gesetzt ist.

## 2.7.7 Funktionen des Grundprogramms mit Aufruf vom Anwenderprogramm

### Allgemeines

Zusätzlich zu den Modulen des Grundprogramms, die am Anfang vom OB 1, 40 und 100 aufzurufen sind, werden Funktionen bereitgestellt, die an geeigneter Stelle im Anwenderprogramm aufgerufen und parametrisiert werden müssen.

Mit diesen Funktionen können z. B. folgende Aufträge von PLC an den NCK übergeben werden:

- Verfahren von konkurrierenden Achsen (FC 15, FC 16),
- Starten asynchroner Unterprogramme (ASUPs) (FC 9),
- Anwahl von NC-Programmen und NC-Sätzen (FB 4),
- Steuern der Spindel (FC 18),
- Lesen/Schreiben von Variablen (FB 2, FB 3).

---

### Hinweis

Hier soll ein Hinweis gegeben werden, die die spätere Kontrolle und Diagnose eines Funktionsaufrufs (FCs, FBs des Grundprogramms) erleichtern. Hierbei handelt es sich um FCs, FBs, die über einen Anstoß gesteuert werden (z. B. über Parameter Req, Start, ...) und die als Ausgangsparameter eine Ausführungsquittierung liefern (z. B. über Parameter Done, NDR, Error, ...). Es sollte eine von anderen Signalen zusammengefasste Variable gesetzt werden, die den Anstoß für den Funktionsaufruf herbeiführen. Das Rücksetzen der Startbedingungen darf nur von den Parametern Done, NDR, Error abgeleitet werden.

Dieses Steuerwerk kann entweder vor oder hinter den Funktionsaufruf platziert werden. Wenn das Steuerwerk hinter dem Aufruf platziert ist, können die Ausgangsvariablen als lokale Variablen definiert sein (Vorteil: Reduzierung der globalen Variablen Merker, Datenvariable und zeitliche Vorteile gegenüber Datenvariable).

Der Anstoßparameter muss eine globale Variable sein (z.B. Merker, Datenvariable). Im OB 100 sind vom Anwenderprogramm noch aktivierte Aufträge (Parameter Req, Start, ... := TRUE) bei den genannten Parametern auf Null zu setzen. Durch ein Aus- und Wiedereinschalten kann ein Zustand mit noch aktivierten Aufträgen entstehen.

---

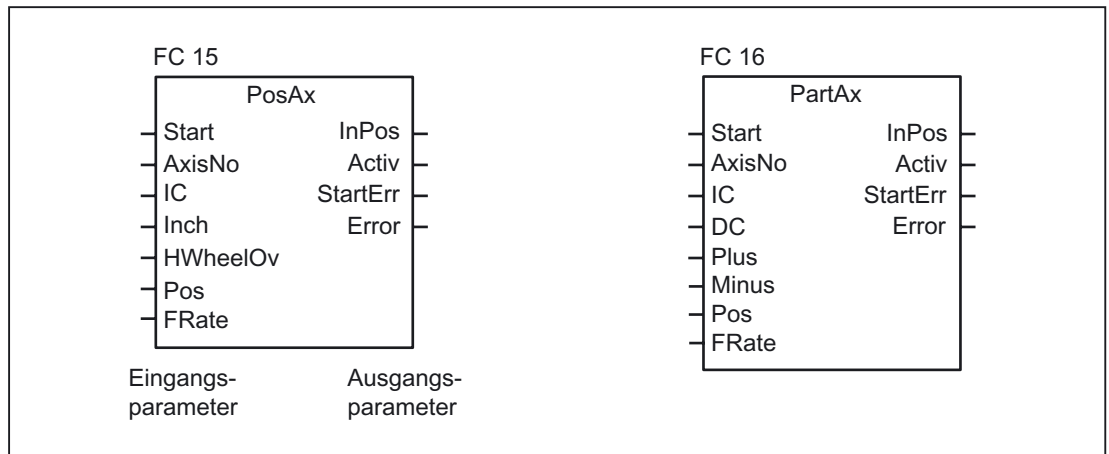


### Konkurrierende Achsen

Konkurrierende Achsen zeichnen sich durch folgende Eigenschaften aus:

- Sie können entweder von der PLC oder von der NC verfahren werden.
- Der Start von der PLC ist in allen Betriebsarten per FC möglich.
- Der Start erfolgt unabhängig von NC-Satzgrenzen.

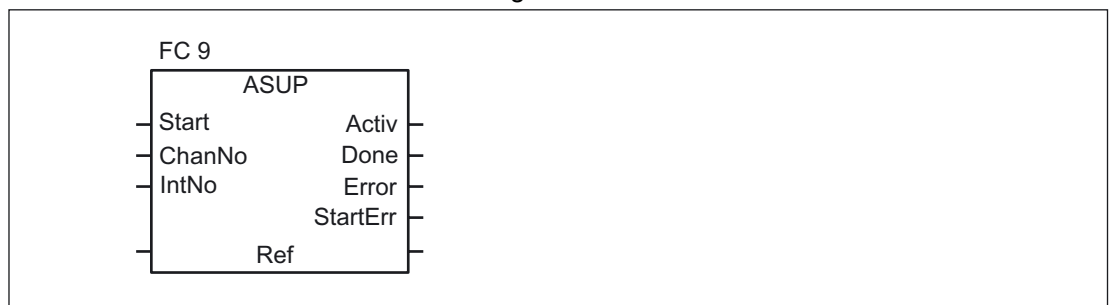
Es stehen FCs für Positionier- (FC 15) und Teilungs-Achsen (FC 16) zur Verfügung.



### ASUPs

Mit Asynchronen Unterprogrammen (ASUPs) können beliebige Funktionen in der NC ausgelöst werden. Voraussetzung dafür, dass ein ASUP von der PLC aus gestartet werden kann, ist dessen Existenz und Vorbereitung vom NC-Programm bzw. durch FB 4 PI-Dienste (ASUP). ASUPs können nur in MDA oder Automatic bei **laufendem** Teileprogramm gestartet werden.

Ein derart vorbereitetes ASUP kann von der PLC zu einem beliebigen Zeitpunkt gestartet werden. Das in dem betreffenden Kanal laufende NC-Programm wird durch das ASUP unterbrochen. Der Start eines ASUPs erfolgt durch den FC 9.



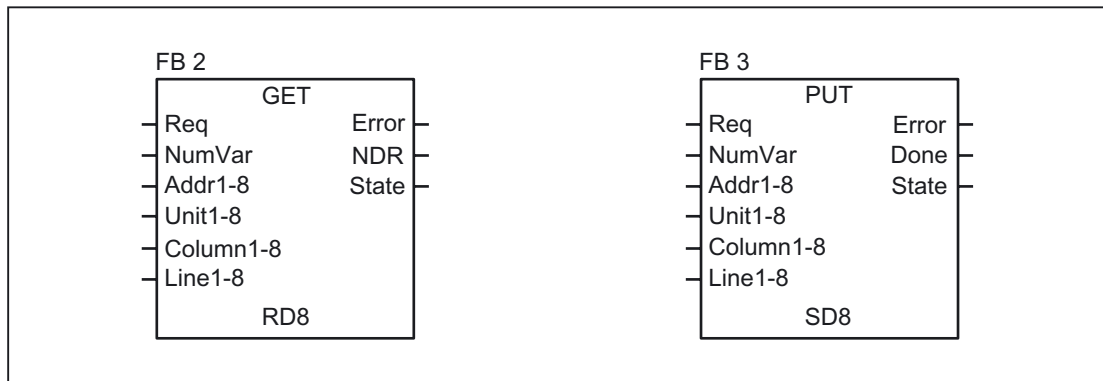
#### Hinweis

Ist ein ASUP noch nicht von einem NC-Programm oder vom FB 4 (ASUP) vorbereitet worden (z.B. keine Interrupt-Nr. vergeben), wird Start-Fehler gemeldet.

### NC-Variable lesen/schreiben

Mit dem FB GET können Variable des NCK gelesen und mit dem FB PUT können Werte in NCK-Variable eingetragen werden. Die NCK-Variablen werden über Bezeichner an den Eingängen Addr1 bis Addr8 adressiert. Die Bezeichner (Symbole) zeigen auf Adressangaben, die in einem globalen DB hinterlegt sein müssen. Zur Erzeugung dieses DBs wird zusammen mit dem Grundprogramm eine PC-Software (NC-Var-Selector) ausgeliefert, mit der aus einer ebenfalls mitgelieferten Tabelle die gewünschten Variablen selektiert werden können. Die selektierten Variablen werden zunächst in einer zweiten, projektabhängigen Liste gesammelt. Mit dem Kommando **DB erzeugen** wird ein \*.AWL-File erzeugt, der in die Programmdatei für die betreffende Maschine einzubinden und mit dem Maschinenprogramm zusammen zu kompilieren ist.

Es können 1 bis 8 Werte mit einem Lese- oder Schreibauftrag gelesen bzw. beschrieben werden. Die Werte werden - wenn notwendig - konvertiert (z.B. werden die NCK-Gleitpunktwerte (64 Bit) in das PLC-Format (32 Bit mit 24 Bit Mantisse und 8 Bit Exponent) gewandelt und umgekehrt). Durch die Wandlung von 64 Bit auf 32 Bit REAL geht Genauigkeit verloren. Die max. Genauigkeit der 32 Bit REAL Zahlen liegt etwa bei  $10^7$ .



## 2.7.8 Symbolische Programmierung des Anwenderprogramms mit Nahtstellen-DB

### Allgemeines

---

#### Hinweis

Die Dateien NST\_UDT.AWL und TM\_UDT.AWL werden mit dem PLC-Grundprogramm mitgeliefert.

---

In dem CPU-Programm des Grundprogramms sind die kompilierten UDT-Bausteine aus diesen beiden Dateien abgelegt.

Ein UDT ist ein vom Anwender definierter Datentyp, der z.B. einem Datenbaustein zugeordnet werden kann, der in der CPU erzeugt wurde.

In diesen UDT-Bausteinen sind symbolische Namen nahezu aller Nahtstellensignale definiert.

Verwendet werden die UDT-Nummern 2, 10, 11, 19, 21, 31, 71, 72, 73.

Folgende Zuordnung ist hierbei getroffen worden:

UDT-Zuordnungen		
UDT-Nummer	Zuordnung Nahtstellen-DB	Bedeutung
UDT2	DB2	Alarmer / Meldungen
UDT10	DB10	NCK-Signale
UDT11	DB11	BAG-Signale
UDT19	DB19	HMI-Signale
UDT21	DB21 bis DB30	Kanal-Signal
UDT31	DB31 bis DB61	Achs/Spindel-Signale
UDT71	DB71	Werkzeugverw.: Be-/Endladestellen
UDT72	DB72	Werkzeugverw.: Wechsel in Spindel
UDT3	DB73	Werkzeugverw.: Wechsel in Revolver

Für eine symbolische Programmierung der Nahtstellensignale müssen zuerst die Datenbausteine der Nahtstelle mit dem Symboleditor symbolisch zugeordnet werden.

Hierzu wird z.B. das Symbol "AchseX" dem Operand DB31 mit dem Datentyp UDT31 in der Symboldatei zugeordnet.

Nach dieser Eingabe kann die Programmierung des STEP 7 Programms für diese Nahtstelle symbolisch erfolgen.

---

#### Hinweis

Bisher erstellte Programme, die mit diesen Nahtstellen-DBs arbeiten, können auch in diese symbolische Form umgesetzt werden. Hierzu ist allerdings im bisher erstellten Programm eine voll qualifizierte Anweisung beim Datenzugriff notwendig (z.B. "U DB31.DBX 60.0", dieser Befehl wird umgesetzt in "AchseX.E\_SpKA" bei Einschaltung der Symbolik im Editor).

---

## Beschreibung

In den beiden AWL-Dateien NST\_UDT.AWL und TM\_UDT.AWL sind abgekürzte symbolische Namen der Nahtstellensignale definiert.

Um den Bezug auf die Namen der Nahtstellensignale herzustellen, ist hinter jedem Signal der Name im Kommentar mit aufgeführt.

Über den STEP 7-Editor können beim Aufschlagen des UDT-Bausteins die symbolischen Namen, Kommentare und Absolutadressen sichtbar gemacht werden.

---

### Hinweis

Die nicht verwendeten Bits und Bytes werden z.B. mit der Bezeichnung "f56\_3" aufgeführt.

"56": Byteadresse des jeweiligen Datenbausteins

"3": Bit-Nummer in diesem Byte

---

## 2.7.9 M-Dekodierung nach Liste

### Funktionsbeschreibung

Mit dem Aktivieren der Funktion **M-Dekodierung nach Liste** über den GP-Parameter des FB 1 "ListMDecGrp" (Anzahl M-Gruppen zur Dekodierung) können bis zu 256 M-Funktionen mit erweiterter Adresse vom Grundprogramm dekodiert werden.

Die Zuordnung von M-Funktion mit erweiterter Adresse und zu setzendem Bit in der Signalliste wird in der Dekodierliste festgelegt. Dabei erfolgt eine Einteilung in Gruppen.

Es gibt in der Signalliste 16 Gruppen mit jeweils 16 Bit als dekodierte Signale.

Die Dekodier- und Signalliste existieren nur ein Mal, also kanalunabhängig.

Die M-Funktionen werden dekodiert und wenn sie in der Dekodierliste enthalten sind, dann wird das zugeordnete Bit in der Signalliste gesetzt.

Mit dem Setzen in der Signalliste erfolgt gleichzeitig ein Setzen der Einlesesperre im zugehörigen NCK-Kanal durch das Grundprogramm.

Ein Rücksetzen der Einlesesperre im Kanal erfolgt dann, wenn vom Anwender alle von diesem Kanal ausgegebenen Bits in der Signalliste zurückgesetzt und damit quittiert wurden.

Bei Ausgabe einer aus Liste dekodierten M-Funktion als schnelle Hilfsfunktion erfolgt keine Einlesesperre.

Folgendes Bild stellt die Struktur der **M-Dekodierung nach Liste** dar:

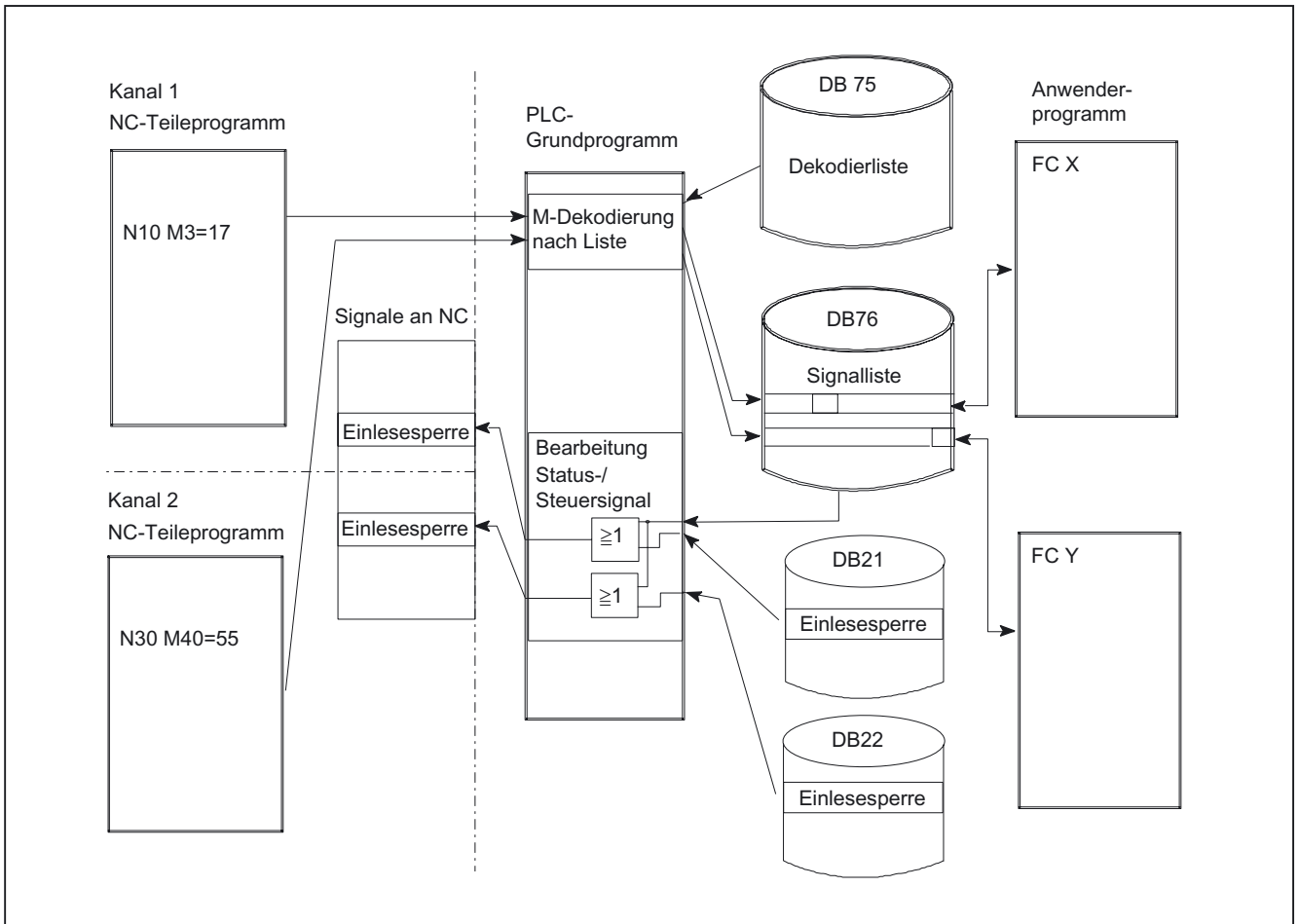


Bild 2-13 M-Dekodierung nach Liste

### Aktivierung der Funktion

Die Anzahl der auszuwertenden / dekodierenden Gruppen wird im Grundprogramm Parameter "ListMDecGrp" beim Aufruf des FB 1 im OB 100 angegeben (siehe auch Beschreibung FB 1). Die M-Dekodierung wird aktiviert, wenn dieser Wert zwischen 1 und 16 liegt. Vor Aktivierung der Funktion muss die Dekodierliste DB 75 in das AG übertragen und ein Neustart durchgeführt werden.

### Aufbau der Dekodierliste

Die Quelldatei für die Dekodierliste (MDECLIST.AWL) wird mit dem Grundprogramm ausgeliefert. Nach der Übersetzung der AWL-Quelle entsteht der DB 75.

Für jede zu dekodierende Gruppe von M-Funktionen muss ein Eintrag in der Dekodierliste DB 75 enthalten sein.

Höchstens 16 Gruppen können gebildet werden.

In jeder Gruppe stehen in der Liste der dekodierten Signale 16 Bit zur Verfügung.

Die Zuordnung von M-Funktion mit erweiterter Adresse und zu setzendem Bit in der Signalliste wird über erste und letzte M-Funktion in der Dekodierliste angegeben.

Dabei wird die Bitadresse entsprechend von der ersten M-Funktion ("MFirstAdr") bis zur letzten M-Funktion ("MLastAdr") von Bit 0 bis maximal Bit 15 für jede Gruppe gebildet.

Jeder Eintrag in der Dekodierliste besteht aus 3 Parametern, die jeweils einer Gruppe zugeordnet sind.

Zuordnung der Gruppen			
Gruppe	Erweiterte M-Adresse	erste M-Adresse in Gruppe	letzte M-Adresse in Gruppe
1	MSigGrp[1].MExtAdr	MSigGrp[1].MFirstAdr	MSigGrp[1].MLastAdr
2	MSigGrp[2].MExtAdr	MSigGrp[2].MFirstAdr	MSigGrp[2].MLastAdr
...	...	...	...
16	MSigGrp[16].MExtAdr	MSigGrp[16].MFirstAdr	MSigGrp[16].MLastAdr

Typ und Wertebereich für die Signale			
Signal	Typ	Wertebereich	Bemerkung
MExtAdr	Int	0 bis 99	Erweiterte M-Adresse
MFirstAdr	DInt	0 bis 99.999.999	erste M-Adresse in Gruppe
MLastAdr	Dint	0 bis 99.999.999	letzte M-Adresse in Gruppe

### Signalliste

Der Datenbaustein DB76 wird bei Aktivierung der Funktion vom Grundprogramm eingerichtet.

Für ein nach Liste dekodiertes M-Signal wird im DB 76 in der entsprechenden Gruppe ein Bit gesetzt.

Gleichzeitig erfolgt in dem Kanal, in dem die M-Funktion ausgegeben wurde, die Beeinflussung der Einlesesperre.

## Beispiel

Im folgenden Beispiel sollen 3 Gruppen an M-Befehlen dekodiert werden:

- · M2 = 1 bis M2 = 5
- · M3 = 12 bis M3 = 23
- · M40 = 55

Aufbau der Dekodierliste im DB 75:

Beispielparameter				
Gruppe	Dekodierliste (DB 75)			Signalliste
	Erweiterte M-Adresse	Erste M-Adresse in Gruppe	Letzte M-Adresse in Gruppe	DB 76
1	2	1	5	DBX0.0 bis DBX0.4
2	3	12	23	DBX2.0 bis DBX3.3
3	40	55	55	DBX4.0

```

DATA_BLOCK DB 75
TITLE =
VERSION : 0.0
    STRUCT
        MSigGrp : ARRAY [1 .. 16 ] OF STRUCT
            MExtAdr : INT ;
            MFirstAdr : DINT ;
            MLastAdr : DINT ;
        END_STRUCT ;
    END_STRUCT ;
BEGIN
    MSigGrp[1].MExtAdr := 2;
    MSigGrp[1].MFirstAdr L#1;
    :=
    MSigGrp[1].MLastAdr L#5;
    :=
    MSigGrp[2].MExtAdr := 3;
    MSigGrp[2].MFirstAdr L#12;
    :=
    MSigGrp[2].MLastAdr L#23;
    :=
    MSigGrp[3].MExtAdr := 40;
    MSigGrp[3].MFirstAdr L#55;
    :=
    MSigGrp[3].MLastAdr L#55;
    :=
END_DATA_BLOCK
    
```

### Aufbau des FB 1 im OB100

(Anzahl M-Gruppen zur Dekodierung eintragen und damit Funktionalität aktivieren):

```
Call FB 1, DB 7(  
...  
ListMDecGrp := 3, // M-Dekodierung von 3 Gruppen  
...  
);
```

Nachdem der Eintrag im OB 100 angefügt ist und der DB 75 (Dekodierliste) in das AG übertragen wurde, muss ein Neustart erfolgen. Das Grundprogramm richtet nun im Neuanlauf den DB 76 (Signalliste) ein.

Wird nun das NC-Programm gestartet und die erweiterte M-Funktion (z.B. M3=17) vom NCK abgearbeitet, dann wird diese M-Funktion dekodiert und im DB 76 wird das Bit 2.5 gesetzt (siehe Dekodierliste DB 75). Gleichzeitig setzt das Grundprogramm die Einlesesperre und die Abarbeitung des NC-Programms wird gestoppt (im entsprechenden NC-Kanal-DB erfolgt ebenfalls der Eintrag der "erweiterten Adresse M-Funktion" und die "M-Funktions-Nr.").

Die Einlesesperre im Kanal wird zurückgesetzt, wenn der Anwender alle von diesem Kanal ausgegebenen Bits in der Signalliste (DB 76) zurückgesetzt und damit quittiert hat.



## 2.7.10 PLC-Maschinendaten

### Allgemeines

Es besteht für den Anwender die Möglichkeit, PLC-spezifische Maschinendaten in der NCK zu speichern. Diese Maschinendaten können nach Hochlauf der PLC (OB 100) vom Anwender verarbeitet werden. Dadurch können z.B. Anwender-Optionen, Maschinenausbaustufen, Maschinenkonfiguration, etc. realisiert werden.

Die Nahtstelle zum Lesen dieser Daten liegt im DB 20. Der DB 20 wird aber nur bei Verwendung der Anwender-Maschinendaten, d.h. Summe aus den GP-Parametern "UDInt", "UDHex" und "UDReal" ist größer NULL, vom Grundprogramm im Hochlauf angelegt.

Die Größe der einzelnen Bereiche, und damit auch die Gesamtlänge des DB 20, wird durch die PLC-Maschinendaten:

MD14504 MAXNUM\_USER\_DATA\_INT

MD14506 MAXNUM\_USER\_DATA\_HEX

MD14508 MAXNUM\_USER\_DATA\_FLOAT

eingestellt und dem Anwender in den GP-Parametern:

"UDInt", "UDHex" und "UDReal"

angegeben.

Die Datenablage im DB 20 erfolgt durch das GP bündig in der Reihenfolge: Integer-MD, Hexa-Felder-MD, Real-MD.

Die Integer- und Realwerte werden im S7-Format im DB 20 abgelegt.

Die Hexa-Daten werden in der Reihenfolge der Eingabe (Nutzung als Bitfelder) im DB 20 gespeichert.

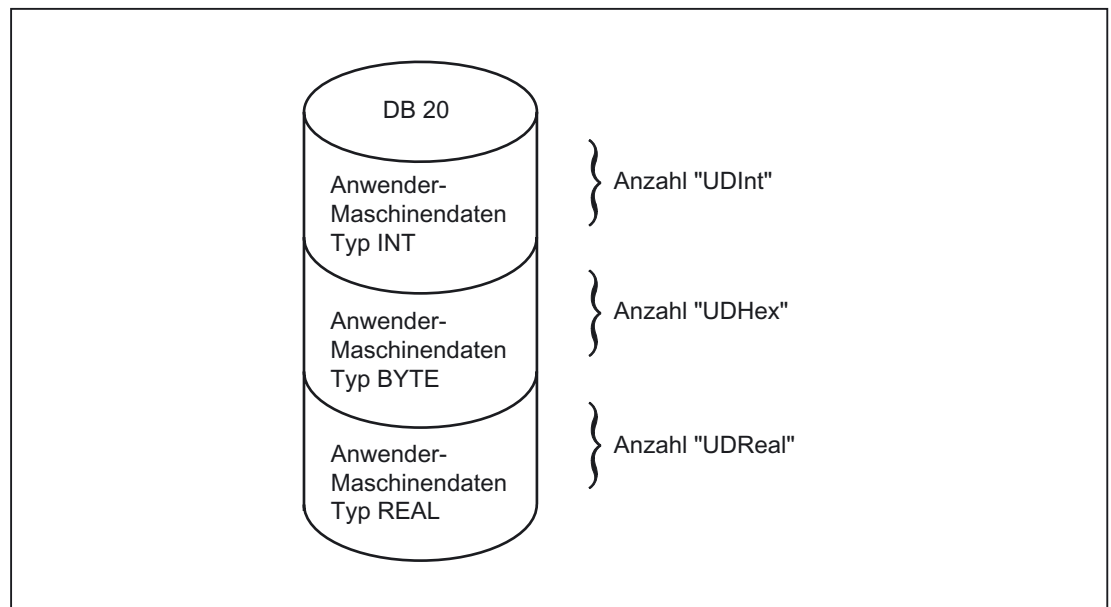


Bild 2-14 DB 20

**Hinweis**

Soll die Anzahl der genutzten PLC-Maschinendaten später erhöht werden, ist es notwendig, den DB 20 vorher zu löschen. Damit solche Erweiterungen keine Auswirkungen auf das bestehende Anwenderprogramm haben, sollten die Zugriffe auf die Daten im DB 20 möglichst symbolisch erfolgen, z.B. über eine Strukturbeschreibung im UDT.

Alarmer	
<b>400120</b>	<b>DB 20 im PLC löschen und Neustart</b>
Erläuterung	DB-Länge ist unterschiedlich zur geforderten DB Länge
Reaktion	Alarmanzeige und PLC-Stopp
Abhilfe	DB 20 löschen und anschließender RESET
Fortsetzung	nach Neustart

**Beispiel**

Für das Projekt im Beispiel werden 4 Integer-Werte, 2 Hexa-Felder mit Bitinformationen und 1 Real-Wert benötigt.

Maschinendaten:

```

MD14510 USER_DATA_INT[0]      123
MD14510 USER_DATA_INT[1]      456
MD14510 USER_DATA_INT[2]      789
MD14510 USER_DATA_INT[3]     1011
...
MD14512 USER_DATA_HEX[0]      12
MD14512 USER_DATA_HEX[1]      AC
...
MD14514 USER_DATA_FLOAT[0]    123.456
    
```

GP-Parameter (OB 100):

```

CALL FB 1, DB 7(
    MCPNum := 1,
    MCP1In := P#E0.0,
    MCP1Out := P#A0.0,
    MCP1StatSend := P#A8.0,
    MCP1StatRec := P#A12.0,
    MCP1BusAdr := 6,
    MCP1Timeout := S5T#700MS,
    MCP1Cycl := S5T#200MS,
    NCCyclTimeout := S5T#200MS,
    NCRunupTimeout := S5T#50S;
)
    
```

GP-Parameter (zur Laufzeit abfragen):

```

1 gp_par.UDInt;    //=4,
1 gp_par.UDHex;   //=2,
1 gp_par.UDReal;  //=1 )
    
```

Im Hochlauf der PLC wurde der DB 20 mit einer Länge von 28 Byte erstellt:

DB 20	
Adresse	Daten
0.0	123
2.0	456
4.0	789
6.0	1011
8.0	b#16#12
9.0	b#16#AC
10.0	1.234560e+02

Die Struktur der genutzten Maschinendaten wird in einem UDT angegeben:

```

TYPE UDT 20
  STRUCT
    UDInt : ARRAY [0 .. 3] OF INT;
    UDHex0 : ARRAY [0 .. 15] OF BOOL;
    UDReal : ARRAY [0 .. 0] OF REAL; //Beschreibung als Feld,
                                     //für spätere Erweiterungen

  END_STRUCT;
END_TYPE
    
```

---

**Hinweis**

ARRAY OF BOOL werden immer an geradzahigen Adressen ausgerichtet. Deshalb ist in der Definition des UDT generell ein Array-Bereich von 0 bis 15 zu wählen bzw. alle Booleschen Variablen sind einzeln aufzuführen.

---

Obwohl im Beispiel zunächst nur ein REAL-Wert genutzt wird, wurde für die Variable ein Feld (mit einem Element) angelegt. Dies sichert eine einfache Erweiterung zu einem späteren Zeitpunkt, ohne dass sich die symbolische Adresse ändert.

### Symbolische Zugriffe

Für den symbolischen Zugriff erfolgt ein Eintrag in der Symbolliste:

Symbol	Operand	Datentyp
UData	DB 20	UDT 20

Zugriffe im Anwenderprogramm (nur symbolische Lesezugriffe dargestellt):

```
...  
  
L      "UData".UDInt[0];  
L      "UData".UDInt[1];  
L      "UData".UDInt[2];  
L      "UData".UDInt[3];  
  
U      "UData".UDHex0[0];  
U      "UData".UDHex0[1];  
U      "UData".UDHex0[2];  
U      "UData".UDHex0[3];  
U      "UData".UDHex0[4];  
U      "UData".UDHex0[5];  
U      "UData".UDHex0[6];  
U      "UData".UDHex0[7];  
...    ...  
U      "UData".UDHex0[15];  
  
L      "UData".UDReal[0];  
  
...
```

## 2.7.11 Projektierung von Maschinensteuertafel, Bedienhandgerät

### Allgemeines

Über die in der NC eingebaute Kommunikation wird bei 810D und 840D eine Übertragung von maximal 2 Maschinensteuertafeln und einem Bedienhandgerät durchgeführt. Für die Übertragung der Signale dieser Komponenten wird kein SDB210 benötigt. Die folgenden Informationen gehen davon aus, dass kein SDB210 für diese Komponenten eingesetzt wird.

Die Parametrierung der Komponenten erfolgt grundsätzlich über den Aufruf des Grundprogrammbausteins FB1 im OB100. Der FB1 hält seine Parameter im zugehörigen Instanz-DB (DB7, symbolisch "gp\_par"). Hierzu existieren für jede Maschinensteuertafel und das Bedienhandgerät eigene Parametersätze. In diesen Parametersätzen ist die Ein-, Ausgangsadresse vom Anwender zu definieren. Diese Ein- und Ausgangsadressen werden auch im FC19, FC24, FC25, FC26 und FC13 verwendet. Weiterhin sind noch Adressen für Statusinformationen, MPI bzw. BTSS (bei dem Bedienhandgerät ist statt einer MPI Adresse ein GD Parametersatz einzustellen) zu definieren. Die Zeiteinstellungen für Timeout und zyklische Zwangsnachtriggerung können auf dem voreingestellten Wert bleiben.

### Aktivierung

Die jeweilige Komponente wird entweder über die Anzahl der Maschinensteuertafeln (Parameter MCPNum) oder beim Bedienhandgerät über den Parameter BHG := 2 aktiviert (BHG := 1 entspricht einer Kopplung über die MPI Schnittstelle in Verbindung mit einem SDB210). Die Festlegung, ob eine Ankopplung an der BTSS oder der MPI erfolgen soll, wird über die Parameter MCPMPI bzw. BHGMPI vorgenommen.

### Bedienhandgerät

Beim Bedienhandgerät wird die Adressierung der MPI bzw. BTSS über einen GD Parametersatz vorgenommen. Die Werte dieser Parametrierung sind von der Einstellung des Bedienhandgerätes zu übernehmen. Allerdings sind die Parameternamen am Bedienhandgerät invers zu den Parameternamen des Grundprogramms. Vom Bedienhandgerät sind alle Parameter vom Typ Send im Grundprogramm auf den Typ Rec (bzw. auch Typ Rec in Send) zu definieren.

### Steuersignale

Mit Hilfe der Parameter MCP1Stop, MCP2Stop und BHGStop ist ein Anhalten der Kommunikation zu den einzelnen Komponenten möglich (Wertzuweisung = 1). Dieses Stoppen bzw. auch aktivieren der Kommunikation ist im laufenden Zyklus möglich. Allerdings darf die Wertveränderung nicht über einen erneuten Aufruf des FB 1 erfolgen, sondern durch die symbolische Schreibweise der Parameter.

Beispiel für Stoppen der Übertragung der 1. Maschinensteuertafel:

```
SET;
S          gp_par.MCP1Stop;
```

Bei gesetzten Parametern MCP1Stop, MCP2Stop, BHGStop erfolgt auch eine Unterdrückung der Alarmer 40260 bis 40262

### Umschaltung MPI, BTSS Adresse

Eine bestehende Verbindung zu einer Maschinensteuertafel (MSTT) oder Bedienhandgerät (BHG) kann abgebaut werden. Eine andere MSTT, BHG Komponente, die sich bereits am Bus befindet (andere MPI, BTSS Adresse), kann anschließend aktiviert werden. Für diese Umschaltung ist folgende Vorgehensweise erforderlich:

1. Anhalten der Kommunikation der abzukoppelnden Komponente über den Parameter MCP1Stop bzw MCP2Stop bzw. BHGStop = 1
2. Nach Rückmeldung im DB10 Byte 104 (relevante Bits 0, 1, 2 auf 0 Zustand). Änderung der Busadresse bzw. GD-Parametersatz dieser Einheit auf die neue Komponente.
3. In diesem PLC Zyklus kann nun die Kommunikation der neuen Komponente wieder aktiviert werden über den Parameter MCP1Stop bzw MCP2Stop bzw. BHGStop = 0
4. Kommunikation mit der neuen Komponente läuft, wenn Rückmeldung im DB10 Byte 104 (relevante Bits 0, 1, 2 auf 1 Zustand).

Alle Parameter sind, wie in dem Abschnitt Steuersignale beschrieben, je nach Datentyp zu programmieren.

### Abschalten des Blinkens der MCP:

Ab der MSTT-Firmware V5.01.02 kann das Blinken im Offline Modus unterdrückt werden. Beim Offline Modus findet keine Kommunikation statt (z.B. auch bei Ausfall der MSTT-Verbindung). Über die Statusbits 24 bzw. 25 im MCP1StatSend, MCP2StatSend kann bei aktiver Kommunikation ein Wert in den Ausgangsdaten vorgegeben werden. Die Rückmeldung, das der Wert übernommen wurde, wird in den gleichen Bits im MCP1StatRec, MCP2StatRec zurückgemeldet. Nach erfolgter Rückmeldung müssen die Statusbits im Send-Status wieder rückgesetzt werden. Beispiel über Realisierung liegt auf der Toolbox.

### Temperaturüberwachung der MCP:

Ab der MSTT Firmware V5.01.02 wird eine erhöhte Temperatur über Bit 28 = 1 in MCP1StatRec, MCP2StatRec zurückgemeldet.

## Projektierung

Grundsätzlich existieren zwei Kommunikationsmechanismen, bedingt durch den Anschluss von MSTT und BHG, zur Übertragung der Daten zwischen MSTT/BHG und PLC. Im ersten Fall werden die Daten durch das Komm-Modul (840D/810D) transportiert. Die Parametrierung erfolgt dabei vollständig über die MSTT/BHG Parameter im FB1. Im anderen Fall erfolgt die Übertragung durch das PLC-Betriebssystem (FM-NC) durch Auswertung des SDB210 (Globale Daten) oder durch die Profibus-Konfiguration. Die Parametrierung erfolgt über STEP-> 7 Globale Daten oder im HW-Konfig. Für den Zugriff des Grundprogrammes auf diese Daten und Ausfallüberwachung von MSTT/BHG müssen die über SDB210 (globale Daten) eingestellten Adressen in FB1.-Parametern dem Grundprogramm bekannt gemacht werden.

Im Weiteren folgt eine Übersicht über die verschiedenen Kopplungsmöglichkeiten, abhängig davon, welche NC genutzt wird. Dazu wird jedes mal der Parametersatz vom FB1 und die gültige Statusinformation angegeben, der für die jeweilige Art der Datenübertragung relevant ist.

Wird ein Fehler aufgrund einer Zeitüberwachung erkannt, erfolgt ein Eintrag im Diagnosepuffer der PLC-CPU (Fehler 400260 bis 400262). In diesem Fall werden die Eingangssignale von der MSTT bzw. vom Bedienhandgerät (MCP1In/MCP2In bzw BHGIn) mit 0 initialisiert. Sollte eine Neusynchronisation zwischen PLC und MSTT/BHG möglich sein, wird die Kommunikation automatisch wieder aufgenommen und die Fehlermeldung vom GP gelöscht.

### 840D: BTSS/MPI-Ankopplung

Kommunikation erfolgt vom PLC-GP aus über NCK und KOMM-Modul, d.h. auch bei Kopplung über MPI ist kein SDB210 erforderlich. Die Parametrierung erfolgt über die relevanten Parameter im FB1.

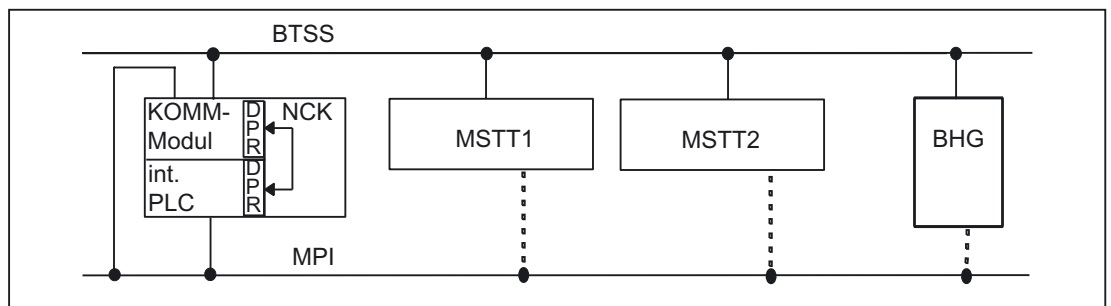


Bild 2-15 840D: BTSS/MPI-Ankopplung

Relevante Parameter (FB1)		
MSTT		BHG
MCPNum=1 oder 2 (Anzahl MSTT)		BHG=2 (Transfer über Komm-Modul)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend	MCP2StatSend	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen

Relevante Parameter (FB1)		
MSTT		BHG
MCP1Cycl	MCP2Cycl	BHGTimeout
MCPMPI = FALSE (BTSS), TRUE (MPI)		BHGCycl
MCP1Stop	MCP2Stop	BHGRecGDNo
MCPBusType=0		BHGRecGBZNo
		BHGRecObjNo
		BHGSendGDNo
		BHGSendGBZNo
		BHGSendObjNo
		BHGMPI = FALSE (BTSS), TRUE (MPI)
		BHGStop

Statusinformationen		
Verfügbar in	Bit Nr.	Beschreibung
MCP1StatSend MCP2StatSend BHGStatSend	4	Syntaxfehler im GD-Paket: Fehler im Parametersatz (FB1)
MCP1StatSend MCP2StatSend BHGStatSend	27	Sender: Zeitüberwachung abgelaufen
MCP1StatRec MCP2StatRec BHGStatRec	10	Empfänger: Zeitüberwachung abgelaufen

Außerdem wird für die Zeitüberwachungen (Bit 10 und Bit 27) ein Fehlereintrag im Diagnosepuffer der PLC generiert.

Daraus entstehen an der Bedienoberfläche die Fehlermeldungen:

- 400260: MSST 1 ausgefallen
- 400261: MSST 2 ausgefallen
- 400262: BHG ausgefallen

Ein Ausfall von MSTT oder BHG wird auch gleich nach dem Neustart erkannt, auch dann wenn noch keine Daten zwischen MSTT/BHG und PLC ausgetauscht wurden. Die Überwachungsfunktion wird aktiviert, sobald alle Komponenten nach dem Hochlauf "Ready" melden.

#### 840D: MPI-Ankopplung für BHG (nicht für Neuentwicklung)

Kommunikation für BHG durch PLC-Besy und Parametrierung über SDB210.

Kommunikation für die MSTT erfolgt vom PLC-GP aus über NCK und KOMM-Modul, wie oben beschrieben.



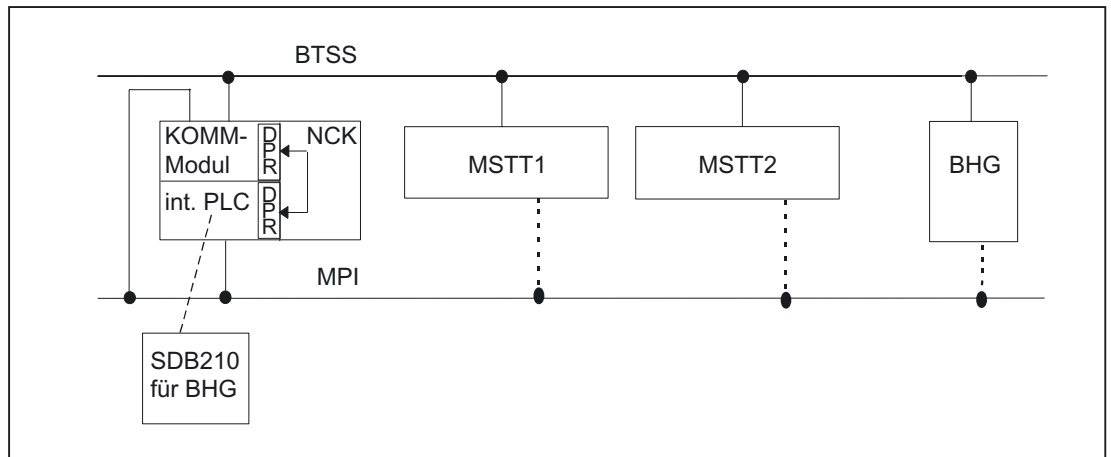


Bild 2-16 840D: MPI-Ankopplung für BHG

Relevante Parameter (FB1):

Die Kommunikation zwischen PLC und BHG erfolgt durch Projektierung und anschließendes Laden des SDB210 (Globale Daten). Für den Zugriff des Grundprogrammes auf die BHG-Daten und Ausfallüberwachung des BHG müssen die über Globale Daten eingestellten Adressen in FB1-Parametern bekanntgemacht werden.

Relevante Parameter (FB1)	
MSTT	BHG (Parametrierung über SDB210)
Die Parametrierung erfolgt über die relevanten Parameter im FB1	BHG = 1 (Transfer über SDB210)
	BHGIn (wie in SDB210 parametriert)
	BHGOut (wie in SDB210 parametriert)
	BHGStatRec (wie in SDB210 parametriert)
	BHGTimeout (wie in SDB210 parametriert)

Statusinformationen (MCP1 und MCP2 siehe Tab. Statusinformation)		
Verfügbar in	Bit Nr.	Beschreibung
BHGStatRec	10	Empfänger: Zeitüberwachung abgelaufen,

Außerdem wird für die Zeitüberwachung ein Fehlereintrag im Diagnosepuffer der PLC generiert.

Daraus entsteht an der Bedienoberfläche die Fehlermeldung 400262: BHG ausgefallen

Ein Ausfall vom BHG wird nur erkannt, wenn zuvor ein Datenaustausch zum BHG stattgefunden hat. Der erste Datenaustausch zum BHG aktiviert die Überwachungsfunktion.

### MPI-Ankopplung

Kommunikation erfolgt vom PLC-GP aus über NCK und KOMM-Modul, d.h. auch bei Kopplung über MPI ist kein SDB210 erforderlich. Die Parametrierung erfolgt über die relevanten Parameter im FB1.

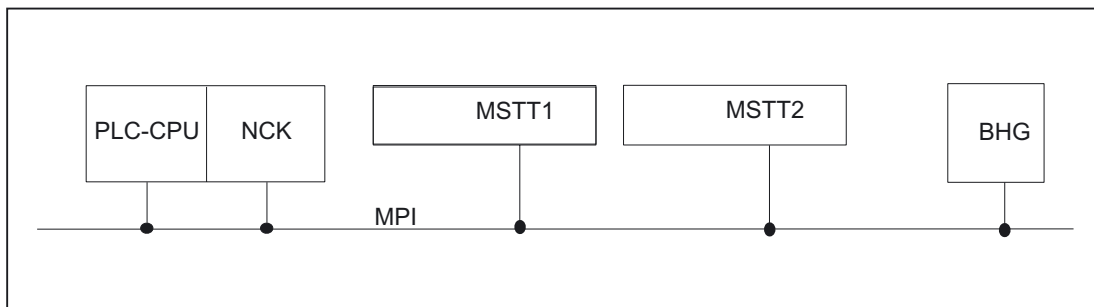


Bild 2-17 MPI-Ankopplung

Relevante Parameter (FB1)		
MSTT		BHG
MCPNum=1 oder 2 (Anzahl MSTT)		BHG=2 (Transfer über Komm-Modul)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend	MCP2StatSend	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl	MCP2Cycl	BHGTimeout
<b>MCPMPI = TRUE (MPI)</b>		BHGCycl
MCP1Stop	MCP2Stop	BHGRecGDNo
MCPBustype=0		BHGRecGBZNo
		BHGRecObjNo
		BHGSendGDNo
		BHGSendGBZNo
		BHGSendObjNo
		<b>BHGMPI = TRUE (MPI)</b>
		BHGStop

Statusinformationen		
Verfügbar in	Bit Nr.	Beschreibung
MCP1StatSendMCP2StatSend BHGStatSend	4	Syntaxfehler im GD-Paket: Fehler im Parametersatz (FB1)
MCP1StatSendMCP2StatSendB HGStatSend	27	Sender: Zeitüberwachung abgelaufen
MCP1StatRec MCP2StatRec BHGStatRec	10	Empfänger: Zeitüberwachung abgelaufen

Außerdem wird für die Zeitüberwachungen (Bit 10 und Bit 27) ein Fehlereintrag im Diagnosepuffer der PLC generiert.

Daraus entstehen an der Bedienoberfläche die Fehlermeldungen:

- 400260: MSST 1 ausgefallen  
oder
- 400261: MSST 2 ausgefallen
- 400262: BHG ausgefallen

Ein Ausfall von MSTT oder BHG wird auch gleich nach dem Neustart erkannt, auch dann wenn noch keine Daten zwischen MSTT/BHG und PLC ausgetauscht wurden. Die Überwachungsfunktion wird aktiviert, sobald alle Komponenten nach dem Hochlauf "Ready" melden.

#### MPI-Ankopplung und 810D (vor SW-Stand 4)

Kommunikation für MSTT und BHG durch PLC-Besy und Parametrierung über SDB210.

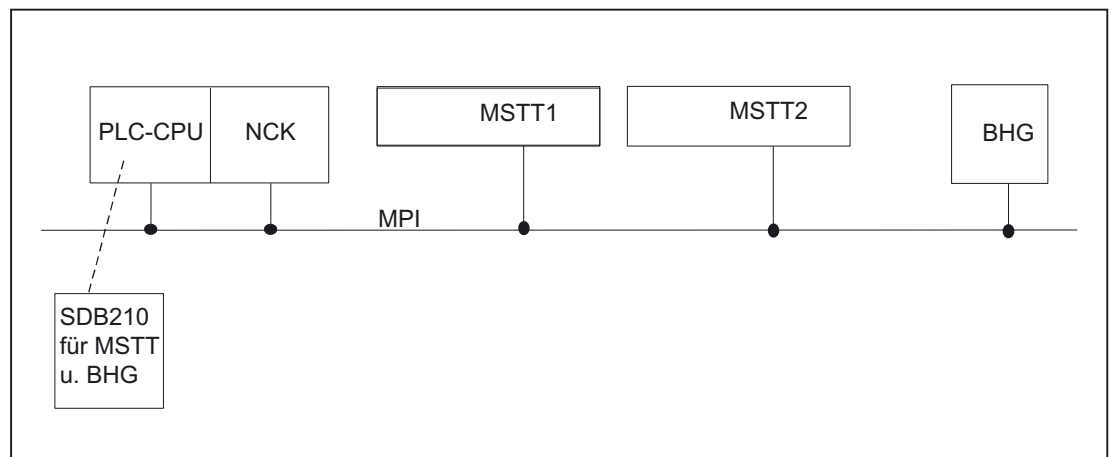


Bild 2-18 MPI-Ankopplung und 810D

Relevante Parameter (FB1):

Die Kommunikation zwischen PLC und BHG erfolgt durch Projektierung und anschließendes Laden des SDB210 (Globale Daten). Für den Zugriff des Grundprogrammes auf die BHG-Daten und Ausfallüberwachung des BHG müssen die über Globale Daten eingestellten Adressen in FB1-Parametern bekanntgemacht werden.

Relevante Parameter (FB1), (alle Einträge wie im SDB210-Globale Daten parametrier)		
MSTT		BHG
MCPNum=1 oder 2 (Anzahl MSTT)		BHG=1 (MPI)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1Timeout	MCP2Timeout	BHGTimeout

Statusinformationen		
Verfügbar in	Bit Nr.	Beschreibung
MCP1StatRec MCP2StatRec BHGStatRec	10	Empfänger: Zeitüberwachung abgelaufen

Außerdem wird für die Zeitüberwachung ein Fehlereintrag im Diagnosepuffer der PLC generiert.

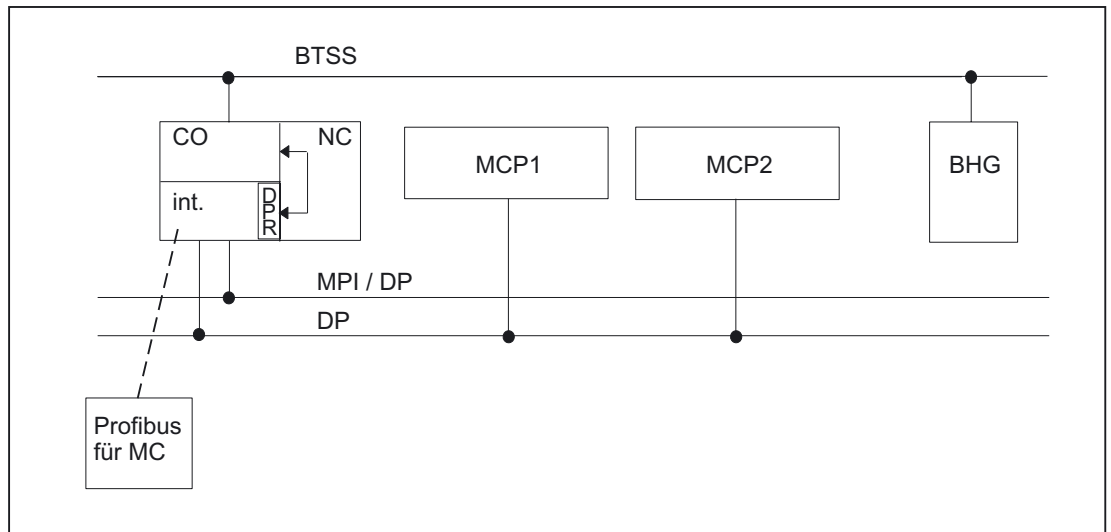
Daraus entstehen an der Bedienoberfläche die Fehlermeldungen:

- 400260: MSST 1 ausgefallen
- 400261: MSST 2 ausgefallen
- 400262: BHG ausgefallen

Ein Ausfall von MSTT/BHG wird nach dem Neustart nur erkannt, wenn zuvor ein Datenaustausch zu MSTT/BHG stattgefunden hat. Der erste Datenaustausch zu MSTT/BHG aktiviert die Überwachungsfunktion.

#### 840D: Profibus-Ankopplung

Bei der Profibusankopplung der MCP muss diese Komponente in der Hardwareprojektierung von STEP7 berücksichtigt werden. Die Adressen sind in den Eingangs- und Ausgangs-Abbildbereich zu legen. Diese Anfangs-Adressen sind auch in den Pointer Parametern des FB1 zu hinterlegen. Die weitere Parametrierung erfolgt über die unten aufgeführten Parameter des FB1. Eine Profibus-Variante des BHG existiert nicht. Deshalb ist in diesem Bild ein BTSS-Anschluss für das BHG gezeichnet. In MCP1BusAdr und MCPBusAdr kann die Profibus-Slave Adresse hinterlegt werden.



Relevante Parameter(FB1)		
MCP		BHG
MCPNum = 1 oder 2 (Anzahl MCP)		BHG = 2 (über COM-Modul)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec (n.r.)	MCP2StatRec (n.r.)	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout (n.r.)	MCP2Timeout (n.r.)	BHGOutLen
MCP1Cycl (n.r.)	MCP2Cycl (n.r.)	BHGTimeout
MCPMPI = FALSE		BHGCycl
MCP1Stop (n.r.)	MCP2Stop (n.r.)	BHGRecGDNo
MCPBusType = 3		BHGRecGBZNo
		BHGRecObjNo
		BHGSendGDNo
		BHGSendGBZNo
		BHGSendObjNo
		BHGMPI = FALSE
		BHGStop

Ein Ausfall der MCP führt die PLC in den STOP-Zustand. Falls dieses nicht gewünscht ist, dann kann durch OB82, OB86 ein Stopp vermieden werden. Weiterhin kann auch die MCP (als Profibus Slave) über den SFC12 an- und abgeschaltet werden. Alarmmeldungen durch das Grundprogramm werden bei einem Ausfall nicht erzeugt. M zu N ist mit der MCP- Variante nicht möglich.

## 2.8 SPL für Safety Integrated

Siehe:

**Literatur:**

/FBSI/ Funktionsbeschreibung Safety Integrated

## 2.9 Belegungsübersicht

### 2.9.1 Belegung: NC/PLC-Nahtstelle

Die Belegung der NC/PLC-Nahtstelle ist ausführlich beschrieben in:

**Literatur:**  
/LIS/ Listen

### 2.9.2 Belegung: FB/FC

FB-Nummer	FC-Nummer	Bedeutung
1		Basis-Grundprogramm
2 - 29		Reserviert für Siemens
	1	Reserviert für Siemens
	2 - 29	Reserviert für Siemens
	30 - 35	Siehe unten: ShopMill, ManualTurn
	30 – 127 (siehe Hinweis) <sup>1)</sup>	<b>Anwenderbereich</b>
30 – 127 (Siehe Hinweis)		<b>Anwenderbereich</b>

#### Hinweis

Die tatsächliche Obergrenze der Baustein-Nummer (FB/FC) ist abhängig von der aktuellen PLC-CPU, die in der gewählten NCU enthalten ist. Siehe Kapitel "Eckdaten der PLC-CPU's für 810D, 840D". Zur Belegung der FC's, FB's siehe Kapitel "Speicherbedarf des PLC-Grundprogramms für 810D, 840D".

#### ManualTurn

ManualTurn verwendet FC 30 bis 35 und DB 81 bis 89.

ManualTurn ist eine Bedienung für konventionelle zyklengesteuerte Drehmaschinen. Die FC's und DB's können verwendet werden, wenn es sich bei der zu projektierenden Maschine nicht um eine Drehmaschine mit max. 2 Achsen und einer Spindel handelt. Wenn es sich um eine solche Maschine handelt, und zusätzlich zur CNC-Bedienung eine konventionelle Bedienung nicht ausgeschlossen ist, sollten die FC's und DB's nicht benutzt werden.

## ShopMill

ShopMill verwendet FC30 bis 35 und DB 81 bis 89.

ShopMill ist eine Bedienung für 2 1/2D Fräsmaschinen im Werkstattbetrieb. Die FC's und DB's können verwendet werden, wenn es sich bei der zu projektierenden Maschine nicht um eine Fräsmaschine für 2 1/2D Bearbeitung handelt. Ist die Maschine jedoch für einen solchen Einsatzfall gedacht, sollten die FC's und DB's nicht benutzt werden.

### 2.9.3 Belegung: DB

---

#### Hinweis

Es werden nur so viele Datenbausteine eingerichtet, wie aufgrund der Projektierung in den NC-Maschinendaten erforderlich sind.

---

Übersicht der Datenbausteine			
DB-Nr.	Bezeichnung	Name	Paket
1		reserviert für Siemens	GP
4	PLC-MELD	PLC-Meldungen	GP
8		reserviert für Siemens	
9	NC-COMPILE	Nahtstelle für NC-Compilezyklen	GP
10	NC-NAHTSTELLE	Zentrale NC-Nahtstelle	GP
11	BAG 1	Nahtstelle BAG	GP
12		Nahtstelle Rechnerkopplung und Transportsystem	
14		reserviert für Siemens	
15		Grundprogramm	
16		PI-Dienst Definition	
17		Versionskennung	
18		reserviert für Grundprogramm	
19		Nahtstelle HMI	
20		PLC-Maschinendaten	
21 - 30	KANAL 1 ... n	Nahtstelle NC-Kanäle	GP
31 - 61	ACHSE 1 ... m	Nahtstellen für Achsen/Spindeln bzw. frei für Anwender	GP
62 - 70		Frei für Anwender	
71 - 74		Werkzeugverwaltung	GP
75 - 76		M-Gruppen-Dekodierung	
77 - 80		reserviert für Siemens	
81 - 89		Siehe unten: ShopMill, ManualTurn	
81 -127 (siehe Hinweis)		<b>Anwenderbereich</b>	

---

**Hinweis**

Die tatsächliche Obergrenze der Baustein-Nummer (DB) ist von der aktuellen PLC-CPU abhängig, die in der gewählten NCU enthalten ist. Siehe Kapitel "Eckdaten der PLC-CPUs für 810D, 840D".

---

**ManualTurn**

ManualTurn verwendet FC30 bis 35 und DB81 bis 89.

ManualTurn ist eine Bedienung für konventionelle zyklengesteuerte Drehmaschinen. Die FC's und DB's können verwendet werden, wenn es sich bei der zu projektierenden Maschine nicht um eine Drehmaschine mit max. 2 Achsen und einer Spindel handelt. Wenn es sich um eine solche Maschine handelt, und zusätzlich zur CNC-Bedienung eine konventionelle Bedienung nicht ausgeschlossen ist, sollten die FC's und DB's nicht benutzt werden.

**ShopMill**

ShopMill verwendet FC30 bis 35 und DB81 bis 89.

ShopMill ist eine Bedienung für 2 1/2D Fräsmaschinen im Werkstattbetrieb. Die FC's und DB's können verwendet werden, wenn es sich bei der zu projektierenden Maschine nicht um eine Fräsmaschine für 2 1/2D Bearbeitung handelt. Ist die Maschine jedoch für einen solchen Einsatzfall gedacht, sollten die FC's und DB's nicht benutzt werden.

**2.9.4 Belegung: Timer**

Timer-Nr.	Bedeutung
0 - 9	Reserviert für Siemens
10 - 127	Anwenderbereich

---

**Hinweis**

Die tatsächliche Obergrenze der Baustein-Nummer (Timer) ist von der aktuellen PLC-CPU abhängig, die in der gewählten NCU enthalten ist. Siehe Kapitel "Eckdaten der PLC-CPUs für 810D, 840D".

---



## 2.10 Speicherbedarf des PLC-Grundprogramms für 840D

### Allgemeines

Das Grundprogramm besteht aus Basisfunktionen und optionalen Funktionen. Zu den **Basisfunktionen** gehört der zyklische Signalaustausch NC/PLC. Zu den **Optionen** gehören z.B. die FCs, die bei Bedarf eingesetzt werden können.

In der folgenden Tabelle ist der Speicherbedarf für die Basisfunktionen und die Optionen aufgelistet. Die Angaben stellen Richtwerte dar, sie sind vom jeweils aktuellen Softwarestand abhängig.

Speicherbedarf der Bausteine bei SINUMERIK 840D				
Baustein Typ, Nr.	Funktion	Bemerkung	Baustein-Größe (Byte)	
			Lade- speicher	Arbeits- speicher
<b>Grundprogramm-Basisfunktionen</b>				
FB 1, 15, 16, 17, 18		muss geladen werden	3616	3052
FC 1, 2, 3, 4, 11, 20		müssen geladen werden	7208	6608
DB 4, 5, 7, 8, 17, 19		müssen geladen werden	2490	966
DB 2, 3, 6		werden vom GP erzeugt	992	812
OB 1, 40, 100		müssen geladen werden	490	282
		Summe	14796	11720
<b>Nahtstelle PLC/NCK, PLC/HMI</b>				
DB 10	Signale PLC/NCK	muss geladen werden	318	262
DB 11	Signale PLC/BAG	wird vom GP erzeugt	80	44
DB 21, 30	Signale PLC/Kanal	werden vom GP abhängig von NC-MD erzeugt	je 352	je 316
DB 31, ...61	Signale PLC/Achse, Spindel	werden vom GP abhängig von NC-MD erzeugt	je 180	je 144
<b>Grundprogramm-Optionen</b>				
Maschinensteuertafel				
FC 19	Transfer MSTT-Signale, M-Variante	muss bei M-Variante der MSTT geladen werden	1498	1258
FC 25	Transfer MSTT-Signale, T-Variante	muss bei T-Variante der MSTT geladen werden	1358	1160
FC 24	Transfer MSTT-Signale, schmale Variante	muss bei schmaler-Variante der MSTT geladen werden	1358	1160
FC 26	Transfer MSTT-Signale, PHG-Variante	muss bei PHG geladen werden	1358	1160

<b>Grundprogramm-Optionen</b>				
FC 14	MPI/BTSS-Transfer	muss bei MCPNum > 0 geladen werden	942	802
Bedienhandgerät				
FC 13	Display-Steuerung BHG	kann bei Bedienhandgeräten geladen werden	1264	1044
Fehler-/Betriebsmeldungen				
FC 10	Erfassung FM/BM	laden bei Anwendung von FM/BM	1572	1350
ASUP				
FC 9	ASUP-Start	laden bei Anwendung von ASUPs von PLC	656	538
<b>Grundprogramm-Optionen</b>				
Konkurrierende Achsen				
FC 15	Positionierung von Linear-/Rundachsen	laden bei Achspositionierung von PLC	656	546
FC 16	Positionierung von Teilungsachsen	laden bei Achspositionierung von PLC	674	560
Stern-/Dreieck-Umschaltung				
FC 17	Stern-/Dreieck-Umschaltung für HSA	laden bei Stern-/Dreieck-Umschaltung	612	494
Spindelsteuerung				
FC 18	Spindelsteuerung	laden bei Spindelsteuerung von PLC	826	676
PLC-/NC-Kommunikation				
FB 2	NC-Variable lesen	laden bei NC-Variable lesen	396	224
DB n1)	NC-Variable lesen	ein Instanz-DB je FB 2-Aufruf	je 426	je 270
FB 3	NC-Variable schreiben	laden bei NC-Variable schreiben	396	224
DB m1)	NC-Variable schreiben	ein Instanz-DB je FB 3-Aufruf	je 426	je 270
FB 4	PI-Dienste	laden bei PI-Diensten	334	214
DB o1)	PI-Dienste	ein Instanz-DB je FB 4-Aufruf	je 234	je 130
DB 16	PI-Dienste Beschreibung	laden bei PI-Diensten	1190	408
FB5	GUD Variablen lesen	laden bei PI-Diensten	532	365
DB p	GUD Variablen lesen	ein Instanz-DB je FB 5-Aufruf	je 308	je 166
FB 6	Allgemeine Kommunikation	laden bei NC-Variable lesen, - schreiben und PI-Dienste	5986	5228
DB 15	Allgemeine Kommunikation	Instanz-DB für FB 6	440	172
Werkzeug-Verwaltung				
FC 6	Basis-Funktion	laden bei Option Werkzeugverwaltung	1382	1182
FC 7	Transfer-Funktion Revolver	laden bei Option Werkzeugverwaltung	530	430
FC 8	Transfer-Funktion	laden bei Option Werkzeugverwaltung	1002	834
FC 22	Richtungsauswahl	laden, wenn Richtungsauswahl benötigt	404	300
DB 71	Beladestellen	wird vom GP abhängig von NC-MD erzeugt	30*B	30*B
DB 72	Spindeln	wird vom GP abhängig von NC-MD erzeugt	48*Sp	48*Sp

Grundprogramm-Optionen				
DB 73	Revolver	wird vom GP abhängig von NC-MD erzeugt	44*R	44*R
DB 74	Basis-Funktion	wird vom GP abhängig von NC-MD erzeugt	(B+ Sp+R)*20	(B+ Sp+R)*20
Compile-Zyklen				
DB 9	Nahtstelle PLC-Compile-Zyklen	wird vom GP abhängig von NC-Option erzeugt	472	436
1): DB-Nummer ist vom PLC-Anwender vorzugeben				

**Beispiel:**

Anhand des in der vorigen Tabelle angegebenen Speicherbedarfs wurde für zwei Musterkonfigurationen der Speicherbedarf ermittelt (siehe folgende Tabelle).

Baustein Typ, Nr.	Funktion	Bemerkung	Baustein-Größe (Byte)	
			Lade- speicher	Arbeits- speicher
Minimal-Konfiguration (1 Spindel, 2 Achsen und T-MSTT)				
s.o.	Grundprogramm, Basis		14796	11720
	Nahtstellen-DBs		1290	1054
	MSTT		2300	1962
		Summe	18386	14736

Baustein Typ, Nr.	Funktion	Bemerkung	Baustein-Größe (Byte)	
			Lade- speicher	Arbeits- speicher
Maximal-Konfiguration (2 Kanäle, 4 Spindeln, 4 Achsen,T-MSTT)				
s.o.	Grundprogramm, Basis		14796	11720
s.o.	Nahtstellen-DBs		2542	2090
s.o.	MSTT		2300	1962
s.o.	Fehler-/Betriebsmeldungen		1572	1350
s.o.	ASUPs	1 ASUP-Anstoß	656	538
s.o.	Konkurrierende Achse	für 2 Revolver	674	560
s.o.	PLC-/NC-Kommunikation	1 x Variable lesen und 1 x Variable schreiben	8070	6388
s.o.	Werkzeug-Verwaltung	2 Revolver mit je 1 Beladestelle	3430	2854
s.o.	Compile-Zyklen		472	436
		Summe	34512	27898

## 2.11 Rahmenbedingungen und NC-VAR-Selektor

### 2.11.1 Rahmenbedingungen

#### 2.11.1.1 Programmier- und Parametrierwerkzeuge

##### Prinzip

##### Hardware

Für die bei 810D und 840D eingesetzten PLCs ist bei den Programmiergeräten oder PCs folgende Ausstattung erforderlich:

	Minimal	Empfehlung
Prozessor	80486	Pentium
RAM (MB)	32	oder mehr
Festplatte, freier Speicherplatz (MB)	200	> 400
Schnittstellen	MPI inkl. Kabel Memory-Card	
Grafik	VGA oder TIGA	SVGA
Maus	ja	
Betriebssystem	Windows 95 / 98 / NT  ab STEP7 Version 4	Windows 95 / 98 / NT oder höher ab STEP7 Version 5.1

Auf Geräten, die den oben aufgeführten Randbedingungen entsprechen, kann das erforderliche **STEP 7-Paket für S7-300** installiert werden, sofern es nicht bereits zum Lieferumfang des PGs gehört.

Mit diesem Paket sind folgende Funktionen möglich:

- Programmieren
  - Editoren und Compiler für AWL (kompletter Sprachumfang incl. SFB-/SFC-Aufrufe), KOP, FUP
  - Erstellen und Bearbeiten von Zuordnungslisten (Symbol-Editor)
  - DB-Editor
  - Ein- und Ausgabe von Bausteinen ON-/OFF-Line
  - Einfügen von Änderungen und Ergänzungen sowohl ON- als auch OFF-Line
  - Übertragen der Bausteine von PG nach PLC und zurück

- Parametrieren
  - Parametrierwerkzeug **HW Config** für CPU- und Peripherie-Parametrierung
  - Parametrierwerkzeug **Communication Configuration** für Einstellung der CPU-Kommunikations-Parameter
  - Ausgabe von Systemdaten wie Hardware-, Softwarestand, Speicherausbau, Peripherieausbau/-belegung
- Test und Diagnose (ON-LINE)
  - Status/Steuern Variable (Ein-/Ausgänge, Merker, DB Inhalte, etc.)
  - Status einzelner Bausteine
  - Anzeige von Systemzuständen (USTACK, BSTACK, SZL)
  - Anzeige von Systemmeldungen
  - PLC STOP/Neustart/Urlöschen auslösen von PG
  - PLC Komprimieren
- Dokumentation
  - Ausdruck einzelner oder aller Bausteine
  - Vergabe von symbolischen Namen (auch für Variable in DBs)
  - Ein- und Ausgabe von Kommentaren innerhalb jedes Bausteines
  - Ausdruck der Test und Diagnoseanzeigen
  - Hardcopy-Funktion
  - Querverweisliste
  - Programmübersicht
  - Belegungsplan E/A/M/T/Z/B/P/D
- Archivieren Dienstprogramme
  - Vergabe von Ausgabeständen einzelner Bausteine
  - Vergleichen von Bausteinen
  - Umverdrahten
  - Umsetzer STEP 5 -> STEP 7
- Optionspakete
  - Programmierung in S7-HIGRAPH, S7-GRAPH, SCL.  
Diese Pakete sind über den SIMATIC-Vertrieb bestellbar.
  - Zusatzpakete für die Projektierung von Baugruppen  
(z.B CP3425 -> NCM-Paket)

---

#### **Hinweis**

Weitergehende Informationen zu den möglichen Funktionen sind den SIMATIC-Katalogen und der STEP7 Dokumentation zu entnehmen.

---

### 2.11.1.2 Notwendige SIMATIC-Dokumentation

**Literatur:**

Systembeschreibung SIMATIC S 7  
Operationsliste S7-300, CPU 314, CPU 315-2DP  
Programmierung mit STEP 7  
Benutzerhandbuch STEP 7  
Programmierhandbuch STEP 7; Entwerfen von-Anwenderprogrammen  
Referenzhandbuch STEP 7; Anweisungsliste AWL  
Referenzhandbuch STEP 7; Kontaktplan KOP  
Referenzhandbuch STEP 7; Standard- und Systemfunktionen  
Handbuch STEP 7: Konvertieren von STEP 5 Programmen  
STEP 7 Gesamtindex  
Handbuch CPU 314, CPU 315-2DP

### 2.11.1.3 Relevante SINUMERIK-Dokumente

**Literatur:**

/IAD/ Inbetriebnahmeanleitung 840 D, 611 D; PLC-Nahtstelle  
/IAG/ Inbetriebnahmeanleitung 810 D, 611 D; PLC-Nahtstelle  
/BH/ Handbuch Bedienkomponenten (HW) / 840 D/ FM-NC/ 810 D  
/FB/ Funktionshandbuch 840 D, 810 D  
/LIS/ Listen 840 D/810 D  
/FBP/ PLC-C-Programmierung

## 2.11.2 NC-VAR-Selector

### 2.11.2.1 Übersicht

#### Allgemeines

Es ist ein Katalog über den Windows-Explorer mit beliebigem Katalognamen einzurichten. In diesen angelegten Katalog werden die selektierten Daten des VAR-Selektors abgespeichert (Daten.VAR und Daten.AWL Dateien). Anschließend ist über den STEP 7 Manager (Version 3 und höher) ein "Einfügen", "externe Quelle" für die Datei "Daten.AWL" in das STEP7 Maschinenprojekt durchzuführen. Hierzu ist im Manager der Quell-Container anzuwählen. Hiermit wird diese Datei in die Projektstruktur abgelegt. Nach der Übertragung der Datei sind diese AWL-Dateien mit STEP 7 zu übersetzen.

Mit der PC-Applikation "NC-VAR-Selector" werden Adressen von benötigten NC-Variablen beschafft und für den Zugriff im PLC-Programm (FB 2/FB 3) aufbereitet. Damit wird es einem PLC-Programmierer ermöglicht, NCK-Variablen aus dem Gesamtangebot der NCK-Variablen auszuwählen, diese Auswahl an Variablen abzuspeichern und sie mittels eines Code-Generators für den STEP7-Compiler aufzubereiten, um sie dann als ASCII-Datei (\*.AWL) im Maschinen-CPU-Programm abzuspeichern. Das Bild "NC-VAR-Selector" gibt eine Vorstellung dieser Bearbeitung.

**Hinweis**

Für jeden NC-Softwarestand (auch ältere Versionen) kann der neueste NC-VAR-Selector verwendet werden. Für ältere NC-Softwarestände können die Variablen auch aus der neuesten Gesamtliste selektiert werden. Der Informationsinhalt im DB 120 (Default DB für Variablen) ist nicht abhängig vom Softwarestand. D.h. selektierte Variablen in einem älteren Softwarestand müssen bei einer Softwarehochrüstung nicht neu selektiert werden.

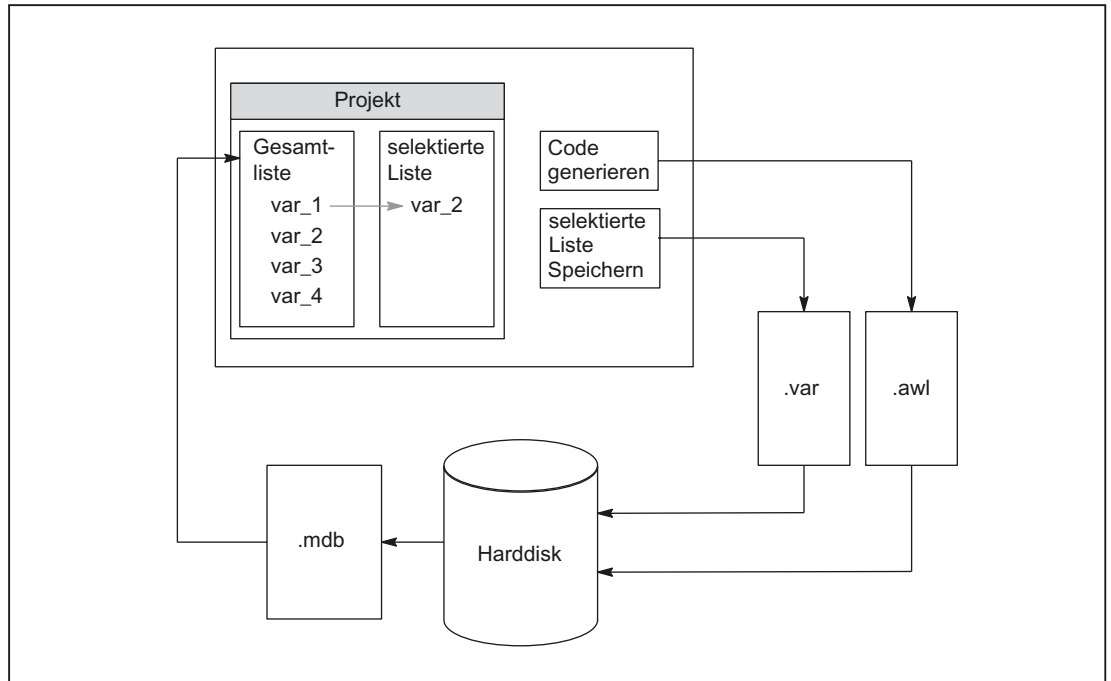


Bild 2-19 NC-VAR-Selector

Nach dem Starten der Applikation "NC-VAR-Selector" werden, nach Auswahl einer Variablenliste einer NC-Variante (Harddisk → File Ncv.mdb), alle in dieser Liste verfügbaren Variablen in einem Fenster angezeigt.

Es gibt ab SW 6.3 die Variablen Listen ncv*.mdb getrennt nach:	
Variablen der NC inklusive Maschinen-, Setting-Daten:	ncv_NcData.mdb
Maschinendaten des Antriebs 611D:	ncv_611d.mdb
Maschinendaten des LinearAntriebs 611D:	ncv_611dLinear.mdb
Maschinendaten des Antriebs 611D, Performance 2:	ncv_611d_P2.mdb
Maschinendaten des LinearAntriebs 611D, Performance 2:	ncv_611d_P2Linear.mdb
Maschinendaten des HydraulikAntriebs:	ncv_Hydraulics.mdb

Es können vom Bediener Variablen in eine zweite Liste (weiteres Fenster) übernommen werden. Diese selektierten Variablen können anschließend in einer ASCII-Datei abgelegt sowie als STEP 7-Source-Datei (.awl) aufbereitet und abgespeichert werden.

Mit der erzeugten STEP 7-Datei ist der PLC-Programmierer in der Lage, nach Generierung eines PLC-Datenbausteins durch den STEP 7 Compiler, NCK-Variablen zu lesen bzw. zu schreiben über die Grundprogramm-Funktionsbausteine "PUT" und "GET".

Die Liste der selektierten Variablen wird ebenfalls als ASCII-Datei abgelegt (Datei-Extension .var).

Die mit dem Werkzeug "NC-VAR-Selektor" mitgelieferte Variablenliste ist passend zu dem aktuellen NC-Softwarestand. In dieser Liste sind keine vom Anwender definierten Variablen (GUD-Variablen) enthalten. Diese Variablen werden durch das Grundprogramm durch den Funktionsbaustein FB 5 behandelt.

---

**Hinweis**

Mit der neuesten Version des "NC-VAR-Selectors" können alle vorherigen NC-Softwarestände bearbeitet werden. Es ist somit keine parallele Installation von verschiedenen Versionen des "NC-VAR-Selectors" nötig.

---

**Systemmerkmale, Rahmenbedingungen**

Die PC-Applikation "NC-VAR-Selektor" setzt WINDOWS 95 (oder höheres Betriebssystem) voraus.

Die Namensgebung der Variablen ist beschrieben in:

**Literatur:**

/LIS/ Listen; Kapitel: Variablen,  
bzw. auch in der Hilfedatei der Variablen (integriert im NC-VAR-Selektor)



## 2.11.2.2 Funktionsbeschreibung

### Übersicht

Folgendes Bild verdeutlicht den Umfang des NC-VAR-Selectors beim Einsatz in der STEP 7-Umgebung.

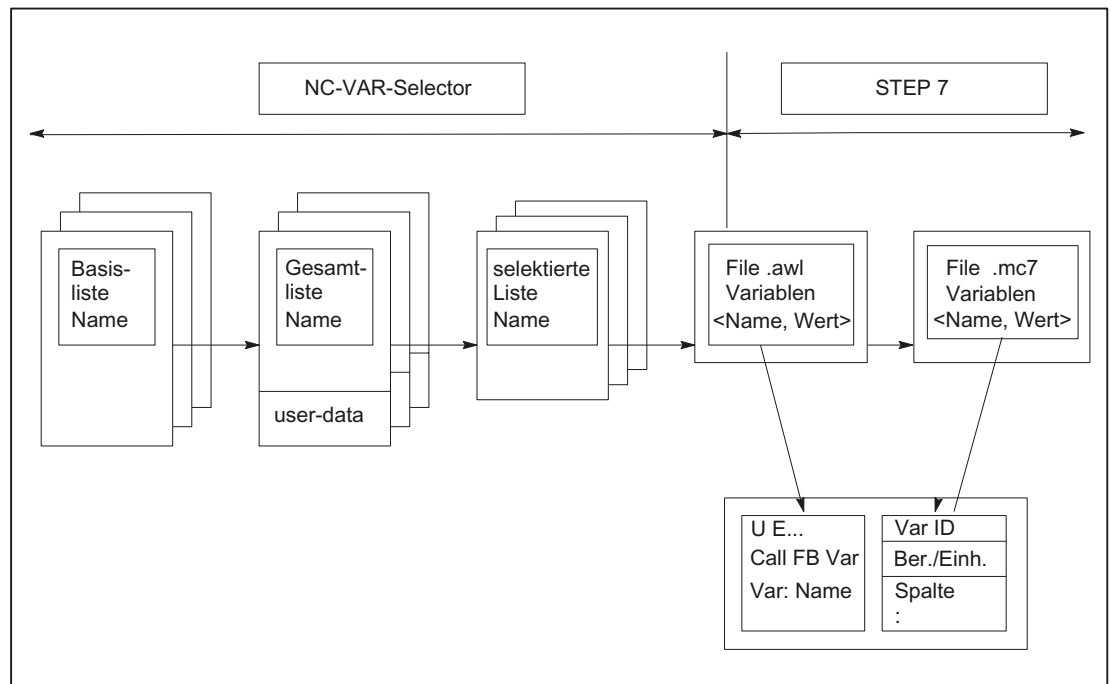


Bild 2-20 Einsatz des NC-VAR-Selectors in der STEP 7-Umgebung

Mit dem NC-VAR-Selector wird aus einer Variablenliste eine Liste selektierter Variablen erstellt und anschließend eine **.awl**-Datei erzeugt, die vom STEP 7-Compiler übersetzt werden kann.

### Hinweis

Ein **\*.awl**-File enthält sowohl die Namen bzw. ALIAS-Namen als auch die Informationen für die Adressparameter der NC-Variablen.

Ein daraus erzeugter Datenbaustein enthält nur noch die Adressparameter (10 Byte je Parameter).

- Die erzeugten Datenbausteine sind stets in der maschinenspezifischen Dateiablage entsprechend der STEP 7-Festlegungen zu speichern.
- Damit die Parametrierung der Bausteine GET/PUT (FB 2/3) bezüglich der NC-Adressen symbolisch erfolgen kann, ist der frei vergebbare symbolische Name des erzeugten Datenbausteins in der STEP 7-Symbolliste aufzunehmen.

## Grundbild / Grundmenü

Nach Anwahl (Starten) des NC-VAR-Selectors wird das Grundbild mit allen Bedienoptionen (obere Menüleiste) eingeblendet. Alle weiteren Fenster, die aufgeblendet werden, werden innerhalb des Gesamtfensters plaziert.

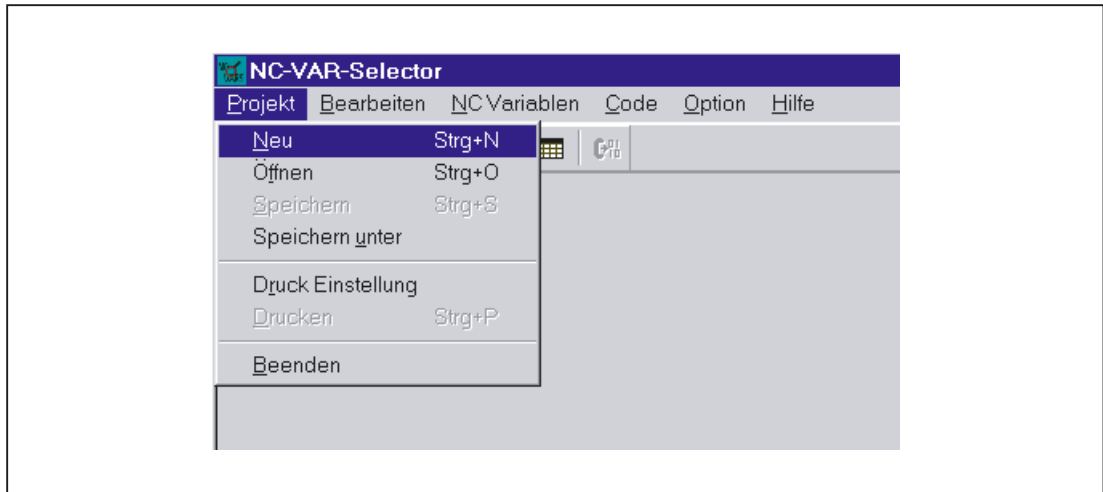


Bild 2-21 Grundbild mit Grundmenü

## Menüpunkt Projekt

Unter diesem Menüpunkt werden sämtliche Bedienhandlungen durchgeführt, die mit dem Projektfile (File der selektierten Variablen) zusammenhängen.

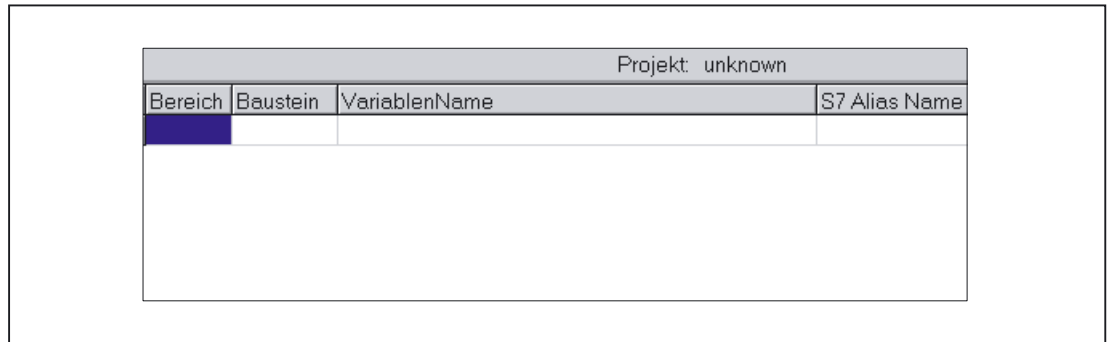
## Beenden der Applikation

Unter dem Menüpunkt "Projekt" wird über die Anwahl "Beenden" die Applikation beendet.

## Anlegen eines neuen Projekts

Über den Menüpunkt "Projekt" wird ein neues Projekt (neue Datei für selektierte Variablen) erstellt.

Die Anwahl "NEU" blendet ein Fenster für die selektierten Variablen auf. Danach wird nach einer Abfrage die Dateiauswahl für die NC Variablen Liste angeboten (nur wenn die NC Variablen Liste noch nicht geöffnet ist).



Projekt: unknown			
Bereich	Baustein	VariablenName	S7 Alias Name

Bild 2-22 Fenster für selektierte Variable bei neuem Projekt

Die selektierten Variablen werden in einem Fenster dargestellt.

### Öffnen eines bereits existierenden Projektes

Unter dem Menüpunkt "Projekt" kann über die Anwahl "Öffnen" ein bereits existierendes Projekt (bereits selektierte Variablen) geöffnet werden. Es erscheint ein File-Auswahl-Fenster, mit dem das entsprechende Projekt mit der Erweiterung ".var" ausgewählt werden kann.

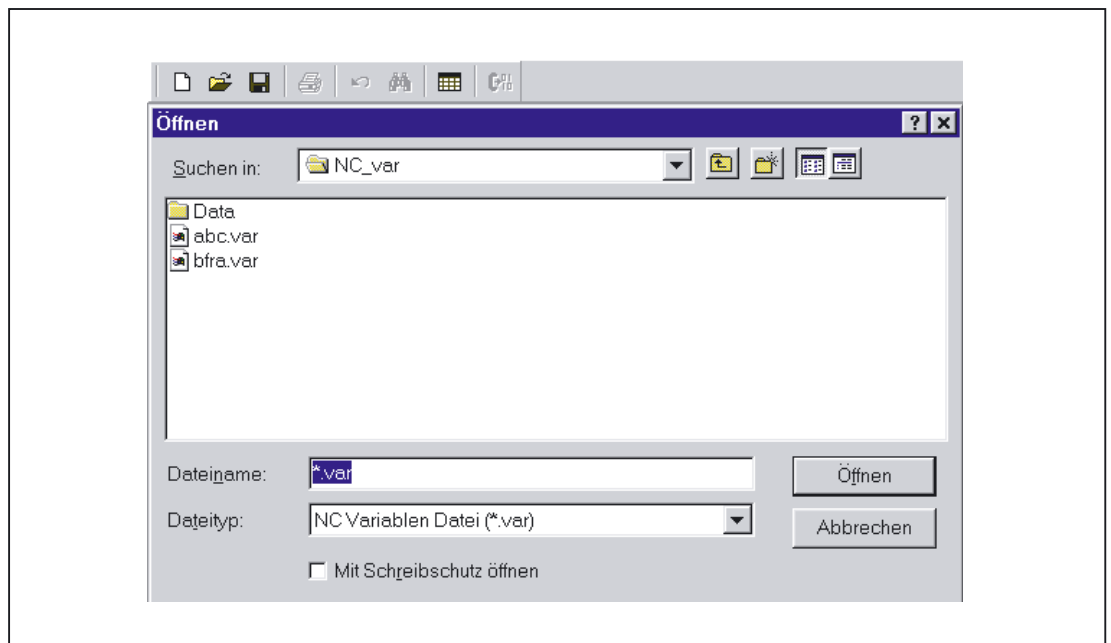


Bild 2-23 Auswahlfenster für vorhandene Projekte

Nach Auswahl des Projektes muss, falls Variable neu aufgenommen werden sollen, wiederum eine Gesamtliste von NCK-Variablen angewählt werden. Sollen nur Variable gelöscht werden, kann dies ohne Anwahl einer Gesamtliste erfolgen.

### Speichern eines Projektes

Über die Menüpunkte "Projekt", "Speichern" bzw. "Speichern unter ..." wird die Variablenliste gespeichert.

"Speichern" legt die Variablenliste unter einem bereits bekannten Pfad ab. Ist der Projektpfad nicht bekannt, ist das Verhalten wie bei "Speichern unter ...".

"Speichern unter ..." blendet ein Fenster auf, um den Pfad für das zu speichernde Projekt angeben zu können.

### Drucken eines Projektes

Unter dem Menüpunkt "Projekt" kann über die Auswahl "Drucken" eine Projektdatei ausgedruckt werden. Die Zeilen je Seite werden durch den Menüpunkt "Druck Einstellung" vorgenommen. Default sind 77 Zeilen.

### Menüpunkt Bearbeiten

Hier können z.B. folgende Bedienaktionen direkt durchgeführt werden:

- Variablen übernehmen
- Variablen löschen
- Aliasnamen ändern
- Variablen suchen

Weiterhin können diese Aktionen rückgängig gemacht werden.

### Rückgängig machen

Bedienaktionen die sich auf die Erstellung der Projektdatei beziehen (Variablen übernehmen, Variablen löschen, Aliasnamen ändern) können hier rückgängig gemacht werden.

### Menüpunkt NC Variablen

Die Ablage der Basisliste aller Variablen erfolgt unter dem NC Var Selector Pfad Data\Swxy (xy steht für SW Stand Nr., z.B. SW 5.3:=xy=53). Diese Liste kann als NC Variablen Liste angewählt werden. Die vorhandenen Variablenlisten sind thematisch vorhanden.

### Wählen einer NC Variablenliste

Mit dem Menüpunkt "NC Variablenliste", "Wählen" wird nun eine Liste der NC-Variablen einer NC-Version ausgewählt und angezeigt.

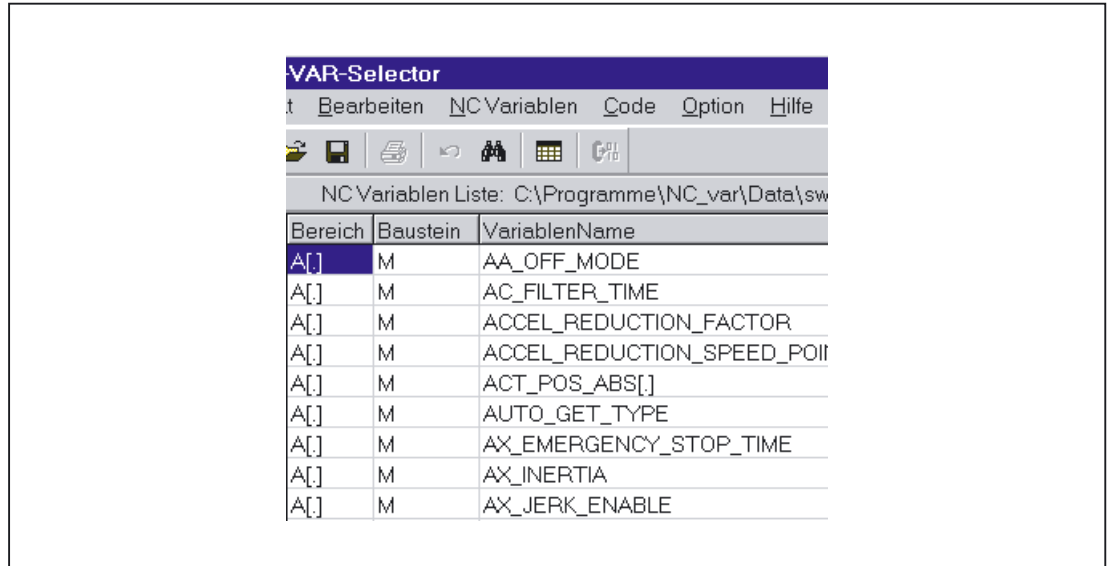


Bild 2-24 Fenster mit angewählter Gesamtliste

Die Feldvariablen (z.B. Achsbereich, T-Bereichsdaten usw.) werden mit Klammern ([.]) angedeutet. An dieser Stelle ist eine Zusatzinformation notwendig. Bei Übernahme der Variablen in die Projektliste wird die benötigte Zusatzinformation abgefragt.

### Teilmengen anzeigen

Mit Doppelklick auf ein beliebiges Tabellenfeld (Ausnahme: Variablenfeld !) wird ein Fenster eingeblendet, in dem Filterkriterien vorgegeben werden können.

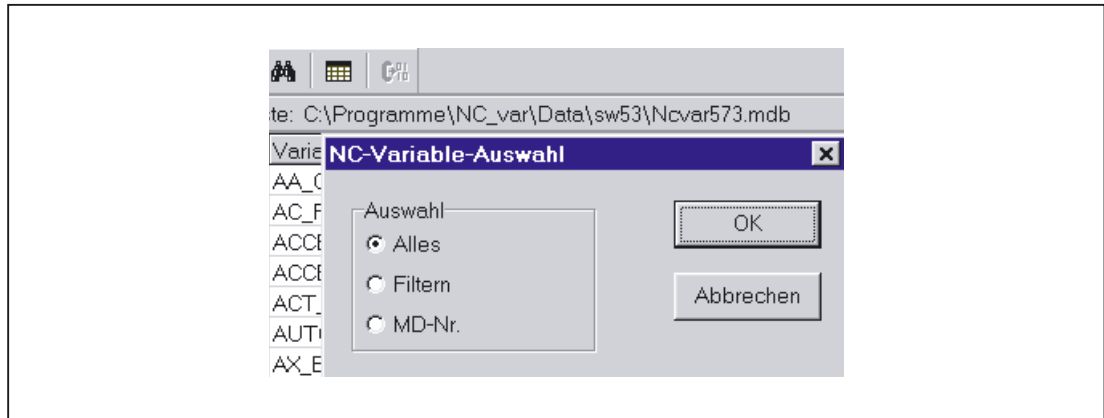


Bild 2-25 Fenster mit Filterkriterien für Anzeige der Variablenliste

Es stehen drei Optionen zur Verfügung:

- Alles anzeigen
- Bereich, Baustein und Name vorgeben (auch kombiniert)
- MD/SE-Daten-Nummer anzeigen

Es besteht auch die Möglichkeit, folgende Wildcards zu benutzen:

*	für eine beliebig lange Ergänzung des Suchkriteriums
---	--

### Beispiel für Suchkriterien

Suchkriterium Name: CHAN\* gefunden wird:

CHAN\_NAME  
chanAlarm  
chanStatus  
channelName  
chanAssignment

### Variablen selektieren

Eine Variable wird mittels einfachem Mausklick selektiert und mit einem Doppelklick in das Fenster der selektierten Variablen übernommen.

Unter dem Menüpunkt "Bearbeiten" kann diese Aktion auch wieder rückgängig gemacht werden.

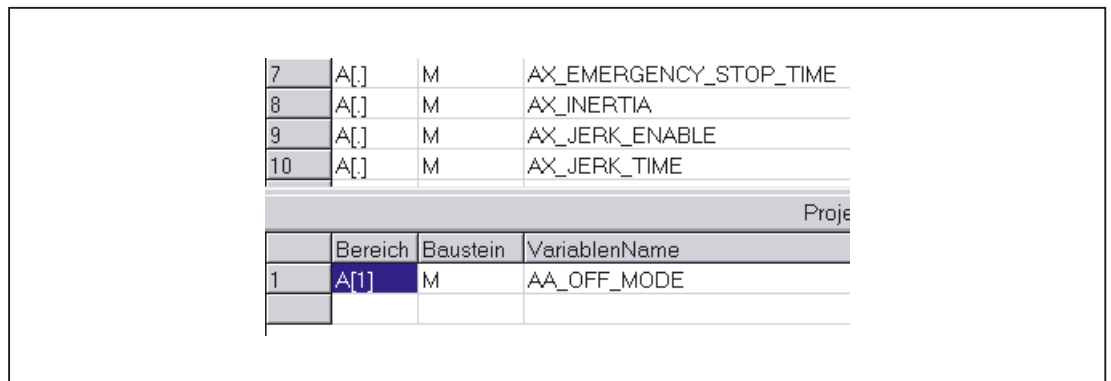
## Alias-Name

Die angebotenen Variablennamen können bis zu 32 Zeichen lang sein. Um Variablen im zu erzeugenden Datenbaustein eindeutig zu machen, wird der gewählte Name weiterhin um einige ASCII-Zeichen ergänzt. Der STEP 7-Compiler erkennt jedoch maximal 24-ASCII-Zeichen als eindeutige S7-Variable. Da nicht ausgeschlossen werden kann, dass Variablennamen sich nur in den letzten 8 Stellen unterscheiden, werden für die zu langen Namen **Alias**-Namen verwendet. Bei der Selektion einer Variablen wird deshalb geprüft, wie lang der zu verwendende S7-Name wird. Wird dieser länger als 24 Zeichen, muss über eine Zusatzeingabe ein vom Anwender vorzugebender Name (Alias-Name) eingesetzt werden.

**Hierbei muss der Anwender darauf achten, dass dieser eindeutig ist.**

Die Alias-Eingabe kann vom Anwender im Menü "Option" auch immer aktiviert werden. Die Alias-Eingabe ist dann bei jeder Übernahme einer Variablen möglich.

Weiterhin kann durch Doppelklick auf das S7-Variablennamen-Feld dieser Name nachträglich editiert werden. Unter dem Menüpunkt "Bearbeiten" kann diese Aktion auch wieder rückgängig gemacht werden.



7	A[.]	M	AX_EMERGENCY_STOP_TIME
8	A[.]	M	AX_INERTIA
9	A[.]	M	AX_JERK_ENABLE
10	A[.]	M	AX_JERK_TIME
Proje			
	Bereich	Baustein	VariablenName
1	A[1]	M	AA_OFF_MODE

Bild 2-26 Bild mit Gesamtliste und selektierten Variablen

## Scrollen

Können nicht alle Variablen im Fenster angezeigt werden, wird ein Scrollbar eingeblendet. Mit Scrollen (Page-Up/Down) können die restlichen Variablen erreicht werden.

## Variablen in mehrdimensionalen Strukturen

Werden Variable aus mehrdimensionalen Strukturen ausgewählt, so wird für die Adressierung dieser Variablen die Eingabe der Spalten- und/oder Zeilennummer sowie die Bereichs-Nummer abgefragt. Die erforderlichen Nummern können der NC-Variablen-Dokumentation entnommen werden.

### Literatur:

/LIS/ Listen; Variablen

Die Eingabe einer Null (0) als Bereichs-Nr., Zeilen- oder Spaltenindex bewirkt, dass die Variable im S7-AG als Zeiger auf diese Daten benutzt werden kann. Beim Lesen bzw. Schreiben dieser Daten über die Funktion "PUT" und "GET" müssen dann die optionalen Parameter "UnitX", "ColoumnX" und "LineX" mit den notwendigen Informationen aufgefüllt werden.

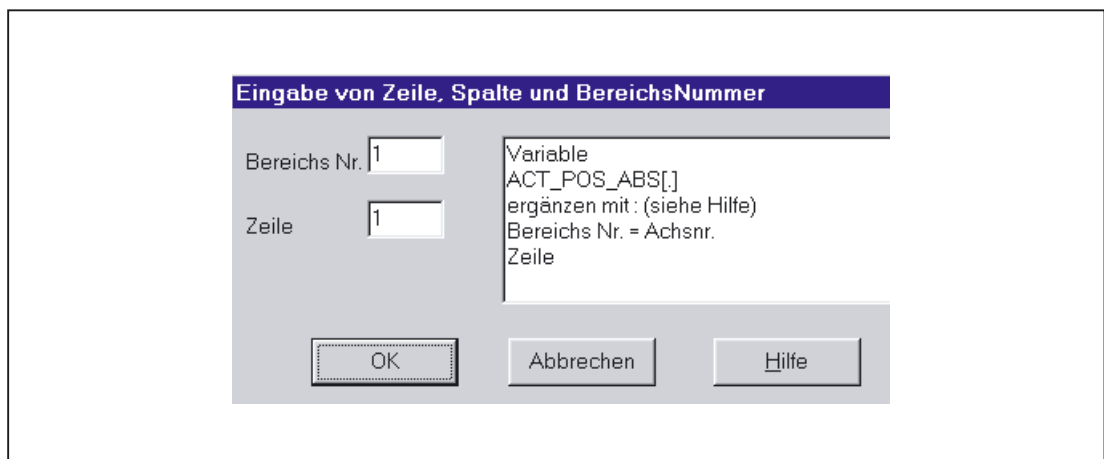


Bild 2-27 Eingabefeld für Zeile, Spalte und Baustein-Nr.

## Löschen von Variablen

Das Löschen von Variablen im Fenster der selektierten Variablen wird durch Auswahl (einfacher Mausklick) der Variablen und anschließendem Betätigen der Taste "Delete" ausgeführt. Für die Funktion Doppelklick gibt es keine Aktion. Eine Anwahl von mehreren Variablen zum Löschen ist möglich (siehe bei "Variablen selektieren").

Unter dem Menüpunkt "Bearbeiten" kann diese Aktion auch wieder rückgängig gemacht werden.

### Hinweis

Beim Löschen von Variablen ergibt sich eine Veränderung der Absolutadressen der Zeiger-Strukturen auf die Variablen. Es ist bei der Änderung der Variablen-Selektion also zwingend erforderlich, **vor der Änderung** eine oder mehrere **Textfiles aller Anwender-Bausteine zu erzeugen**. Nur so kann nach erneuter Compilierung sichergestellt werden, dass die Zuordnung der Variablen im FB "GET" oder FB "PUT" noch übereinstimmt.



## Selektierte Liste speichern

Nach erfolgter Variablenauswahl können diese unter einem Projektnamen angelegt werden. Die Ablage der Dateien erfolgt projektspezifisch.

Für die abzulegende Datei wird ein Fenster aufgeblendet, in welchem der Projektpfad und Name für diese Datei auszuwählen ist.

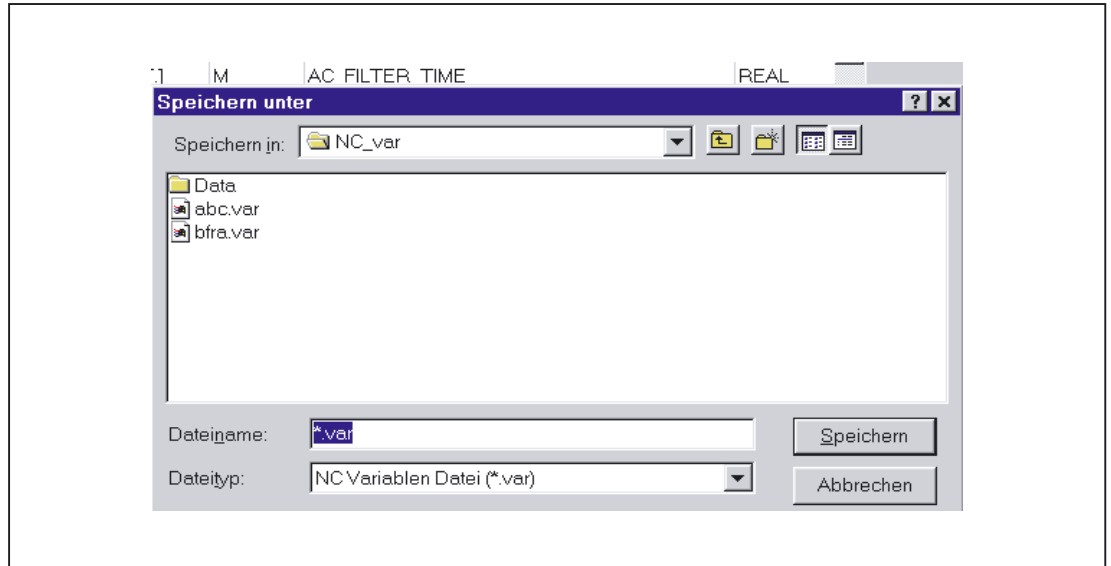


Bild 2-28 Fenster für Projektpfad und Name der abzulegenden Datei

## Code-Generierung

Unter diesem Menüpunkt stehen drei Unterpunkte zur Auswahl:

1. Einstellungen (Vorgabe der zu erzeugenden Datenbausteinnummer) und weitere Einstellungen
2. Generieren (Datenbaustein generieren)
3. In STEP7 Projekt (Übernahme des Datenbausteins in ein STEP7 Projekt)

## Einstellungen

Unter diesem Menüpunkt erfolgt die Eingabe der DB-Nummer und des Symbols für diese DB-Nummer, für den der Code erzeugt wird.

Unter dem Tabulator "Mass System" kann eine Auswahl getroffen werden wie Mass-Systembehaftete Variablen in der PLC verrechnet werden.

Unter dem Tabulator "Generierung" kann die Erzeugung für das jeweilige Zielsystem festgelegt werden.

## Generieren

Unter diesem Menüpunkt erfolgt die Einstellung der STEP 7-Datei aus der selektierten Variablenliste mit der Erweiterung ".awl".

Mit "Anwahl" wird eine Datei erzeugt:

Eine .awl-Datei, die als Input für den STEP 7-Compiler genutzt werden kann.

Für die zu speichernde Datei wird ein Fenster eingeblendet, in welchem Pfad und Name für die zu erzeugende .awl-Datei anzugeben ist.

## In STEP7 Projekt

Die generierte AWL-Datei wird in ein auswählbares SIMATIC Projekt (Programmpfad) übertragen und kompiliert. Zusätzlich kann auch das Symbol übernommen werden. Diese Funktion ist erst ab STEP7 Version 5.1 und NCVar Selektor 6.04.05 verfügbar. Dieser Vorgang dauert eine längere Zeit wegen Zeitbedarf in STEP7. Vor Übertragen eines neuen AWL-Files ist im KOP/FUP/AWL Editor das Dateifenster des AWL-Files zu schließen.

## Menüpunkt Option

Unter dem Menüpunkt "Option" kann gewählt werden:

- die aktuelle Sprache
- der Modus für die Alias-Eingabe (immer / > 24 Zeichen)

## Menüpunkt Hilfe

Hier kann durch Auswahl des entsprechenden Untermenüpunktes nachgelesen werden:

- die Bedienungsanleitung
- die Variablenbeschreibung

Weiterhin wird hier das Copyright und die Versionsnummer angezeigt.

### 2.11.2.3 Inbetriebnahme, Installation

Die Installation der Windows-Applikation "NC-VAR-Selector" erfolgt über das mitgelieferte SETUP-Programm.

## 2.12 Bausteinbeschreibungen

### 2.12.1 FB 1: RUN\_UP Grundprogramm, Anlaufteil

#### Funktionsbeschreibung

Im Anlauf erfolgt die Synchronisation zwischen NCK und PLC. Es werden die Datenbausteine für die Anwendernahtstelle NC/PLC anhand der über Maschinendaten festgelegten NC-Konfiguration erzeugt und die wichtigsten GP-Parameter auf Plausibilität geprüft. Im Fehlerfall übergibt der FB 1 eine Fehlerkennung in den Diagnosepuffer und überführt die PLC in den Stopp-Zustand.

Für den ordnungsgemäßen Hochlauf der Steuerung ist es erforderlich, die für sich autark hochlaufenden Systeme NCK, PLC zu synchronisieren. Deshalb ist der Anlauf derart konzipiert, dass die CPUs jeweils für sich "Teilanlauffunktionen" ausführen und der jeweils anderen CPU durch Austausch von Kennungen den korrekten Funktionsablauf mitteilen. Da der Anlauf asynchron erfolgt, ist es nicht vermeidbar, dass unter Umständen eine CPU "warten" muss, bis die andere vom Ablauf her "gleichgezogen" hat. Dieses wird automatisch durch das Grundprogramm realisiert.

Die PLC 314 und die PLC 315-2DP kennen nur die Anlaufart Neustart. Es existiert kein Wiederanlauf, d.h. das Betriebssystem durchläuft nach der Systeminitialisierung den Organisationsbaustein OB100 und beginnt den zyklischen Betrieb immer am Anfang des OB 1.

Der Anwender braucht nur die für ihn relevanten Parameter des FB zu versorgen. Die voreingestellten Werte im zugehörigen Instanz-DB 7 brauchen nicht zugewiesen zu werden. Der Baustein darf nur im OB 100 aufgerufen werden.

#### Ausgangsparameter

Die Ausgangsparameter im FB 1 dienen dem PLC-Anwender als Informationsquelle über die Konfiguration der Steuerung. Auf diese Daten kann auch im zyklischen Programmteil zugegriffen werden.

Hierzu gibt es 2 Möglichkeiten des Zugriffs:

1. Zugriff über einen Direktzugriff in den Datenbaustein DB7 (Instanz des FB 1) in symbolischer Form (z.B. L gp\_par.MaxChan, gp\_par ist hier der symbolische Name des DB 7).
2. Zuweisung eines Merker, Datenelements bei der Parametrierung des FB 1 an den zugehörigen Parameter (z. B. MaxChan:=MW 20). Anschließend kann im weiteren Anwenderprogramm die Information über die maximale Anzahl Kanäle über das Merkerwort 20 abgefragt werden.

---

#### Hinweis

Bei den **Bedienkomponenten**, die **an der MPI-Schnittstelle** angeschlossen werden, muss zusätzlich ein SDB210 über das STEP 7 Tool Communication Configuration erzeugt werden.

Die Vorgehensweise hierzu ist in der Inbetriebnahmeanleitung erläutert.

---

## Deklaration SINUMERIK 810D, 840D

Code	Kommentar
FUNCTION_BLOCK FB 1	
VAR_INPUT	
MCPNum :	INT:= 1; //0: keine MSTT //1: 1 MSTT (default) //2: 2 MSTT
MCP1In :	POINTER; //Anfangsadr. Eingangssignale MSTT 1
MCP1Out :	POINTER; //Anfangsadr. Ausgangssignale MSTT 1
MCP1StatSend :	POINTER; //Status-DW für Senden MSTT 1
MCP1StatRec :	POINTER; //Status-DW für Empfangen MSTT 1
MCP1BusAdr :	INT:= 6; //default
MCP1Timeout :	S5TIME:= S5T#700MS;
MCP1Cycl :	S5TIME:= S5T#200MS;
MCP2In :	POINTER; //Anfangsadr. Eingangssignale MSTT 2
MCP2Out :	POINTER; //Anfangsadr. Ausgangssignale MSTT 2
MCP2StatSend :	POINTER; //Status-DW für Senden MSTT 2
MCP2StatRec :	POINTER; //Status-DW für Empfangen MSTT 2
MCP2BusAdr :	INT;
MCP2Timeout :	S5TIME:= S5T#700MS;
MCP2Cycl :	S5TIME:= S5T#200MS;
MCPMPI :	BOOL:= FALSE;
MCP1Stop :	BOOL:= FALSE;
MCP2StopI :	BOOL:= FALSE;
MCP1NotSend :	BOOL:= FALSE;
MCP2NotSend :	BOOL:= FALSE;
MCPsDB210 :	BOOL:= FALSE;
MCPCopyDB77 :	BOOL:= FALSE;
MCPBusType :	BYTE = 0;
BHG :	INT; //Bedienhandgerät-Schnittstelle //0: kein BHG //1: BHG an MPI //2: BHG an BTSS
BHGIn :	POINTER; //Sendedaten des Bedienhandgerätes
BHGOut :	POINTER; //Empfangsdaten des Bedienhandgerätes
BHGStatSend :	POINTER; //Status-DW für Senden BHG
BHGStatRec :	POINTER; //Status-DW für Empfangen BHG
BHGInLen :	BYTE:= B#16#6; //Input 6 Byte
BHGOutLen :	BYTE:= B#16#14; //Output 20 Byte
BHGTimeout :	S5TIME:= S5T#700MS;
BHGCycl :	S5TIME:= S5T#100MS;
BHGRecGDNo :	INT:= 2;
BHGRecGBZNo :	INT:= 2;
BHGRecObjNo :	INT:= 1;

Code	Kommentar
BHGSendGDNo :	INT:= 2;
BHGSendGBZNo :	INT:= 1;
BHGSendObjNo :	INT:= 1;
BHGMPI :	BOOL:= FALSE;
BHGStop :	BOOL:= FALSE;
BHGNotSend :	BOOL:= FALSE;
NCCyclTimeout :	S5TIME:= S5T#200MS;
NCRunupTimeout :	S5TIME:= S5T#50S;
ListMDecGrp :	INT:=0;
NCKomm :	BOOL:= FALSE;
MMCToIF :	BOOL:= TRUE;
HWheelMMC :	BOOL:= TRUE; //Handradanwahlen durch HMI
MsgUser :	INT:= 10; //Anzahl der Userbereiche im DB2
UserIR :	BOOL:= FALSE; //Anwenderprogramme im OB40, //Lokaldatenerweiterung beachten!
IRAuxfuT :	BOOL:= FALSE; //T-Funktion in OB40 auswerten
IRAuxfuH :	BOOL:= FALSE; //H-Funktion in OB40 auswerten
IRAuxfuE :	BOOL:= FALSE; //DL-Funktion in OB40 auswerten
UserVersion :	POINTER; //Zeiger auf String-Variable, die im //Versionsbild angezeigt wird
END_VAR	
VAR_OUTPUT	
MaxBAG :	INT;
MaxChan :	INT;
MaxAxis :	INT;
ActivChan :	ARRAY[1..10] OF BOOL;
ActivAxis :	ARRAY[1..31] OF BOOL;
UDInt :	INT;
UDHex :	INT;
UDRea l :	INT;
END_VAR	

### Erläuterung der Formalparameter SINUMERIK 810D, 840D

Die folgende Tabelle zeigt alle Formalparameter der Funktion RUN\_UP für 810D, 840D:

Signal	Art	Typ	Wertebereich	Bemerkung
MCPNum	E	INT	0 bis 2	Anzahl der aktiven MSTT 0: keine MSTT vorhanden
MCP1In MCP2In	E	POINTER	E0.0 bis E120.0 oder M0.0 bis M248.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse für die Eingangssignale der betr. Maschinensteuertafel
MCP1Out MCP2Out	E	POINTER	A0.0 bis A120.0 oder M0.0 bis M248.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse für die Ausgangssignale der betr. Maschinensteuertafel
MCP1StatSend MCP2StatSend	E	POINTER	A0.0 bis A124.0, M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse für das Statusdoppelwort für das Senden an die Maschinensteuertafel: DW#16#08000000: Zeitüberwachung abgelaufen, sonst 0
MCP1StatRec MCP2StatRec	E	POINTER	A0.0 bis A124.0, M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse für das Statusdoppelwort für das Empfangen von der Maschinensteuertafel: DW#16#00000400: Zeitüberwachung abgelaufen, sonst 0
MCP1BusAdr MCP2BusAdr	E	INT	1...15	Busadresse der Maschinensteuertafel
MCP1Cycl MCP2Cycl	E	S5time	Empfehlung: 200 ms	Zeitraster für zyklische Aktualisierung der Signale an Maschinensteuertafel
MCPMPI	E	BOOL		1: alle Maschinensteuertafeln am MPI Bus angeschlossen (ohne GD Parametrierung)
MCP1Stop MCP2Stop	E	BOOL		0: Übertragung der Maschinensteuertafel-Signale starten 1: Übertragung der Maschinensteuertafel-Signale anhalten
MCP1NotSend MCP2NotSend	E	BOOL		0: Sende- und Empfangsbetrieb aktiviert 1: Nur Empfang der Maschinensteuertafel-Signale
MCPsDB210	E	BOOL		0: kein SDB210 für MCP 1: Bei SDB210 für MCP Zeitüberwachungen aktivieren

Signal	Art	Typ	Wertebereich	Bemerkung
MPCopyDB77	E	BOOL		1: Kopieren zwischen DB77 und MCP-Pointern am DB7. Nur anwendbar mit Standard SDB210 Projektierung auf DB77
MCPBusType	E	BYTE		0: MPI oder BTSS b#16#33: Profibus für MCP1 und MCP2 b#16#55: MSTT (ie Ethernet)
BHG	E	INT		Bedienhandgerät-Schnittstelle: 0: kein BHG 1: BHG an MPI mit SDB 210 Projektierung (für SW 3.x) 2: BHG an BTSS bzw. MPI, wenn zusätzlich der FB1-Parameter BHGMPI auf TRUE geschaltet ist (ab SW 4.x).
BHGIn	E	POINTER	E0.0 bis E124.0, M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse Empfangsdaten der PLC vom Bedienhandgerät
BHGOut	E	POINTER	A0.0 bis A124.0, M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse Sendedaten der PLC zum Bedienhandgerät
BHGStatSend	E	POINTER	A0.0 bis A124.0, M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse für das Statusdoppelwort für Senden an das BHG: DW#16#08000000: Zeitüberwachung abgelaufen, sonst 0
BHGStatRec	E	POINTER	A0.0 bis A124.0, M0.0 bis M252.0 oder DBn DBX0.0 bis DBXm.0	Anfangsadresse für das Statusdoppelwort fürs Empfangen vom BHG: DW#16#00000400: Zeitüberwachung abgelaufen, sonst 0
BHGInLen	E	BYTE	BHG default: B#16#6 (6 Byte)	Anzahl der Daten, die vom Bedienhandgerät empfangen werden
BHGOutLen	E	BYTE	BHG default: B#16#14 (20 Byte)	Anzahl der Daten, die an das Bedienhandgerät gesendet werden
BHGTimeout	E	S5time	Empfehlung: 700 ms	Zyklische Lebenszeichen-Überwachung für das Bedienhandgerät
BHGCycl	E	S5time	Empfehlung: 100 ms	Zeitraster für Zyklische Aktualisierung der Signale an das Bedienhandgerät
BHGRecGDNo	E	INT	BHG default: 2	Empfangs-GD-Kreisnr.
BHGRecGBZNo	E	INT	BHG default: 2	Empfangs-GBZ-Nr.
BHGRecObjNo	E	INT	BHG default: 1	Objektnummer für Empfangs-GBZ
BHGSendGDNo	E	INT	BHG default: 2	Sende-GD-Kreisnr.
BHGSendGBZNo	E	INT	BHG default: 1	Sende-GBZ-Nr.

Signal	Art	Typ	Wertebereich	Bemerkung
BHGSendObjNo	E	INT	BHG default: 1	Objektnummer für Sende-GBZ
BHGMPI	E	BOOL		1: Bedienhandgerät an MPI gekoppelt (ohne SDB 210 Proj.) Parameter BHG muss auf 2 gesetzt sein.
BHGStop	E	BOOL		0: Übertragung der Bedienhandgerät-Signale starten 1: Übertragung der Bedienhandgerät-Signale anhalten
BHGNotSend	E	BOOL		0: Sende und Empfangsbetrieb aktiviert 1: Nur Empfang der Bedienhandgerät-Signale ab SW 4
NCCyclTimeout	E	S5time	Empfehlung: 200 ms	Zyklische Lebenszeichen-Überwachung NCK
NCRunupTimeout	E	S5time	Empfehlung: 50 s	Hochlauf-Überwachung NCK
ListMDecGrp	E	INT	0...16	Aktivierung der erweiterten M-Gruppen-Decodierung 0: nicht aktiv 1...16: Anzahl der M-Gruppen
NCKomm	E	BOOL		PLC-NC-Kommunikationsdienste (FB 2/3/4/5/7: Put/Get/PI_SERV/GETGUD) TRUE: aktiv
MMCToIF	E	BOOL		Übertragung der MMC/HMI-Signale an die Nahtstelle (Betriebsarten, Programmbeeinflussung, usw.) TRUE: aktiv
HWheelMMC	E	BOOL		TRUE: Handradanwahl über MMC/HMI FALSE: Handradanwahl durch Anwenderprogramm
MsgUser	E	INT	0...32	Anzahl Userbereiche für Meldungen (DB 2)
UserIR	E	BOOL		Lokaldatenerweiterung OB 40 notwendig zur Verarbeitung der dortigen Signale vom Anwender
IRAuxfuT	E	BOOL		T-Funktion in OB 40 auswerten
IRAuxfuH	E	BOOL		H-Funktion in OB 40 auswerten
IRAuxfuE	E	BOOL		DL-Funktion in OB 40 auswerten
UserVersion	E	POINTER		Zeiger auf String-Variable. Die zugehörige Stringvariable wird im Versionsbild angezeigt (max. 41 Zeichen).
MaxBAG	A	INT	1..10	Anzahl BAGs
MaxChan	A	INT	1..10	Anzahl Kanäle
MaxAxis	A	INT	1..31	Anzahl Achsen



Signal	Art	Typ	Wertebereich	Bemerkung
ActivChan	A	ARRAY [1..10] OF BOOL		Bitleiste aktive Kanäle
ActivAxis	A	ARRAY [1..31] OF BOOL		Bitleiste aktive Achsen
UDInt	A	INT		Anzahl Integer-Maschinendaten im DB 20
UDHex	A	INT		Anzahl Hexadezimal- Maschinendaten im DB 20
UDReal	A	INT		Anzahl Real –Maschinendaten im DB 20

---

### Hinweis

Erläuterungen der Formalparameter dieser Funktion für die FM-NC.  
Für die Signale:

MCP1 StatRec (Maschinensteuertafel)  
MCP2 StatRec (Maschinensteuertafel)  
BHGStatRec (Bedienhandgerät)

gelten die Anfangsadressen für das Statusdoppelwort für das Empfangen von  
DW#16#00000400;

MCP1 Timeout (Maschinensteuertafel)  
MCP2 Timeout (Maschinensteuertafel)

werden als zyklische Lebenszeitüberwachung der Maschinensteuertafel 700 ms empfohlen.

---

### Überwachung MSTT/BHG (für 810D, 840D)

Für die Kommunikation mit den Maschinensteuertafeln werden im Fehlerfalle folgende Statusinformationen angezeigt:

Verfügbar in:	Bit Nr.	Beschreibung
MCP1StatRec MCP2StatRec BHGStatRec	10	Empfänger: Zeitüberwachung abgelaufen
nur SINUMERIK 840D: MCP1StatSend MCP2StatSend BHGStatSend	27	Sender: Zeitüberwachung abgelaufen

Außerdem wird ein Fehlereintrag im Diagnosepuffer der PLC generiert.

Daraus entstehen an der Bedienoberfläche die Fehlermeldungen:

- 400260: MSTT 1 ausgefallen oder
- 400261: MSTT 2 ausgefallen.
- 400262: BHG ausgefallen.

In diesem Falle werden die Eingangssignale von der MSTT bzw. vom Bedienhandgerät (MCP1In/MCP2In bzw. BHGIn) mit 0 initialisiert. Sollte eine Neusynchronisation zwischen PLC und MSTT/BHG möglich sein, wird die Kommunikation automatisch wieder aufgenommen und die Fehlermeldung vom GP gelöscht.

### Aufrufbeispiel für SINUMERIK 810D

Im Folgenden ist ein Aufrufbeispiel für den FB 1 im OB 100 aufgeführt. Dieses Beispiel ist Bestandteil der Diskette mit dem Grundprogramm für 810D.

```

ORGANIZATION_BLOCK OB 100
VAR_TEMP
    OB100_EV_CLASS :          BYTE;
    OB100_STRTUP :          BYTE;
    OB100_PRIORITY :        BYTE;
    OB100_OB_NUMBR :        BYTE;
    OB100_RESERVED_1 :      BYTE;
    OB100_RESERVED_2 :      BYTE;
    OB100_STOP :            WORD;
    OB100_RESERVED_3 :      WORD;
    OB100_RESERVED_4 :      WORD;
    OB100_DATE_TIME :       DATE_AND_TIME;
END_VAR
BEGIN
    CALL FB 1, DB 7(
                                MCPNum :=      1,
                                MCP1In :=      P#E0.0,

```

```

MCP1Out :=          P#A0.0,
MCP1StatSend :=     P#A8.0,
MCP1StatRec :=      P#A12.0,
MCP1BusAdr :=       14,
MCP1Timeout :=      S5T#700MS,
MCPMPI :=           TRUE,
NCCyclTimeout :=    S5T#200MS,
NCRunupTimeout :=   S5T#50S);

//HIER ANWENDERPROGRAMM EINFÜGEN
END_ORGANIZATION_BLOCK
```

### Aufrufbeispiel für SINUMERIK 840D

Im Folgenden ist ein Aufrufbeispiel für den FB 1 im OB 100 aufgeführt. Dieses Beispiel ist Bestandteil der Diskette mit dem Grundprogramm für 840D.

```

ORGANIZATION_BLOCK OB 100
VAR_TEMP
    OB100_EV_CLASS :          BYTE;
    OB100_STRTUP :          BYTE;
    OB100_PRIORITY :        BYTE;
    OB100_OB_NUMBR :        BYTE;
    OB100_RESERVED_1 :      BYTE;
    OB100_RESERVED_2 :      BYTE;
    OB100_STOP :            WORD;
    OB100_RESERVED_3 :      WORD;
    OB100_RESERVED_4 :      WORD;
    OB100_DATE_TIME :       DATE_AND_TIME;
END_VAR
BEGIN
    CALL FB 1, DB 7(
        MCPNum :=            1,
        MCP1In :=            P#E0.0,
        MCP1Out :=           P#A0.0,
        MCP1StatSend :=      P#A8.0,
        MCP1StatRec :=       P#A12.0,
        MCP1BusAdr :=        6,
        MCP1Timeout :=       S5T#700MS,
        MCP1Cycl :=          S5T#200MS,
        NC-CyclTimeout :=    S5T#200MS,
        NC-RunupTimeout :=   S5T#50S);

//HIER ANWENDERPROGRAMM EINFÜGEN
END_ORGANIZATION_BLOCK
```

## 2.12.2 FB 2: GET NC-Variable lesen

### Funktionsbeschreibung

Mit dem FB GET kann das Anwender PLC-Programm Variablen aus dem NCK-Bereich lesen. Dieser Funktionsbaustein ist Multi-Instanzfähig und zum FB 2 gehört ein Instanz-DB aus dem Anwenderbereich.

Durch Aufruf des FB 2 mit positivem Flankenwechsel am Steuereingang "Req" wird ein Auftrag gestartet, die durch Addr1 bis Addr8 referenzierten NC-Variablen zu lesen und nach erfolgtem Lesevorgang in die durch RD1 bis RD8 referenzierten PLC-Operandenbereiche zu kopieren. Der erfolgreiche Abschluss des Lesevorganges wird am Zustandsparameter NDR mit logisch "1" angezeigt.

Der **Lesevorgang** erstreckt sich über mehrere (in der Regel 1...2) PLC-Zyklen. Der Baustein kann nur im zyklischen Betrieb aufgerufen werden.

Eventuell aufgetretene Fehler werden über Error und State angezeigt.

Um die NC-Variablen zu referenzieren, werden zunächst alle benötigten Variablen mit dem Tool "NC-VAR-Selector" ausgewählt und in einen Datenbaustein als AWL-Quelle generiert. Für diesen DB muss dann in der Symbolliste ein Name vergeben werden. Als Aktualparameter der NCK Variablenadresse (Addr1...Addr8) wird "DB-Name.S7-Name" beim Aufruf von FB 2 übergeben.

### Variable Adressierung

Für einige NC-Variable ist es notwendig, im NC-VAR-Selector Bereichs-Nr. und/oder Zeile bzw. Spalte auszuwählen. Für diese Variablen ist es möglich, einen Basistyp auszuwählen, d.h. Bereich/Spalte/Zeile wird mit "0" vorbelegt.

Im FB wird der Inhalt der vom NC-VAR-Selector vorgegebenen Bereichs-Nr., Zeile und Spalte auf "0" geprüft. Liegt "0" vor, wird der Wert vom Eingangsparameter übernommen. Vor Aufruf des FB GET muss der Anwender den gewünschten Parameter versorgen "UnitX/ColumnX/LineX".

Hierbei entspricht Unit der Bereichs-Nr., Column der Spalte und Line der Zeile.

---

### Achtung

Der FB 2 kann NC-Variablen nur dann lesen, wenn der Grundprogramm-Parameter "NCKomm" auf 1 gesetzt wurde (in OB 100: FB 1, DB 7). Der Aufruf ist nur im zyklischen Programm OB 1 zulässig.

Beim Lesen von **kanalspezifischen** Variablen dürfen in einem Auftrag (FB 2-Aufruf) über Addr1 bis Addr8 nur Variablen von genau **einem** Kanal adressiert werden.

Bei den Bereichen V bzw. H dürfen nicht verschiedene logische Achsnummern in einem Auftrag zugeordnet werden (bei Nichteinhaltung: Error:= TRUE, State:= W#16#02).

---

In einem Auftrag können NC-Variablen innerhalb **einer** Gruppe kombiniert werden:

	Bereich				
Gruppe 1	C[1]	N	B	A	T
Gruppe 2	C[2]	N	B	A	T
Gruppe 3	V[.]	H[.]			

Für Kanal 3 bis Kanal 10 gelten die gleichen Regeln, wie in der vorstehenden Tabelle in Gruppe 1 und Gruppe 2 beispielhaft dargestellt wurden.

---

#### **Hinweis**

Speziell beim Lesen von mehreren langen Strings kann die Anzahl der nutzbaren Variablen geringer als 8 sein.

---

#### **Deklaration der Funktion**

```
FUNCTION_BLOCK FB 2
VAR_INPUT
    Req :                BOOL;
    NumVar :             INT;
    Addr1 :              ANY;
    Unit1 :              BYTE;
    Column1 :            WORD;
    Line1 :              WORD;
    Addr2 :              ANY;
    Unit2 :              BYTE;
    Column2 :            WORD;
    Line2 :              WORD;
    Addr3 :              ANY;
    Unit3 :              BYTE;
    Column3 :            WORD;
    Line3 :              WORD;
    Addr4 :              ANY;
    Unit4 :              BYTE;
    Column4 :            WORD;
    Line4 :              WORD;
    Addr5 :              ANY;
    Unit5 :              BYTE;
    Column5 :            WORD;
    Line5 :              WORD;
    Addr6 :              ANY;
    Unit6 :              BYTE;
```

```
Column6 :          WORD;
Line6   :          WORD;
Addr7   :          ANY;
Unit7   :          BYTE;
Column7 :          WORD;
Line7   :          WORD;
Addr8   :          ANY;
Unit8   :          BYTE;
Column8 :          WORD;
Line8   :          WORD;
FMNCno  :          INT;1)
END_VAR
VAR_OUTPUT
Error   :          BOOL;
NDR     :          BOOL;
State   :          WORD;
END_VAR
```

```
VAR_IN_OUT
RD1     :          ANY;
RD2     :          ANY;
RD3     :          ANY;
RD4     :          ANY;
RD5     :          ANY;
RD6     :          ANY;
RD7     :          ANY;
RD8     :          ANY;
END_VAR
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion GET.

Signal	Art	Typ	Wertebereich	Bemerkung
Req	E	BOOL		Auftragsstart mit positiver Flanke
NumVar	E	INT	1 bis 8 (entspricht Nutzung von Addr1 bis Addr8)	Anzahl zu lesender Variablen
Addr1 bis Addr8	E	ANY	[DBName].[VarName]	Variablenbezeichner aus <b>NC-VAR-Selector</b>
Unit1 bis Unit8	E	BYTE		Bereichsadresse, optional für variable Adressierung
Column1 bis Column8	E	WORD		Spaltenadresse, optional für variable Adressierung
Line1 bis Line8	E	WORD		Zeilenadresse, optional für variable Adressierung
Error	A	BOOL		Auftrag wurde negativ quittiert bzw. konnte nicht ausgeführt werden
NDR	A	BOOL		Auftrag wurde erfolgreich ausgeführt. Daten stehen zur Verfügung
State	A	WORD		siehe Fehlerkennungen
RD1 bis RD8	E/A	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	Zielbereich für gelesene Daten

## Fehlerkennungen

Konnte ein Auftrag nicht ausgeführt werden, wird dies am Zustandsparameter Error mit "logisch 1" angezeigt. Die Fehlerursache ist am Bausteinausgang State kodiert:

State		Bedeutung	Hinweis
WORT-H	WORT-L		
1 bis 8	1	Zugriffsfehler	im High-Byte Nummer der Var, bei der der Fehler auftrat
0	2	Fehler im Auftrag	falsche Zusammenstellung von Var. in einem Auftrag
0	3	negative Quittung, Auftrag nicht ausführbar	interner Fehler, evtl. Abhilfe: NC-RESET
1 bis 8	4	nicht ausreichend lokaler Anwenderspeicher zur Verfügung	gelesene Var. ist länger als in RD1 bis RD8 angegeben; im High-Byte Nummer der Var, bei der der Fehler auftrat
0	5	Formatwandlungsfehler	Fehler bei Wandlung vom Var.-Typ double: Var. liegt nicht im Bereich von S7-REAL
0	6	FIFO voll	Auftrag muss wiederholt werden, da die Warteschlange voll ist
0	7	Option nicht gesetzt	GP-Parameter "NCKomm" ist nicht gesetzt
1 bis 8	8	falscher Zielbereich (RD)	RD1 bis RD8 dürfen keine Lokaldaten sein
0	9	Übertragung belegt	Auftrag muss wiederholt werden
1 bis 8	10	Fehler bei variabler Adressierung	Unit oder Column/Line enthält den Wert 0
0	11	Variablenadresse ungültig	Addr (bzw. Variablenname), Area, Unit überprüfen
0	12	NumVar = 0	Parameter NumVar prüfen

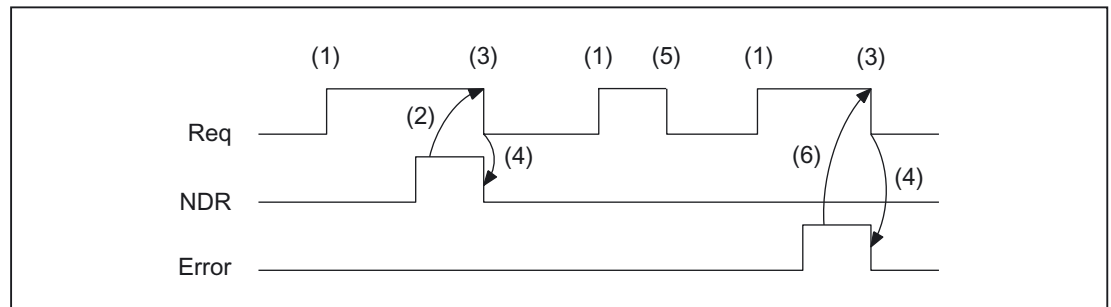
## Projektierungsschritte

Für das Lesen von NC-Variablen sind folgende Projektierungsschritte nötig:

- Auswahl der Variablen mit dem NC-VAR-Selector
- Speichern der ausgewählten Variablen in einer Datei \*.VAR im gewünschten Projektkatalog (\*.S7D)
- Erzeugen einer STEP 7-Quelldatei \*.AWL
- Erzeugen eines DBs mit den zugehörigen Adressangaben
- Eintrag des Symbols für den generierten DB in die Symboltabelle, damit im Anwenderprogramm symbolisch auf die Adressparameter zugegriffen werden kann
- Parametrierung des FB 2.



## Impulsdiagramm



- (1) Funktionsanstoß
- (2) Positive Quittung: Neue Daten empfangen
- (3) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FB
- (5) Wird Funktionsanstoß vor Erhalt der Quittung zurückgesetzt, werden die Ausgangssignale nicht aktualisiert, ohne Einfluss auf Ablauf der angestoßenen Funktion
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter State

## Aufrufbeispiel

Lesen von drei kanalspezifischen Maschinendaten von Kanal 1, deren Adressangaben im DB120 hinterlegt werden.

**Auswahl der Daten mit NC-VAR-Selector** und Speicherung in der Datei DB120.VAR; anschließend erzeugen der Datei DB120.AWL:

Bereich	Baustein	Name	Typ	Nr.	Byte	S7-Name
C[1]	M	MD20070 : AXCONF_MACHAX_USED[1]	char	20070	1	C1AxConfMachAxUsed1
C[1]	M	MD20070: AXCONF_MACHAX_USED[2]	char	20070	1	C1AxConfMachAxUsed2
C[1]	M	MD20090: SPIND_DEF_MASTER_SPIND	int	20090	1	C1SpindDefMasterSpind

Es wurden S7-(ALIAS-)Namen gewählt, um:

- die Kanal-Bezeichnung in den Namen aufzunehmen.  
und
- die Zeichen [ ] zu entfernen, die in einem STEP 7-Symbol nicht zulässig sind.

**Eintrag des Namens in die S7-SYMBOL-Tabelle** (z.B. NCVAR für DB 120):

Symbol	Operand	Datentyp
NCVAR	DB 120	DB 120

Die Datei DB120.AWL muss kompiliert und in die PLC übertragen werden.

**Parametrierung des FB 2 mit Instanz-DB 110:**

```

DATA_BLOCK DB 110          //freier Anwender-DB, als Instanz für FB 2
FB 2
BEGIN
END_DATA_BLOCK
Function FC "VariablenCall" : VOID
    U      E 7.7;          //freie Taste Maschinensteuertafel
    S      M 100.0;        //Req aktivieren
    U      M 100.1;        //Fertigmeldung NDR
    R      M 100.0;        //Auftrag beenden
    U      E 7.6;          //Fehlerquittierung von Hand
    U      M 102.0;        //Fehler steht an
    R      M 100.0;        //Auftrag beenden
    CALL FB 2, DB 110(
        Req :=          M 100.0,
        NumVar :=       3,                //3 Variablen lesen
        Addr1 :=        NCVAR.C1AxConfMachAxUsed1,
        Addr2 :=        NCVAR.C1AxConfMachAxUsed2,
        Addr3 :=        NCVAR.C1SpindDefMasterSpind,
        Error :=        M102.0,
        NDR :=          M100.1,
        State :=        MW104,
        RD1 :=          P#DB99.DBX0.0 BYTE 1,
        RD2 :=          P#DB99.DBX1.0 BYTE 1,
        RD3 :=          P#M110.0 INT 1);

```

**Beispiel: Variable Adressierung**

Lesen von zwei R-Parametern von Kanal 1, deren Adressangaben im DB 120 als Basistyp hinterlegt werden. Die R-Parameter-Nummer wird über den Parameter LineX parametrier.

```

DATA_BLOCK DB 120
VERSION : 0.0
STRUCT
    C1_RP_rpa0_0 :
    STRUCT
        SYNTAX_ID :          BYTE := B#16#82;
        bereich_u_einheit :  BYTE := B#16#41;
        spalte :            WORD := W#16#1;
        zeile :             WORD := W#16#0;
        bausteintyp :       BYTE := B#16#15;
        ZEILENANZAHL :      BYTE := B#16#1;
        typ :               BYTE := B#16#F;
        laenge :            BYTE := B#16#8;

```

```

        END_STRUCT;
END_STRUCT;
BEGIN
END_DATA_BLOCK
    CALL FB 2 , DB 110 (
        Req :=          M 0.0,
        NumVar :=       2,
        Addr1 :=        "NCVAR".C1_RP_rpa0_0,
        Line1 :=        W#16#1,
        Addr2 :=        "NCVAR".C1_RP_rpa0_0,
        Line2 :=        W#16#2,
        FMNCNo :=       1,
        Error :=        M 1.0,
        NDR :=          M 1.1,
        State :=        MW 2,
        RD1 :=          P#M 4.0 REAL 1,
        RD2 :=          P#M 24.0 REAL 1);

```

## Datentypen

Im NC-VAR-Selector werden die Datentypen der NCK bei den Variablen aufgeführt. In der folgenden Tabelle sind die Zuordnungen zu S7-Datentypen angegeben.

Zuordnung der Datentypen	
NCK-Datentyp	S7-Datentyp
double	REAL
float	REAL
long	DINT
integer	DINT
uint_32	DWORD
int_16	INT
uint_16	WORD
unsigned	WORD
char	CHAR oder BYTE
string	STRING
bool	BOOL

### 2.12.3 FB 3: PUT NC-Variable schreiben

#### Funktionsbeschreibung

Mit dem FB PUT kann das PLC-Anwenderprogramm Variablen im NCK Bereich schreiben.

Zu jedem FB 3-Aufruf muss ein separater Instanz-DB aus dem Anwender-Bereich zugeordnet werden. (Ab SW 3.7 Multi-Instanzfähig).

Durch Aufruf des FB 3 mit positivem Flankenwechsel am Steuereingang Req wird ein Auftrag gestartet, um die durch Addr1 bis Addr8 referenzierten NC-Variablen mit den Daten der lokal durch SD1 Bis SD8 referenzierten PLC-Operandenbereiche zu überschreiben. Der erfolgreiche Abschluss des Schreibvorgangs wird am Zustandsparameter "Done" mit logisch "1" angezeigt.

Der **Schreibvorgang** erstreckt sich über mehrere (in der Regel 1..2) PLC-Zyklen. Der Baustein kann nur im zyklischen Betrieb aufgerufen werden.

Eventuell aufgetretene Fehler werden über Error und State angezeigt.

Um die NC-Variablen zu referenzieren, werden zunächst alle benötigten Variablen mit dem Tool "NC-VAR-Selector" ausgewählt und in einem Datenbaustein als AWL-Quelle generiert. Für diesen DB muss dann in der Symbolliste ein Name vergeben werden. Als Aktualparameter der NCK Variablenadresse (Addr1...Addr8) wird "DB-Name.S7-Name" beim Aufruf von FB 3 übergeben.

#### Variable Adressierung

Für einige NC-Variable ist es notwendig, im NC-VAR-Selector Bereichs-Nr und/oder Zeile bzw. Spalte auszuwählen. Für diese Variablen ist es möglich, einen Basistyp auszuwählen, d.h. Bereich/Spalte/Zeile wird mit "0" vorbelegt.

Im FB wird der Inhalt der vom NC-VAR-Selector vorgegebenen Bereichs-Nr., Zeile und Spalte auf "0" geprüft. Liegt "0" vor, wird der Wert vom Eingangsparameter übernommen. Vor Aufruf des FB PUT muss der Anwender den gewünschten Parameter versorgen (UnitX/ColumnX/LineX).

Hierbei entspricht Unit der Bereichs-Nr., Column der Spalte und Line der Zeile.

## Maschinendaten, GUD

Um Maschinendaten und GUDs ohne Passwort beschreiben zu können, müssen die Schutzstufen des jeweiligen Datums auf die niedrigste Stufe undefiniert werden.

Das Vorgehen ist in der /IADC/ Inbetriebnahmeanleitung, "Schutzstufenkonzept" bzw. im Programmierhandbuch Arbeitsvorbereitung, "Schutzstufen für Anwenderdaten (GUD) definieren" beschrieben.

---

### Achtung

Der FB 3 kann NC-Variablen nur dann schreiben, wenn der Grundprogramm-Parameter "NCKomm" auf 1 gesetzt wurde (in OB 100: FB 1, DB 7). Der Aufruf ist nur im zyklischen Programm OB 1 zulässig.

Beim Schreiben von **kanalspezifischen** Variablen dürfen in einem Auftrag (FB 3-Aufruf) über Addr1 bis Addr8 nur Variablen von genau **einem** Kanal adressiert werden.

Bei den Bereichen V bzw. H dürfen nicht verschiedene logische Achsnummern in einem Auftrag zugeordnet werden (bei Nichteinhaltung: Error:= TRUE, State:= W#16#02).

---

In einem Auftrag können NC-Variablen innerhalb **einer** Gruppe kombiniert werden:

	Bereich				
Gruppe 1	C[1]	N	B	A	T
Gruppe 2	C[2]	N	B	A	T
Gruppe 3	V[.]	H[.]			

Für Kanal 3 bis Kanal 10 gelten die gleichen Regeln, wie in der vorstehenden Tabelle in Gruppe 1 und Gruppe 2 beispielhaft dargestellt wurden.

---

### Hinweis

Speziell beim Schreiben von mehreren langen Strings kann die Anzahl der nutzbaren Variablen geringer als 8 sein.

---

## Deklaration der Funktion

```
FUNCTION_BLOCK FB 3
VAR_INPUT
    Req :          BOOL;
    NumVar :       INT;
    Addr1 :        ANY;
    Unit1 :        BYTE;
    Column1 :      WORD;
    Line1 :        WORD;
    Addr2 :        ANY;
    Unit2 :        BYTE;
    Column2 :      WORD;
    Line2 :        WORD;
    Addr3 :        ANY;
    Unit3 :        BYTE;
    Column3 :      WORD;
    Line3 :        WORD ;
    Addr4 :        ANY;
    Unit4 :        BYTE;
    Column4 :      WORD;
    Line4 :        WORD;
    Addr5 :        ANY;
    Unit5 :        BYTE;
    Column5 :      WORD;
    Line5 :        WORD;
    Addr6 :        ANY;
    Unit6 :        BYTE;
    Column6 :      WORD;
    Line6 :        WORD;
    Addr7 :        ANY;
    Unit7 :        BYTE;
    Column7 :      WORD;
    Line7 :        WORD;
    Addr8 :        ANY;
    Unit8 :        BYTE;
    Column8 :      WORD;
    Line8 :        WORD;
END_VAR
VAR_OUTPUT
    Error :        BOOL;
    Done :         BOOL;
    State :        WORD;
END_VAR
VAR_IN_OUT
```

```

SD1 : ANY ;
SD2 : ANY ;
SD3 : ANY ;
SD4 : ANY ;
SD5 : ANY ;
SD6 : ANY ;
SD7 : ANY ;
SD8 : ANY ;
END_VAR

```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion PUT.

Signal	Art	Typ	Wertebereich	Bemerkung
Req	E	BOOL		Auftragsstart mit positiver Flanke
NumVar	E	INT	1 bis 8 (entspricht Nutzung von Addr1 bis Addr8)	Anzahl zu schreibender Variablen
Addr1 bis Addr8	E	ANY	[DBName].[VarName]	Variablenbezeichner aus <b>NC-VAR-Selector</b>
Unit 1 bis Unit 8	E	BYTE		Bereichsadresse, optional für variable Adressierung
Column 1 bis Column 8	E	WORD		Spaltenadresse, optional für variable Adressierung
Line 1 bis Line 8	E	WORD		Zeilenadresse, optional für variable Adressierung
Error	A	BOOL		Auftrag wurde negativ quittiert, bzw. konnte nicht ausgeführt werden
Done	A	BOOL		Auftrag wurde erfolgreich ausgeführt.
State	A	WORD		siehe Fehlerkennungen
SD1 bis SD8	E/A	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	zu schreibende Daten

## Fehlerkennungen

Konnte ein Auftrag nicht ausgeführt werden, wird dies am Zustandsparameter Error mit "logisch 1" angezeigt. Die Fehlerursache ist am Bausteinausgang State kodiert:

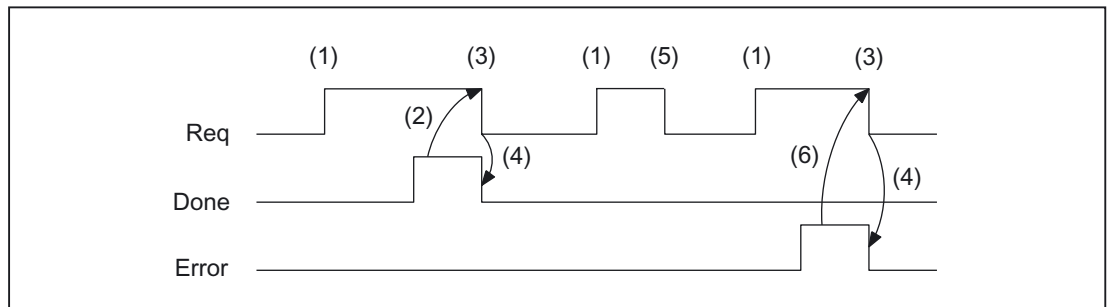
State		Bedeutung	Hinweis
WORT-H	WORT-L		
1 bis 8	1	Zugriffsfehler	im High-Byte Nummer der Var, bei der der Fehler auftrat
0	2	Fehler im Auftrag	falsche Zusammenstellung von Var in einem Auftrag
0	3	negative Quittung, Auftrag nicht ausführbar	interner Fehler, evtl. Abhilfe: NC-RESET
1 bis 8	4	Datenbereiche oder Datentypen stimmen nicht überein oder String ist leer	zu schreibende Daten in SD1 bis SD8 überprüfen; im High-Byte Nummer der Var, bei der der Fehler auftrat
0	6	FIFO voll	Auftrag muss wiederholt werden, da die Warteschlange voll ist
0	7	Option nicht gesetzt	GP-Parameter "NCKomm" ist nicht gesetzt
1 bis 8	8	falscher Zielbereich (SD)	SD1 bis SD8 dürfen keine Lokaldaten sein
0	9	Übertragung belegt	Auftrag muss wiederholt werden
1 bis 8	10	Fehler bei variabler Adressierung	Unit oder Column/Line enthält den Wert 0
0	11	Variablenadr. ungültig oder Var. nur lesbar	Addr (bzw. Variablenname), Area, Unit überprüfen
0	12	NumVar = 0	Parameter NumVar prüfen

## Projektierungsschritte

Für das Schreiben von NC-Variablen sind die gleichen Projektierungsschritte wie für das Lesen von NC-Variablen notwendig. Es ist zweckmäßig, die Adressangaben aller NC-Variablen, die gelesen oder geschrieben werden sollen, in einem DB zu hinterlegen.



## Impulsdiagramm



- (1) Funktionsanstoß
- (2) Positive Quittung: Variablen wurden geschrieben
- (3) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FB
- (5) Wird Funktionsanstoß vor Erhalt der Quittung zurückgesetzt, werden die Ausgangssignale nicht aktualisiert, ohne Einfluss auf Ablauf der angestoßenen Funktion
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter State

### Aufrufbeispiel

Schreiben von drei kanalspezifischen Maschinendaten von Kanal 1:

**Auswahl der drei Daten mit NC-VAR-Selector** und Speicherung in der Datei DB120.VAR:

Bereich	Baustein	Name	Typ	Byte	S7-Name
C[1]	RP	rpa[5]	double	4	rpa_5C1RP
C[1]	RP	rpa[11]	double	4	rpa_11C1RP
C[1]	RP	rpa[14)	double	4	rpa_14C1RP

**Eintrag NCVAR für DB 120 mit dem S7-SYMBOL-Editor:**

Symbol	Operand	Datentyp
NCVAR	DB120	DB 120

Die Datei DB120.AWL muss kompiliert und in die PLC übertragen werden.

**Aufruf und Parametrierung des FB 3 mit dem Instanz-DB 111:**

```

DATA_BLOCK DB 111           //freier Anwender DB, als Instanz für FB 3
FB 3
BEGIN
Function FC "VariablenCall" : VOID
END_DATA_BLOCK
    U    E 7.7;              //freie Taste Maschinensteuertafel
    S    M 100.0;            //Req aktivieren
    U    M 100.1;            //Fertigmeldung Done
    R    M 100.0;            //Auftrag beenden
    U    E 7.6;              //Fehlerquittierung von Hand
    U    M 102.0;            //Fehler steht an
    R    M 100.0;            //Auftrag beenden
    CALL FB 3, DB 111(
        Req :=      M 100.0,
        NumVar :=    3,                //3 Variablen schreiben
        Addr1 :=    NCVAR.rpa_5C1RP,
        Addr2 :=    NCVAR.rpa_11C1RP,
        Addr3 :=    NCVAR.rpa_14C1RP,
        FMNCno :=   1,
        Error :=    M102.0,
        Done :=     M100.1,
        State :=    MW104,
        SD1 :=      P#DB99.DBX0.0 REAL 1,
        SD2 :=      P#DB99.DBX4.0 REAL 1,
        SD3 :=      P#M110.0 REAL 1);
    
```

### Beispiel: Variable Adressierung

Schreiben von zwei R-Parametern von Kanal 1, deren Adressangaben im DB 120 als Basistyp hinterlegt werden. Die R-Parameter-Nummer wird über den Parameter LineX parametrisiert.

```
DATA_BLOCK DB 120
VERSION : 0.0
STRUCT
  C1_RP_rpa0_0:
  STRUCT
    SYNTAX_ID :          BYTE := B#16#82;
    bereich_u_einheit :  BYTE := B#16#41;
    spalte :            WORD := W#16#1;
    zeile :             WORD := W#16#0;
    bausteintyp :       BYTE := B#16#15;
    ZEILENANZAHL :      BYTE := B#16#1;
    typ :               BYTE := B#16#F;
    laenge :            BYTE := B#16#8;
  END_STRUCT ;
END_STRUCT ;
BEGIN
END_DATA_BLOCK
CALL FB 3 , DB 122 (
  Req :=          M 10.0,
  NumVar :=       2,
  Addr1 :=        "NCVAR".C1_RP_rpa0_0,
  Line1 :=        W#16#1,
  Addr2 :=        "NCVAR".C1_RP_rpa0_0,
  Line3 :=        W#16#2
  Error :=        M 11.0,
  Done :=         M 11.1,
  State :=        MW 12,
  SD1 :=          P#M 4.0 REAL 1,
  SD2 :=          P#M 24.0 REAL 1);
```

## 2.12.4 FB 4: PI\_SERV Allgemeine PI-Dienste

### Funktionsbeschreibung

Mit dem FB PI\_SERV können Programminstanz-Dienste im NCK Bereich gestartet werden.

Durch Anforderung über den PI-Dienst wird im NCK ein Programmteil abgearbeitet, welches eine bestimmte Funktion ausübt (z.B. Leerplatzsuche in einem Magazin bei Werkzeugverwaltung).

Zu jedem FB 4-Aufruf muss ein separater Instanz-DB aus dem Anwender-Bereich zugeordnet werden. Die Dokumentation zur Anwendung der Multi-Instanzfähigkeit ist in den STEP 7-Beschreibungen enthalten.

Über Parameter "PIService" wird der spezifizierte Dienst referenziert. Über die frei zuordenbaren zusätzlichen Eingangsvariablen mit unterschiedlichen Datentypen (Addr1 ... Addr4 für Strings, WVar1 ... WVar10 für Integer oder Word Variablen) wird die Versorgung des angewählten PI-Dienstes durchgeführt.

Durch **Aufruf des FB 4** mit positivem Flankenwechsel am Steuereingang Req wird ein Auftrag gestartet. Die erfolgreiche Durchführung wird am Zustandsparameter "Done" mit logisch 1 angezeigt. Eventuell aufgetretene Fehler werden über Error und State angezeigt. Der Datenbaustein "PI" (DB 16) enthält interne Beschreibungen der möglichen PI-Dienste. Für diesen DB muss in der Symbolliste ein Name vergeben werden. Als Aktualparameter für "PIService" wird "DB-Name.PI-Name" beim Aufruf des FB 4 übergeben.

Die Ausführung des PI-Dienstes erstreckt sich über mehrere in der Regel (1..2) PLC-Zyklen. Der Baustein kann nur im zyklischen Betrieb aufgerufen werden.

---

### Hinweis

Der FB4 kann PI-Dienste nur dann starten, wenn der Grundprogramm-Parameter "NCKomm" auf 1 gesetzt wurde (in OB 100: FB 1, DB 7). Der Aufruf ist nur im zyklischen Programm OB 1 zulässig.

---

### Deklaration der Funktion

```
FUNCTION_BLOCK FB 4
VAR_INPUT
    Req :                BOOL;
    PIService :          ANY;
    Unit :               INT;
    Addr1 :              ANY;
    Addr2 :              ANY;
    Addr3 :              ANY;
    Addr4 :              ANY;
    WVar1 :              WORD;
    WVar2 :              WORD;
    WVar3 :              WORD;
    WVar4 :              WORD;
```

```

WVar5 :          WORD;
WVar6 :          WORD;
WVar7 :          WORD;
WVar8 :          WORD;
WVar9 :          WORD;
WVar10 :         WORD;
FMNCNo :         INT;          //(nur bei FM-NC)
END_VAR
VAR_OUTPUT
  Error :        BOOL;
  Done :         BOOL;
  State :        WORD;
END_VAR

```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion PI\_SERV.

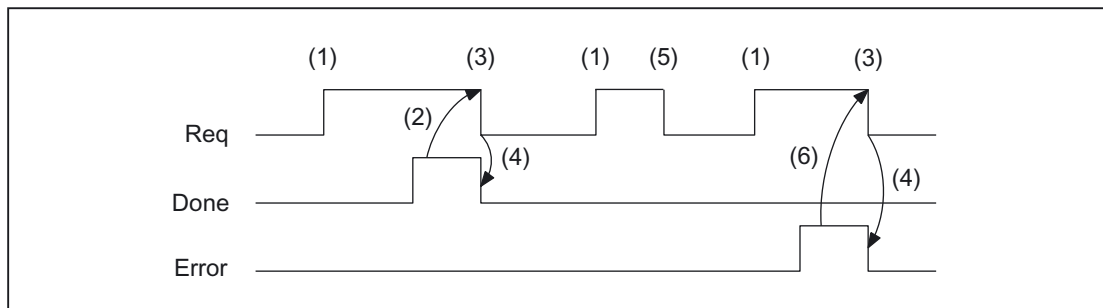
Signal	Art	Typ	Wertebereich	Bemerkung
Req	E	BOOL		Auftragsanforderung
PIService	E	ANY	[DBName].[VarName] Standard ist: "PI".[VarName]	PI-Dienst Beschreibung <sup>1)</sup>
Unit	E	INT	1...	Bereichsnummer
Addr1 bis Addr4	E	ANY	[DBName].[VarName]	Referenz auf Strings Spezifikation gemäß angewähltem PI-Dienst
WVar1 bis WVar10	E	WORD	1...	Integer oder Word-Variablen. Spezifikation gemäß angewähltem PI- Dienst
Error	A	BOOL		Auftrag wurde negativ quittiert bzw. konnte nicht ausgeführt werden
Done	A	BOOL		Auftrag wurde erfolgreich ausgeführt
State	A	WORD		siehe Fehlerkennungen
1) s. README-Datei auf Grundprogramm-Auslieferungsdiskette				

### Fehlerkennungen

Konnte ein Auftrag nicht ausgeführt werden, wird dies am Zustandsparameter Error mit "logisch 1" angezeigt. Die Fehlerursache ist am Bausteinausgang State kodiert:

State	Bedeutung	Hinweis
3	negative Quittung, Auftrag nicht ausführbar	interner Fehler, evtl. Abhilfe: NC-RESET
6	FIFO voll	Auftrag muss wiederholt werden, da die Warteschlange voll ist
7	Option nicht gesetzt	GP-Parameter "NCKomm" ist nicht gesetzt
9	Übertragung belegt	Auftrag muss wiederholt werden

### Impulsdiagramm



- (1) Funktionsanstoß
- (2) Positive Quittung: PI-Dienst wurde ausgeführt
- (3) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FB
- (5) Wird Funktionsanstoß vor Erhalt der Quittung zurückgesetzt, werden die Ausgangssignale nicht aktualisiert, ohne Einfluss auf Ablauf der angestoßenen Funktion
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter State

### 2.12.4.1 Überblick verfügbarer PI-Dienste

#### Überblick PI-Dienste

Im folgenden Abschnitt wird ein Überblick über die PI-Dienste, die von PLC aus startbar sind, gegeben. Die Verwendung und Bedeutung der allgemeinen Eingangsvariablen des FB 4 (Unit, Addr ..., WVar ...) ist vom jeweiligen PI-Dienst abhängig.

PI-Dienst	Funktion	verfügbar SINUMERIK	
		FM-NC	810/840D
ASUP	Interrupt zuordnen	*	*
CANCEL	Cancel durchführen	*	*
CONFIG	Umkonfiguration von gekennzeichneten Maschinendaten	*	*
DIGION	Digitalisieren ein		*
DIGIOF	Digitalisieren aus		*
FINDBL	Suchlauf aktivieren	*	*
LOGIN	Kennwort aktivieren	*	*
LOGOUT	Kennwort rücksetzen	*	*
NCRES	NC RESET auslösen		*
SELECT	Programm zur Abarbeitung für einen Kanal anwählen	*	*
SETUDT	Setzt die aktuellen User Daten aktiv		*
PI-Dienst	<b>Werkzeugverwaltungs-Funktion</b>		
CRCEDN	Lege neue Schneide an		*
CREACE	Werkzeugschneide anlegen	*	*
CREATO	Werkzeug erzeugen	*	*
SETUFR	Anwender-Frames aktivieren	*	*
DELECE	Löschen einer Werkzeugschneide		*
DELETO	Werkzeug löschen	*	*
MMCSEM	Semaphoren für verschiedene PI-Dienste		*
TMCRT0	Werkzeug anlegen		*
TMFDPL	Leerplatzsuche zum Beladen		*
TMFPBP	Leerplatzsuche		*
TMGETT	T-Nummer zum vorhergehenden Werkzeugbezeichner mit Duplonummer		*
TMMVTL	Magazinplatz zum Beladen bereitstellen, Werkzeug entladen		*
TMPOSM	Magazinplatz oder Werkzeug positionieren		*
TMPCIT	Inkrementwert für Stückzahlzähler setzen		*
TMRASS	Rücksetzen des Aktiv-Status		*
TRESMO	Rücksetzen der Überwachungswerte		*
TSEARC	Komplexes Suchen über Suchmasken		*
x: PI-Dienst ist verfügbar			

### 2.12.4.2 Allgemeine PI-Dienste

Die möglichen Dienste sind unter diesem Kapitel beschrieben.

#### PI-Dienst: ASUP

##### Funktion Interrupt zuordnen:

Ein auf dem NCK abgelegtes Programm wird einem Interruptsignal für einen Kanal zugeordnet. Dies ist nur möglich, wenn das File ausgeführt werden darf. Die Pfad-Namen und Programm-Namen sind, wie im Programmierhandbuch Arbeitsvorbereitung, Datei- und Programmverwaltung Kapitel "Programmspeicher" beschrieben, einzugeben. Für die Schreibweise von Pfad- und Programmnamen siehe auch Beispiel des FB 4.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.ASUP	Interrupt zuordnen
Unit	INT	1 bis 10	Kanal
WVar1	WORD	1 bis 8	Interrupt-Nummer
WVar2	WORD	1 bis 8	Priorität
WVar3	WORD	0/1	LIFTFAST
WVar4	WORD	0/1	BLSYNC
Addr1	STRING		Pfadname
Addr2	STRING		Programmname

##### Hinweis

Für die Zuweisung wird ebenso die SETINT-Anweisung benutzt.

Der PI-Dienst ASUP darf nur im RESET-Zustand des zu aktivierenden Kanals ausgeführt werden.

##### Literatur:

/PGA/ Programmierhandbuch Arbeitsvorbereitung, Flexible NC-Programmierung Kaptitel "Interruptroutine (SETINT, DISABLE, ENABLE, CLRINT)".

#### PI-Dienst: CANCEL

##### Funktion Cancel durchführen:

Das Kommando CANCEL bewirkt, dass die Funktion Cancel (entsprechend der Taste am HMI) durchgeführt wird.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CANCEL	Cancel



## PI-Dienst: CONFIG

### Funktion Umkonfiguration:

Das Kommando Umkonfiguration bewirkt, dass Maschinendaten, die sequentiell vom Bediener oder auch PLC eingegeben wurden, quasi parallel aktiviert werden. Das Kommando kann nur im RESET-Zustand der Steuerung bzw. bei Programmunterbrechung (NC-Stopp an Satzgrenze) aktiviert werden. Bei Nichteinhaltung dieser Bedingung erfolgt eine Fehlerrückmeldung des FB 4 (State = 3).

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CONFIG	Umkonfiguration
Unit	INT	1	
WVar1	INT	1	Klassifizierung

## PI-Dienst: DIGION

### Funktion Digitalisieren ein:

Digitalisieren im vorgegebenen Kanal anwählen.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.DIGION	Digitalisieren ein
Unit	INT	1 bis 10	Kanal

## PI-Dienst: DIGIOF

### Funktion Digitalisieren aus:

Digitalisieren im vorgegebenen Kanal ausschalten.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.DIGIOF	Digitalisieren aus
Unit	INT	1 bis 10	Kanal

### PI-Dienst: FINDBL

#### Funktion Suchlauf aktivieren:

Ein Kanal wird auf Suchlaufmodus geschaltet und daraufhin die Quittung gesendet. Der Suchlauf wird danach vom NCK sofort ausgeführt. Der Suchlaufzeiger muss sich zu diesem Zeitpunkt bereits im NCK befinden. Der Suchlauf kann jederzeit durch NC-RESET abgebrochen werden. Ist der Suchlauf abgeschlossen, wird automatisch wieder der normale Abarbeitungsmodus aktiviert. NC-Start wirkt dann ab dem gefundenen Suchziel. Der Bediener sorgt selbst für einen kollisionsfreien Anfahrweg.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.FINDBL	Suchlauf
Unit	INT	1 bis 10	Kanal
WVar1	WORD	x	Vorlaufmodus
x: beschreibt den Vorlaufmodus x = 1 ohne Berechnung x = 2 mit Berechnung x = 3 mit Hauptsatzbetrachtung			

### PI-Dienst: LOGIN

#### Funktion Kennwort anlegen:

Übergibt das parametrisierte Kennwort an den NCK. Das Kennwort besteht generell aus 8 Zeichen. Gegebenenfalls müssen in dem String des Kennworts Leerzeichen ergänzt werden.

#### Beispiel:

Kennwort : STRING[8] := 'SUNRISE';

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.LOGIN	Kennwort anlegen
Unit	INT	1	NCK
Addr1	STRING	8 Zeichen	Kennwort

### PI-Dienst: LOGOUT

#### Funktion Kennwort rücksetzen:

Das zuletzt an den NCK übergebene Kennwort wird zurückgesetzt.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.LOGOUT	Kennwort rücksetzen
Unit	INT	1	NCK

### PI-Dienst: NCRES

#### Funktion NC-RESET auslösen:

Löst einen NCK-RESET aus. Parameter Unit und WVar1 sind mit 0 zu versorgen.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.NCRES	NC-RESET auslösen
Unit	INT	0	0
WVar1	WORT	0	0

### PI-Dienst: SELECT

#### Funktion Abarbeitung für einen Kanal anwählen:

Ein auf dem NCK abgelegtes Programm wird für einen Kanal zur Abarbeitung angewählt. Dies ist nur möglich, wenn das File ausgeführt werden darf. Die Pfad-Namen und Programm-Namen sind, wie im Programmierhandbuch Arbeitsvorbereitung, Datei- und Programmverwaltung, Kapitel "Programmspeicher" beschrieben, einzugeben. Für die Schreibweise von Pfad- und Programmnamen siehe auch Beispiel des FB 4.

#### Mögliche Bausteintypen

Bausteintypen	
Werkstückverzeichnis	WPD
Hauptprogramm	MPF
Unterprogramm	SPF
Zyklen	CYC
Asynchrone Unterprogramme	ASP
Binärfiles	BIN

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.SELECT	Programm-Anwahl
Unit	INT	1 bis 10	Kanal
Addr1	STRING		Pfad-Name
Addr2	STRING		Programm-Name

**PI-Dienst: SETUdT**

**Funktion aktuellen User Daten aktiv setzen**

Die aktuellen User Daten wie Werkzeugkorrekturen, Basisframes und einstellbare Frames werden nur im Stopp-Zustand zum nächsten NC-Satz gesetzt.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.SETUdT	User Daten aktivieren
Unit	INT	1 bis 10	Kanal
WVar1	WORD	1 bis 5	User Data Type 1 = aktive Werkzeug-Korrektur 2 = aktiver Basis-Frame 3 = aktiver einstellbarer Frame 4 = aktiver globaler Basis-Frame 5 = aktiver globaler einstellbarer Frame
WVar2	WORD	0	Reserve
Wvar3	WORD	0	Reserve

**PI-Dienst: SETUFR**

**Funktion Anwender-Frames aktivieren:**

User Frames werden in den NCK geladen. Alle notwendigen Werte des Frames müssen vorher durch Variablen schreiben mit FB 3 an den NCK übermittelt werden.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.SETUFR	User-Frames aktivieren
Unit	INT	1 bis 10	Kanal

### 2.12.4.3 Werkzeugverwaltungsdienste

#### Überblick PI-Dienste

Verfügbare PI-Dienste der Funktion Werkzeugverwaltung.

PI-Dienst	Funktion	FM-NC	840D
<b>CRCEDN</b>	Lege neue Schneide an		*
<b>CREACE</b>	Werkzeugschneide anlegen	*	*
<b>CREATO</b>	Werkzeug erzeugen	*	*
<b>SETUFR</b>	Anwender-Frames aktivieren	*	*
<b>DELECE</b>	Löschen einer Werkzeugschneide		*
<b>DELETO</b>	Werkzeug löschen	*	*
<b>MMCSEM</b>	Semaphoren für verschiedene PI-Dienste		*
<b>TMCRT0</b>	Werkzeug anlegen		*
<b>TMFDPL</b>	Leerplatzsuche zum Beladen		*
<b>TMFPBP</b>	Leerplatzsuche		*
<b>TMMVTL</b>	Magazinplatz zum Beladen bereitstellen, Werkzeug entladen		*
<b>TMPOSM</b>	Magazinplatz oder Werkzeug positionieren		*
<b>TMPCIT</b>	Inkrementwert für Stückzahlzähler setzen		*
<b>TMRASS</b>	Rücksetzen des Aktiv-Status		*
<b>TRESMO</b>	Rücksetzen der Überwachungswerte		*
<b>TSEARC</b>	Komplexes Suchen über Suchmasken		*
x: PI-Dienst ist verfügbar			

**PI-Dienst: CRCEDN**

**Funktion Lege neue Schneide an:**

Anlegen einer Werkzeugschneide unter Vorgabe der Schneidenummer.

Wird in dem PI-Dienst unter dem Parameter T-Nummer die T-Nummer eines existierenden Werkzeugs angegeben, so wird die Schneide zu diesem Werkzeug angelegt (in diesem Fall hat der Parameter D-Nummer - die Nummer der anzulegenden Schneide - einen Wertebereich von 00001 - 00009. Ist eine positive T-Nummer als Parameter angegeben und das Werkzeug zu der angegebenen T-Nummer existiert nicht, so schlägt der PI-Dienst fehl. Wird für die T-Nummer der Wert 00000 angegeben (Modell der absoluten D-Nummern), so kann sich der Wertebereich der D-Nummer von 00001 - 31999 erstrecken. Die neue Schneide wird mit der vorgegebenen D-Nummer erzeugt. Existiert die angegebene Schneide schon, so schlägt der PI-Dienst in beiden Fällen fehl.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CRCEDN	Lege neue Schneide an
Unit	INT	1, 2	TOA
WVar1	INT		T-Nummer des Werkzeuges, zu dem die Werkzeugschneide angelegt werden soll. Ein Wert 00000 besagt, dass kein Bezug zu einem Werkzeug existieren soll (absolute D-Nummer)
WVar2	INT	1 - 9 bzw. 01 - 31999	Schneiden-Nummer der Werkzeugschneide

**PI-Dienst: CREACE**

**Funktion Werkzeugschneide anlegen:**

Erzeugen der Schneide mit der nächst höheren/freien D-Nummer zu dem Werkzeug mit der übergebenen T-Nummer in TO, TS (sofern vorhanden), im TUE-Baustein - sofern vorhanden - wird die Schneide für die OEM-Schneidendaten mit angelegt.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CREACE	Werkzeugschneide anlegen
Unit	INT	1, 2	TOA
WVar1	INT		T-Nummer

## PI-Dienst: CREATO

### Funktion Werkzeug anlegen:

Anlegen eines Werkzeuges unter Vorgabe einer T-Nummer. Das Werkzeug wird im Bereich TV (Werkzeug-Directory) als vorhanden eingetragen, im TO-Baustein wird die erste Schneide "D1" (mit Null-Inhalt) für die Korrekturen angelegt, im TUE-Baustein - sofern vorhanden - wird die erste Schneide "D1" (mit Null-Inhalt) für die OEM-Schneidendaten angelegt. Falls ein TU-Baustein vorhanden ist, ist dort der Datensatz für das Werkzeug bereitgestellt.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.CREATO	Werkzeug anlegen
Unit	INT	1, 2	TOA
WVar1	INT		T-Nummer

## PI-Dienst: DELECE

### Funktion Löschen einer Werkzeugschneide:

Wird in dem PI-Dienst unter dem Parameter T-Nummer die T-Nummer eines existierenden Werkzeugs angegeben, so wird die Schneide zu diesem Werkzeug gelöscht (in diesem Fall hat der Parameter D-Nummer - die Nummer der anzulegenden Schneide - einen Wertebereich von 00001 - 00009. Ist eine positive T-Nummer als Parameter angegeben und das Werkzeug zu der angegebenen T-Nummer existiert nicht, so schlägt der PI-Dienst fehl. Wird für die T-Nummer der Wert 00000 angegeben (Modell der absoluten D-Nummern), so kann sich der Wertebereich der D-Nummer von 00001 - 31999 erstrecken. Existiert die angegebene Schneide nicht, so schlägt der PI-Dienst in beiden Fällen fehl.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.DELECE	Lösche Schneide
Unit	INT	1, 2	TOA
WVar1	INT		T-Nummer des Werkzeuges, zu dem die Werkzeugschneide angelegt werden soll. Ein Wert 00000 besagt, dass kein Bezug zu einem Werkzeug existieren soll (absolute D-Nummer)
WVar2	INT	1 - 9 bzw. 01 - 31999	Schneiden-Nummer der Werkzeugschneide, welche gelöscht werden soll

**PI-Dienst: DELETO**

**Funktion Werkzeug löschen:**

Löscht das Werkzeug zu der übergebenen T-Nummer mit allen Schneiden (in TO, ggf. TU, TUE und TG (Typ 400), TD und TS Bausteinen).

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.DELETO	Werkzeug löschen
Unit	INT	1, 2	TOA
WVar1	INT		T-Nummer

**PI-Dienst: MMCSEM**

Semaphoren für verschiedene PI-Dienste

**Anwendbar durch MMC und PLC**

Kanalspezifisch werden jeweils 10 Semaphore zum Schützen von kritischen Funktionen für MMC/PLC angeboten. Durch Setzen der Semaphore zu der entsprechenden Funktionsnummer können sich mehrere MMC/PLC-Einheiten an der Semaphore synchronisieren, wenn eine Funktion einen kritischen Abschnitt bezüglich der von NCK abzuholenden Daten besitzt. MMC/PLC verwaltet die Semaphore. Ein Semaphore-Wert von 1 spezifiziert eine Teste & Setze-Operation für die Semaphore zu der angegebenen Funktionsnummer. Der Rückgabewert des PI-Dienstes gibt das Ergebnis dieser Operation an:

- Rückgabewert Done := TRUE  
Semaphore konnte gesetzt werden, kritische Funktion kann aufgerufen werden.
- Rückgabewert Error := TRUE mit State = 3: Semaphore war bereits gesetzt, kritische Funktion kann nicht aufgerufen werden. Dies muss zu einem späteren Zeitpunkt wiederholt werden.

---

**Hinweis**

Nach Beenden der Operation (Lesen der Daten dieses PI-Dienstes) **muss** die **Semaphore** unbedingt wieder **freigegeben** werden.

---



**Parameter:**

WVar1 = FunctionNumber

Dies ist eine Funktionsnummer, die einen PI-Dienst repräsentiert:

```

1:      TMCRT0 (Werkzeug anlegen)
2:      TMFDPL (Leerplatzsuche zum Beladen)
3:      TMMVTL (Magazinplatz zum Beladen bereitstellen, Werkzeug entladen)
4:      TMFPBP (Suche Platz)
5:      TMGETT (Suche Werkzeugnummer)
6:      TSEARC (Werkzeug suchen)
7 ...   reserviert
10:

```

WVar2=SemaphorValue

```

0:      Semaphore zurücksetzen
1:      Semaphore testen und setzen

```

<b>Parametrierung</b>			
<b>Signal</b>	<b>Typ</b>	<b>Wertebereich</b>	<b>Bedeutung</b>
PIService	ANY	PI.MMCSEM	Semaphore setzen
Unit	INT	1, 2 bis 10	Kanal
WVar1	INT	1 bis 10	FunctionNumber
WVar2	WORD	0, 1	SemaphoreValue

**PI-Dienst: TMCRTO**

**Funktion Werkzeug anlegen:**

Anlegen eines Werkzeuges unter Vorgabe

- eines Bezeichners, einer Duplo-Nummer z.B. mit  
 \$TC\_TP1[y] = Duplonummer;  
 \$TC\_TP2[y] = "Werkzeugbezeichner"
- und optional einer T-Nummer z.B. mit y = T-Nummer

Das Werkzeug wird im Bereich TV (Werkzeug-Directory) als vorhanden eingetragen, im TO-Baustein wird die erste Schneide "D1" (mit Null-Inhalt) für die Korrekturen angelegt, im TS-Baustein wird die erste Schneide "D1" (mit Null-Inhalt) für die Überwachungsdaten angelegt, im TUE-Baustein - sofern vorhanden - wird die Schneide "D1" für die OEM-Schneidendaten mit angelegt, im TD-Baustein stehen Bezeichner, Duplonummer und Anzahl der Schneiden (=1) zu der optional vorgegebenen oder von NCK vergebenen T-Nummer.

Falls ein TU-Baustein vorhanden ist, ist dort der Datensatz für das Werkzeug bereitgestellt. Nach Ausführung des PIs steht im TV-Baustein unter **TnumWZV** die T-Nummer zu dem angelegten Werkzeug.

**Hinweis**

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechendem Parameter "WVar1" für diesen PI-Dienst aufzurufen. Näheres siehe PI-Dienst MMCSEM.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMCRTO	Werkzeug anlegen
Unit	INT	1, 2	TOA
WVar1	INT		T-Nummer
WVar2	INT		Duplo-Nummer
Addr1	STRING	max. 32 Zeichen	Werkzeugbezeichner
T-Nummer > 0 bedeutet, eine T-Nummer vorgeben T-Nummer = -1 bedeutet, NCK soll T-Nummer vergeben Beispiel zeigt T-Nummer = -1 ⇒ T-Nummer von NCK vergeben			

**PI-Dienst: TMFDPL**

**Funktion Leerplatzsuche zum Beladen:**  
(in Abhängigkeit der Parameterbelegung)

**Platz-Nummer\_Wohin = -1, Magazin-Nummer\_Wohin = -1:**

Sucht unter allen Magazinen zum angegebenen Bereich (= Kanal) einen Leerplatz für das mit T-Nummer spezifizierte Werkzeug. Die gefundene Magazin- und Platznummer steht nach Ausführung des PIs im Konfigurationsblock des Kanals (Komponente **magCMCcmdPar1** (Magazinnummer) und **magCMCcmdPar2** (Platznummer)). Platz-Nummer\_Bez und Magazin-Nummer\_Bez können als Suchkriterium besetzt sein oder nicht (= -1). Der PI wird in Abhängigkeit vom Suchergebnis positiv oder negativ quittiert.

**Platz-Nummer\_Wohin = -1, Magazin-Nummer\_Wohin = Magazin-Nummer:**

Es wird in dem angegebenen Magazin ein Leerplatz für das mit T-Nummer spezifizierte Werkzeug gesucht. Platz-Nummer\_Bez und Magazin- Nummer\_Bez können als Suchkriterium besetzt sein oder nicht (= -1). Der PI wird in Abhängigkeit vom Suchergebnis positiv oder negativ quittiert.

**Platz-Nummer\_Wohin = Platz-Nummer, Magazin-Nummer\_Wohin = Magazin-Nummer:**

Der angegebene Platz wird geprüft, ob er zum Beladen mit dem angegebenen Werkzeug frei ist. Platz-Nummer\_Bez und Magazin-Nummer\_Bez können als Suchkriterium besetzt sein oder nicht (= -1). Der PI wird in Abhängigkeit vom Prüfergebnis positiv oder negativ quittiert.

Die Kommando-Parameter 1 und 2 stehen bei der Quelle.

Laden: Wenn Quelle ein internes Belademagazin ist, dann stehen die Kommando-Parameter beim Ziel (einem realen Magazin).

Entladen: Quelle ist immer reales Magazin.

---

**Hinweis**

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechendem Parameter "WVar1" für diesen PI-Dienst aufzurufen. Näheres siehe PI-Dienst MMCSEM.

---

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMFDPL	Leerplatz zum Beladen
Unit	INT	1, 2	TOA
WVar1	INT		T-Nummer
WVar2	INT		Platz-Nummer _Wohin
WVar3	INT		Magazin-Nummer _Wohin
WVar4	INT		Platz-Nummer _Bez
WVar5	INT		Magazin-Nummer _Bez

## PI-Dienst: TMFPBP

### Funktion Leerplatzsuche

(in Abhängigkeit der Parameterbelegung):

Siehe hierzu die Beschreibung von FB 7: PI\_SERV2 Allgemeine PI-Dienste):

## PI-Dienst: TMMVTL

### Funktion Magazinplatz zum Beladen bereitstellen, Werkzeug entladen:

Der PI-Dienst wird sowohl zum Beladen als auch zum Entladen verwendet. Welche Operation mit dem PI angestoßen wird, ist in der Zuordnung der realen Plätze zu den "Von"-Parametern und den "Zu"-Parametern abhängig: Beladen ⇒ "Von" = Beladestelle/-station, Entladen ⇒ "Zu" = Beladestelle/-station.

Der PI-Dienst TMMVTL wird für alle Bewegungen verwendet:

1. Be- und Entladen (Beladestelle ↔ Magazin)
2. Be- und Entladen (Beladestelle ↔ Zwischenspeicher, z. B. Spindel)
3. Umsetzen innerhalb eines Magazins
4. Umsetzen zwischen verschiedenen Magazinen
5. Umsetzen zwischen Magazin und Zwischenspeicher
6. Umsetzen innerhalb des Zwischenspeichers

Für die Überwachung von Fall 1, 3, 4, 5 werden folgende Variablen aus Baustein TM verwendet:

magCmd (BereichsNr = TO-Einheit, Zeile = Magazinnummer)

magCmdState <- "Quittung"

Für die Überwachung von Fall 2), 6) werden folgende Variablen aus Baustein TMC verwendet:

magCBCmd (BereichsNr = TO-Einheit)

magCBCmdState <- "Quittung"

### Funktion Beladen

Bereitet das angegebene reale Magazin zu dem spezifizierten Kanal zum Beladen vor, i. e. verfährt das Magazin auf den gewünschten Platz zum Beladen an der angegebenen Beladestation/-platz (Platz-Nummer\_Von, Magazin-Nummer\_Von) und wechselt des Werkzeug ein.

Bei Platznummer\_Zu = -1 wird in dem angegebenen Magazin zuerst ein Leerplatz für das mit T-Nummer spezifizierte Werkzeug gesucht und dann das Magazin verfahren. Die gefundene Platznummer steht nach Ausführung des PI's im TM-Bereich in der Komponente **magCMCmdPar2** zu dem **realen** Magazin des Kanals.

Bei Platznummer\_Zu = -2 und gültiger Magazinnummer wird in die aktuell anstehende Magazinposition des angegebenen Magazins beladen. Die Platznummer des zu beladenden Platzes steht nach Ausführung des PI's im TM-Bereich in der Komponente **magCMCmdPar2** zu dem realen Magazin des Kanals.

### **Funktion Entladen**

Das mit T-Nummer spezifizierte Werkzeug wird an der angegebenen Beladestelle/-station (Platz-Nummer\_Zu, Magazin-Nummer\_Zu) entladen, d.h. das Magazin wird zum Entladen verfahren, das Werkzeug ausgewechselt. Im TP-Baustein wird der Magazinplatz zu dem Werkzeug als frei gekennzeichnet. Das Werkzeug kann wahlweise über T-Nummer oder über Platz- und Magazinnummer angegeben werden. Eine nicht benutzte Spezifikation trägt den Wert -1.

---

### **Hinweis**

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechendem Parameter "WVar1" für diesen PI-Dienst aufzurufen. Näheres siehe PI-Dienst MMCSEM.

---

<b>Parametrierung</b>			
<b>Signal</b>	<b>Typ</b>	<b>Wertebereich</b>	<b>Bedeutung</b>
PIService	ANY	PI.TMMVTL	Magazinplatz zum Beladen bereitstellen, WZ entladen
Unit	INT	1, 2	TOA
WVar1	INT		T-Nummer
WVar2	INT		Platz-Nummer_Von
WVar3	INT		Magazin-Nummer_Von
WVar4	INT		Platz-Nummer_Zu
WVar5	INT		Magazin-Nummer_Zu

**PI-Dienst: TMPOSM**

**Funktion Magazinplatz oder Werkzeug positionieren:**  
 (in Abhängigkeit der Parameterbelegung)

Über den PI-Dienst wird ein Magazinplatz, der entweder direkt angegeben ist oder über ein darauf befindliches Werkzeug qualifiziert wurde, an eine vorgegebene Position (z.B. vor einen Beladeplatz) gefahren.

Der PI-Dienst lässt einen auf unterschiedliche Arten qualifizierbaren Magazinplatz vor einen angegebenen Beladeplatz fahren. Der Beladeplatz wird in den PI-Parametern "Platz-Nummer\_Von" und "Magazin-Nummer\_Von" angegeben (muss!).

Die Qualifikation des zu fahrenden Magazinplatzes kann erfolgen durch:

- T-Nummer des Werkzeuges

Der Platz, auf dem sich das Werkzeug befindet, fährt; die Parameter "WZ-Bezeichner", "Duplonummer", "Platz-Nummer\_Von" und "Magazin-Nummer\_Von" sind irrelevant (d.h. Werte "", "-0001", "-0001", "-0001").

oder

- WZ-Bezeichner und Duplo-Nummer

Der Platz, auf dem sich das Werkzeug befindet, fährt; die Parameter "T-Nummer", "Platz-Nummer\_Von" und "Magazin-Nummer\_Von" sind irrelevant (d.h. jeweils Wert "-0001").

oder

- Direkte Angabe des Platzes in den Parametern "Platz-Nummer\_Von" und "Magazin-Nummer\_Von".

Die das Werkzeug qualifizierenden Parameter "T-Nummer", "WZ-Bezeichner" und "Duplonummer" sind irrelevant (d.h. Werte "-0001", "", "-0001").

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMPOSM	Magazinplatz oder Werkzeug positionieren
Unit	INT	1, 2	TOA
Addr1	STRING	max 32. Zeichen	Werkzeugbezeichner
WVar1	INT		T-Nummer
WVar2	INT		Duplonummer
WVar3	INT		Platz-Nummer_Von
WVar4	INT		Magazin-Nummer_Von
WVar5	INT		Platz-Nummer_Bezug
WVar6	INT		Magazin-Nummer_Bezug

### PI-Dienst: TMPCIT

#### Funktion Inkrementwert für Stückzahlzähler setzen:

Inkrementieren des Stückzahlzählers des Spindelwerkzeugs.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMPCIT	Inkrementwert für Stückzahlzähler setzen
Unit	INT	1 bis 10	TOA
WVar1	WORD	0 ... max.	Spindelnummer; entspricht dem Artindex bei den Platzdaten mit Platzart Spindel des Zwischenspeichermagazins im Kanal.000 = Hauptspindel
WVar2	WORD	0 ... max.	Inkrementwert; gibt an, nach wie vielen Umdrehungen der Spindel der Stückzahlzähler inkrementiert wird.

### PI-Dienst: TMRASS

#### Funktion Rücksetzen des Aktiv-Status:

Rücksetzen des Aktiv-Status bei verschlissenen Werkzeugen.

Mit diesem PI-Dienst werden alle Werkzeuge gesucht, die den Werkzeug-Zustand aktiv und gesperrt haben. Diesen Werkzeugen wird dann der Aktiv-Zustand wieder weggenommen. Mögliche sinnvolle Zeitpunkte für diesen PI-Dienst sind die negative Flanke des VDI-Signals "Werkzeug-Sperre unwirksam", ein Programmende, ein Kanal-RESET. Dieser PI-Dienst ist hauptsächlich für die PLC sinnvoll, weil hier auch bekannt ist, wann das gesperrte Werkzeug endgültig nicht mehr verwendet werden soll.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI. TMRASS	Rücksetzen des Aktiv-Status
Unit	INT	1 bis 10	TO-Bereich

**PI-Dienst: TRESMO**

**Funktion Zurücksetzen der Überwachungswerte:**

Mit diesem PI-Dienst können die Überwachungswerte der bezeichneten Schneiden der bezeichneten Werkzeuge auf die Soll- (Ausgangs-) werte zurückgesetzt werden. Dies erfolgt nur für Werkzeuge, die eine Überwachung aktiv haben.

Vergleiche NC-Sprachbefehl `RESETMON`.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI. TRESMO	Zurücksetzen der Überwachungswerte
Unit	INT	1 bis 10	TO-Bereich
WVar1	WORD	-max ..max	Toolnummer 0: alle Werkzeuge behandelt >0: nur dieses Werkzeug wird behandelt <0: alle Schwesterwerkzeuge zur angegebenen T-Nr. werden behandelt
WVar2	WORD	0 ... max.	D-Nummer >0: Überwachung der angegebenen Schneide der angegebenen Werkzeuge wird zurückgesetzt. 0: Überwachung aller Schneiden der angegebenen Werkzeuge wird zurückgesetzt.
Wvar3	WORD	0 ...15	Überwachungsarten Art der Überwachung, die zurückgesetzt werden soll. Dieser Parameter ist binär codiert. 1: Standzeit-Überwachung wird zurückgesetzt. 2: Stückzahl-Überwachung wird zurückgesetzt. 4: Verschleiß-Überwachung wird zurückgesetzt. 8: Summenkorrektur-Überwachung wird zurückgesetzt. Es können durch Addition der obigen Werte auch Kombination von Überwachungen zurückgesetzt werden. 0: Es werden alle aktiven Überwachungen des Werkzeuges (\$TC_TP9) zurückgesetzt.



## PI-Dienst: TSEARCH

### **Funktion Komplexes Suchen über Suchmasken:**

(in Abhängigkeit der Parameterbelegung)

Mit dem PI-Dienst können innerhalb eines Suchbereichs (in einem oder über mehrere Magazine, beginnend ab einem bestimmten Platz bis zu einem bestimmten Platz) Werkzeuge mit vorgegebenen Eigenschaften gesucht werden. Die vorgebbaren Eigenschaften beziehen sich nur auf Daten der Werkzeuge und deren Schneiden.

Der PI-Dienst ist nur bei aktivierter Werkzeugverwaltung verfügbar.

Für die Ausführung des Dienstes können eine Suchrichtung vorgegeben werden sowie die Ergebnismenge (ein Werkzeug (entspricht nächstes Werkzeug mit dieser Eigenschaft) oder alle Werkzeuge mit der vorgegebenen Eigenschaft) spezifiziert werden.

Als Ergebnis des Dienstes erhält der Aufrufer eine Liste mit den internen T-Nummern der gefundenen Werkzeuge zurück.

Die Suchkriterien können nur als UND-Verknüpfung vorgegeben werden. Will eine Applikation eine ODER-Verknüpfung als Suchkriterium vorgeben, so muss sie dies über mehrere nacheinander ausgeführte Anfragen mit UND-Verknüpfungen realisieren und die jeweiligen Ergebnisse der einzelnen Anfragen entsprechend verknüpfen/auswerten.

Zur Parametrierung des PI-Dienstes werden zunächst über Variablendienst im Baustein TF die Eigenschaften der gesuchten Werkzeuge vorgegeben. Dazu werden im Baustein TF in den Operanden-Masken (parMaskT..) die relevanten

- Vergleichskriterien (welche Werkzeugdaten sollen verglichen werden?) markiert,
- Vergleichsoperatoren-Daten (parDataT..) mit den entsprechend auszuführenden Vergleichsarten (==, <, >, <=, >=, &&) belegt
- und in den Operandendaten werden die Vergleichswerte eingetragen.

Dann wird der PI-Dienst angestoßen und nach dessen erfolgreicher Rückkehr werden über den Variablendienst aus dem Baustein TF unter der Variablen resultNrOfTools die Anzahl der Treffer und unter der Variablen resultToolNr die Ergebnisliste (interne T-Nummern der Werkzeuge, die bei der Suche gefunden wurden - resultNrOfTools Stück) ausgelesen.

Der PI-Dienst ist von seiner Vorbereitung bis zum erfolgten Abholen des Ergebnisses mit einer Semaphore zu kapseln. Nur so kann der exklusive Zugriff und die exklusive Verwendung des Bausteins TF zusammen mit dem PI-Dienst TSEARCH sichergestellt werden. Die hierfür vorgesehene Funktionsnummer bei dem Semaphore-Mechanismus (PI-Dienst MMCSEM) ist die Funktionsnummer für TSEARCH.

Ist der Dienst fehlerhaft parametrierung, so schlägt er fehl. Ansonsten liefert er immer ein Ergebnis, auch wenn kein Werkzeug gefunden wird (resultNrOfTools = 0).

Der Suchbereich kann über die Belegung der Parameter "MagNrFrom", "PlaceNrFrom", "MagNrTo", "PlaceNrTo" wie folgt vorgegeben werden:

MagNr From	PlaceNr From	MagNr To	PlaceNr To	Suchbereich
WVar1	WVar2	WVar3	WVar4	
#M1	#P1	#M2	#P2	es werden die Plätze beginnend von Magazin #M1, Platz #P1 bis Magazin #M2, Platz #P2 durchsucht
#M1	-1	#M1	-1	es werden alle Plätze von Magazin #M1 - und nur diese - durchsucht
#M1	-1	-1	-1	es werden alle Plätze beginnend ab Magazin #M1 durchsucht
#M1	#P1	-1	-1	es werden alle Plätze beginnend ab Magazin #M1 und darin Platz #P1 durchsucht
#M1	#P1	#M1	-1	es werden die Plätze in Magazin #M1 beginnend ab Magazin #M1 und darin Platz #P1 durchsucht
#M1	#P1	#M2	-1	es werden die Plätze beginnend ab Magazin #M1 und darin Platz #P1 bis einschließlich Magazin #M2 durchsucht
#M1	-1	#M2	#P2	es werden die Plätze beginnend ab Magazin #M1 bis einschließlich Magazin #M2 und darin Platz #P2 durchsucht
#M1	-1	#M2	-1	es werden die Plätze beginnend ab Magazin #M1 bis einschließlich Magazin #M2 durchsucht
-1	-1	-1	-1	es werden alle Magazinplätze durchsucht

Für eine symmetrische Suche (vgl. Parameter "SearchDirection")

- darf sich der Suchbereich nur über ein einziges Magazin erstrecken (Fälle 2 und 5 aus obiger Tabelle). Ist ein anderer Suchbereich angegeben, schlägt der Dienst fehl.
- muss in den Parametern "MagNrRef" und "PlaceNrRef" ein Referenzplatz angegeben werden, bezüglich dem die symmetrische Suche erfolgt.

Der Referenzplatz ist ein Zwischenspeicher-Platz (ein Platz aus dem Magazin Zwischenspeicher, d.h. Wechselstelle, Greifer ...) oder ein Beladeplatz/-stelle (ein Platz aus dem internen Belademagazin). Die symmetrische Suche erfolgt bezüglich dem Magazinplatz vor dem angegebenen Referenzplatz. Für den angegebenen Referenzplatz muss eine Mehrfachzuordnung zu dem zu durchsuchenden Magazin im Baustein TPM konfiguriert sein. Ist dies nicht der Fall, so schlägt der Dienst fehl. Liegt der Magazinplatz vor dem Referenzplatz außerhalb des Suchbereichs, so verhält sich der Dienst so, als ob er keinen passenden Platz gefunden hat.

---

#### Hinweis

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechenden Parameter "WVar1" für diesen PI-Dienst aufzurufen. Näheres siehe PI-Dienst MMCSEM.

---

**Parametrierung:**

<b>Parametrierung</b>			
<b>Signal</b>	<b>Typ</b>	<b>Wertebereich</b>	<b>Bedeutung</b>
PIService	ANY	PI.TSEARC	Komplexes Suchen über Suchmasken
Unit	INT	1, 2	TOA
WVar1	INT		MagNrFrom Magazinnummer des Magazins, ab dem der Suchbereich beginnen soll.
WVar2	INT		PlaceNrFrom Platznummer des Platzes in dem Magazin MagNrFrom, ab dem der Suchbereich beginnen soll.
WVar3	INT		MagNrTo Magazinnummer des Magazins, bei dem der Suchbereich enden soll.
WVar4	INT		PlaceNrTo Platznummer des Platzes in dem Magazin MagNrTo, bei dem der Suchbereich enden soll.
WVar5	INT		MagNrRef Magazinnummer des (internen) Magazins, bezüglich dem symmetrisch gesucht werden soll. (dieser Parameter ist nur relevant bei einer Suchrichtung "symmetrisch")
WVar6	INT		PlaceNrRef Platznummer des Platzes in dem Magazin MagNrRef, bezüglich dem symmetrisch gesucht werden soll. Dieser Parameter ist nur relevant bei einer Suchrichtung "symmetrisch".
WVar7	INT	1, 2, 3	SearchDirection spezifiziert die gewünschte Suchrichtung. 1: vorwärts vom ersten Platz des Suchbereichs 2: rückwaerts vom letzten Platz des Suchbereichs 3: symmetrisch zu realen Magazinplatz, der vor dem mit MagazinNummer_Bezug und PlatzNummer_Bezug angegebenen Platz steht
WVar8	INT	0, 1, 2, 3	KindofSearch 0: schneidenspezifisch alle Werkzeuge mit dieser Eigenschaft suchen 1: das erste gefundene Werkzeug mit dieser Eigenschaft suchen (schneidenspezifisch) 2: alle Werkzeuge über alle Schneiden mit dieser Eigenschaft suchen 3: das erste gefundene Werkzeug mit dieser Eigenschaft suchen (über alle Schneiden)

**Aufrufbeispiel**

**Programmanwahl im Kanal 1 (Hauptprogramm und Werkstückprogramm)**

Eintrag PI für DB 16 und STR für DB 124 mit dem S7-SYMBOL-Editor:

Parametrierung		
Symbol	Operand	Datentyp
PI	DB 16	DB 16
STR	DB 124	DB 124

```

DATA_BLOCK DB 126           //freier Anwender-DB, als Instanz für FB 4
FB 4
BEGIN
END_DATA_BLOCK

DATA_BLOCK DB 124
  struct
    PName:      string[32]:= '_N_TEST_MPF';
    Path:       string[32]:= '/_N_MPF_DIR/'; //Hauptprogramm
    PName_WST:  string[32]:= '_N_ABC_MPF';
    Path_WST:   string[32]:=           //Werkstückprogramm
               '/_N_WKS_DIR/_N_ZYL_WPD';
  end_struct
BEGIN
END_DATA_BLOCK

Function FC "PICall" : VOID
  U E 7.7;           //freie Taste Maschinensteuertafel
  S M 0.0;           //Req aktivieren
  U M 1.1;           //Fertigmeldung Done
  R M 0.0;           //Auftrag beenden
  U E 7.6;           //Fehlerquittierung von Hand
  U M 1.0;           //Fehler steht an
  R M 0.0;           //Auftrag beenden

  CALL FB 4, DB 126(
    Req:=           M0.0,
    PIService:=     PI.SELECT,
    Unit:=          1, //CHAN 1
    Addr1:=         STR.Path,
    Addr2:=         STR.PName, //Hauptprogrammmanwahl
                   //Addr1:=STR.Path_WST,
                   //Addr2:=STR.PName_WST, //Werkstückprogrammmanwahl
    FMNCNo:=        1, //((nur bei FM-NC)
    Error:=         M1.0,
    Done:=          M1.1,
    State:=         MW2);
  
```

## 2.12.5 FB 5: GETGUD GUD-Variable lesen

### Funktionsbeschreibung

Mit dem FB GETGUD kann das PLC-Anwenderprogramm eine GUD Variable (GUD = Global User Data, Globales anwenderdefiniertes Datum) im NCK- oder Kanal-Bereich lesen. Der FB ist Multi-Instanzfähig. Bei den Namen der GUD Variablen sind nur Großbuchstaben zulässig. Zu jedem FB 5-Aufruf muss ein separater Instanz-DB aus dem Anwender-Bereich zugeordnet werden.

Durch **Aufruf des FB 5** mit positivem Flankenwechsel am Steuereingang "Req" wird ein Auftrag gestartet. Zu diesem Auftrag gehört der Name der zu lesenden GUD Variable in dem Parameter "Addr" mit dem Datentyp "STRING". Der Zeiger auf den Namen der GUD Variablen wird dem Parameter "Addr" symbolisch mit <DatenbausteinName>.<VariablenName> zugewiesen. In den weiteren Parametern "Area" "Unit" "Index1" und "Index2" werden Zusatz-Informationen zu dieser Variablen angegeben, siehe Tabelle der Bausteinparameter.

Mit Aktivierung des Parameters "CnvtToken" kann optional ein Variablenzeiger (Token) für diese GUD Variable erstellt werden. Dieser Variablenzeiger wird bei Systemvariablen der NC über den VAR-Selektor erstellt. Für die GUD Variablen besteht nur diese Variante der Zeigergenerierung. Nachdem dieser GUD Variablen Zeiger gebildet wurde, dann kann auch über den FB 2 und FB 3 (GET, PUT) mit Referenzierung auf diesen Variablenzeiger gelesen bzw. auch geschrieben werden. Dies ist der einzige Weg GUD Variablen zu schreiben. Hierbei muss bei der Parametrierung des FB 2 bzw. FB 3 nur der Parameter Addr1 .. Addr8 auf diesen GUD Variablenzeiger parametrieren. Eine Ausnahme bilden GUD Variablenfelder. Bei diesen Feldern ist zusätzlich Line1 .. Line8 mit dem Feldindex dieser Variable zu parametrieren.

Der erfolgreiche Abschluss des Lesevorgangs wird am Zustandsparameter "Done" mit logisch 1 angezeigt.

Der Lesevorgang erstreckt sich über mehrere (In der Regel 1..2) PLC-Zyklen. Der Baustein kann nur im zyklischen Betrieb aufgerufen werden.

Eventuell aufgetretene Fehler werden über Error und State angezeigt.

---

### Hinweis

FB 5 kann GUD-Variablen nur dann lesen, wenn der Grundprogramm-Parameter "NCKomm" auf 1 gesetzt wurde (in OB 100: FB 1, DB 7).

---

## Deklaration der Funktion

```
FUNCTION_BLOCK FB 5                                //Name-Server
    KNOW_HOW_PROTECT
    VERSION: 3.0
VAR_INPUT
    Req :          BOOL;
    Addr :         ANY;                                //NamenString der Variablen
    Area :         BYTE;                              //Bereich: NCK = 0, Kanal = 2
    Unit :         BYTE;
    Index1 :       INT;                               //Feldindex 1
    Index2 :       INT;                               //Feldindex 2
    CnvtToken :    BOOL;                              //Umsetzung in 10 Byte Token
VarToken :        ANY;                              //Struct mit 10 Byte für den Variablen
                                                         Token
FMNCNo :          INT;                              //(nur bei FM-NC)

END_VAR

VAR_OUTPUT
    Error :        BOOL;
    Done :         BOOL;
    State :        WORD;
END_VAR

VAR_IN_OUT
    RD :           ANY;
END_VAR

BEGIN
END_FUNCTION_BLOCK
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion GETGUD.

Signal	Art	Typ	Wertebereich	Bemerkung
Req	E	BOOL		Auftragsstart mit positiver Flanke
Addr	E	ANY	[DBName].[VarName]	GUD Variablenname in einer Variable vom Datentyp String
Area	E	BYTE		Bereichsadresse: 0: NCK-Variable 2: Kanal-Variable
Unit	E	BYTE		Bereich NCK: Unit:= 1 Bereich Kanal: Kanalnr.
Index1	E	INT		Feldindex 1 der Variable Der Wert der Variable ist 0, wenn der Feldindex nicht benutzt wird.
Index2	E	INT		Feldindex 2 der Variable Der Wert der Variable ist 0, wenn der Feldindex nicht benutzt wird.
CnvtToken	E	BOOL		Aktivierung der Generierung eines Variablen-Token
VarToken	E	ANY		Adresse auf einen 10 Byte Token (siehe Beispiel)
FMNCNo (nur bei FM-NC)	E	INT	0, 1, 2	0, 1=1 NCU, 2=2 NCU's
Error	A	BOOL		Auftrag wurde negativ quittiert, bzw. konnte nicht ausgeführt werden
Done	A	BOOL		Auftrag wurde erfolgreich ausgeführt.
State	A	WORD		siehe Fehlerkennungen
RD	E/A	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	zu schreibende Daten

## Fehlerkennungen

Konnte ein Auftrag nicht ausgeführt werden, wird dies am Zustandsparameter Error mit "logisch 1" angezeigt. Die Fehlerursache ist am Bausteinausgang State kodiert:

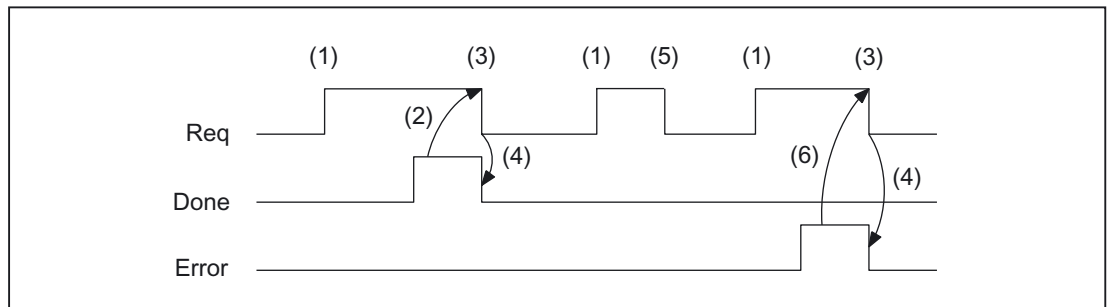
State		Bedeutung	Hinweis
WORT-H	WORT-L		
0	1	Zugriffsfehler	
0	2	Fehler im Auftrag	falsche Zusammenstellung von Var. in einem Auftrag
0	3	negative Quittung, Auftrag nicht ausführbar	interner Fehler, evtl. Abhilfe: NC-RESET
0	4	Datenbereiche oder Datentypen stimmen nicht überein	zu lesendes Datum in RD überprüfen;
0	6	FIFO voll	Auftrag muss wiederholt werden, da die Warteschlange voll ist
0	7	Option nicht gesetzt	GP-Parameter "NCKomm" ist nicht gesetzt
0	8	falscher Zielbereich (SD)	RD dürfen keine Lokaldaten sein
0	9	Übertragung belegt	Auftrag muss wiederholt werden
0	10	Fehler bei Adressierung	Unit enthält Wert 0
0	11	Variablenadresse ungültig	Addr (bzw. Variablenname), Area, Unit überprüfen

## Projektierungsschritte

Für das Lesen von einer GUD Variable ist der Name der GUD Variable in einer String-Variablen zu hinterlegen. Der Datenbaustein mit dieser String-Variablen ist in der Symbolliste zu definieren, so dass eine symbolische Zuweisung des Parameters "Addr" für den FB GETGUD erfolgen kann. Optional ist eine Strukturvariable in einem beliebigen Datenbereich der PLC für den Empfang des Variablenzeigers zu definieren, siehe Vorgabe im folgenden Beispiel.



## Impulsdiagramm



- (1) Funktionsanstoß
- (2) Positive Quittung: Variablen wurden geschrieben
- (3) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FB
- (5) Wird Funktionsanstoß vor Erhalt der Quittung zurückgesetzt, werden die Ausgangssignale nicht aktualisiert, ohne Einfluss auf Ablauf der angestoßenen Funktion
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter State

## Aufrufbeispiel

Lesen einer GUD-Variable mit dem Namen "GUDVAR1" als Integer-Variable (siehe auch Tabelle im FB 2: Zuordnung NC-Datentyp in SIMATIC-Datentyp).

Aufruf und Parametrierung des FB 5 mit dem Instanz-DB 111:

```
DATA_BLOCK DB GUDVAR //Zuordnung in Symbolliste vornehmen
STRUCT
  GUDVar1 : STRING[32] := 'GUDVAR1'; //Name ist vom Anwender definiert
  GUDVar1T :
  STRUCT
    SYNTAX_ID : BYTE;
    bereich_u_einheit : BYTE;
    spalte : WORD;
    zeile : WORD;
    bausteintyp : BYTE;
    ZEILENANZAHL : BYTE;
    typ : BYTE;
    laenge : BYTE;
  END_STRUCT;
END_STRUCT;

BEGIN
END_DATA_BLOCK
DATA_BLOCK DB 111 //freier Anwender-DB, als Instanz für FB 5
FB 5
```

```
BEGIN
END_DATA_BLOCK
//Eine Anwenderdefinierte Kanalvariable aus Kanal 1 soll gelesen werden mit
//einer Konvertierung in einen Variablen-Zeiger für ein nachfolgendes
//Schreiben dieser Variable.
Function FC "VariablenCall" : VOID
    U   E 7.7;                //freie Taste Maschinensteuertafel
    S   M 100.0;              //Req aktivieren
    U   M 100.1;              //Fertigmeldung Done
    R   M 100.0;              //Auftrag beenden
    U   E 7.6;                //Fehlerquittierung von Hand
    U   M 102.0;              //Fehler steht an
    R   M 100.0;              //Auftrag beenden

    CALL FB 5, DB 111(
        Req :=          M 100.0,                //Startflanke für Lesen
        Addr :=         GUDVAR.GUDVar1,
        Area :=         B#16#2,                //Kanal-Variable
        Unit :=         B#16#1,                //Kanal 1
        Index1 :=       0,                    //kein Feldindex
        Index2 :=       0,                    //kein Feldindex
        CnvtToken :=    TRUE,                 //Umsetzung in 10 Byte Token
        FMNCNo :=       1                     //(nur bei FM-NC)
        VarToken :=     GUDVAR.GUDVar1T,
        Error :=        M102.0,
        Done :=         M100.1,
        State :=        MW104,
        RD :=           P#DB99.DBX0.0 DINT 1);
```

## 2.12.6 FB 7: PI\_SERV2 Allgemeine PI Dienste

### Funktionsbeschreibung

Die ausführliche Beschreibung des FB 7 ist in der Beschreibung des FB 4 enthalten.  
Der Aufruf ist nur im zyklischen Programm OB 1 zulässig.

Der einzige Unterschied zu FB 4 ist die Anzahl der WVar1 und folgender Parameter. Im FB 7 ist WVar1 bis WVar16 definiert in der VAR\_INPUT (FB4 hat WVar1 bis WVar10). Alle anderen Parameter sind identisch zum FB 4.

Dieser PI-Server kann für alle PI-Dienste, die bisher mit FB 4 realisiert wurden, verwendet werden. Zusätzlich lassen sich unten aufgeführte PI-Dienste nur mit dem FB 7 abwickeln.

### Deklaration der Funktion

```
FUNCTION_BLOCK FB 7
Var_INPUT
    Req:          BOOL;
    PIService:    ANY;
    Unit:         INT;
    Addr1:        ANY;
    Addr2:        ANY;
    Addr3:        ANY;
    Addr4:        ANY;
    WVar1:        WORD;
    WVar2:        WORD;
    WVar3:        WORD;
    WVar4:        WORD;
    WVar5:        WORD;
    WVar6:        WORD;
    WVar7:        WORD;
    WVar8:        WORD;
    WVar9:        WORD;
    WVar10:       WORD;
    WVar11:       WORD;
    WVar12:       WORD;
    WVar13:       WORD;
    WVar14:       WORD;
    WVar15:       WORD;
    WVar16:       WORD;
    FMNCNo :     INT;                               //(nur bei FM-NC)
END_VAR
VAR_OUTPUT
    Error:       BOOL;
    Done:        BOOL;
    State:       WORD;
END_VAR
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion PI\_SERV2.

Signal	Art	Typ	Wertebereich	Bemerkung
Req	E	BOOL		Auftragsanforderung
PIService	E	ANY	[DBName].[VarName] Standard ist: "PI".[VarName]	PI Dienst Beschreibung
Unit	E	INT	1...	Bereichsnummer
Addr1 bis Addr4	E	ANY	[DBName].[VarName]	Referenz auf Strings Spezifikation, gemäß angewähltem PI-Dienst
WVar1 bis WVar16	E	WORD	1...	Integer oder Word-Variablen. Spezifikation gemäß angewähltem PI-Dienst,
FMNCNo (nur FM-NC)	E	INT	0, 1, 2	0, 1=1 NCU, 2=2 NCU`s
Error	A	BOOL		Auftrag wurde negativ quittiert, bzw. konnte nicht ausgeführt werden
Done	A	BOOL		Auftrag wurde erfolgreich ausgeführt.
State	A	WORD		siehe Fehlerkennungen

### Überblick zusätzliche PI Dienste

Folgende zusätzliche PI-Dienste zu denen von FB 4 sind vom PLC aus startbar. Die Verwendung und Bedeutung der allgemeinen Eingangsvariablen des FB 7 (Unit, Addr..., WVar...) ist vom jeweiligen PI-Dienst abhängig.

PI-Dienst	Funktion	Verfügbar in
TMFPBP	Leerplatzsuche	<b>SINUMERIK 840D/810D</b> *(ab SW 4)

**PI-Dienst: TMFPBP**

**Funktion: Leerplatzsuche**  
(in Abhängigkeit der Parameterbelegung)

Der Dienst sucht unter den in der Parametrierung angegebenen Magazin(en) einen Leerplatz, welcher den angegebenen Kriterien (Werkzeuggröße und Platztyp genügt. Das Ergebnis der Leerplatzsuche kann unter den Variablen magCMCmdPar1 (Magazin-Nummer) und magCMCmdPar2 (Platznummer) im Baustein TMC bei erfolgreichem Dienst abgeholt werden. Da der PI-Dienst ein Ergebnis in den Variablen magCMCmdPar1 und magCMCmdPar2 zurückliefert, ist der Dienst im Falle, dass mehrere Bedieneinheiten, PLC an einer NC arbeiten, mit dem Semaphore-Mechanismus (PI-Dienst MMCSEM) mit der Funktionsnummer für \_N\_TMFDP abzusichern. Der Suchbereich kann über die Belegung der Parameter "MagazinNummer\_Von", "PlatzNummer\_Von", "MagazinNummer\_Bis", "PlatzNummer\_Bis" wie folgt vorgegeben werden:

MagazinNummer_Von	PLatzNummer_Von	MagazinNummer_Bis	PLatzNummer_Bis	Suchbereich
WVar1	WVar2	WVar3	WVar4	
#M1	#P1	#M1	#P1	es wird nur der Platz #P1 im Magazin #M1 überprüft
#M1	#P1	#M2	#P2	es werden die Plätze beginnend von Magazin #M1, Platz #P1 bis Magazin #M2, Platz #P2 durchsucht
#M1	-1	#M1	-1	es werden alle Plätze von Magazin #M1 - und nur diese - durchsucht
#M1	-1	-1	-1	es werden alle Plätze beginnend ab Magazin #M1 durchsucht
#M1	#P1	-1	-1	es werden alle Plätze beginnend ab Magazin #M1 und darin Platz #P1 durchsucht
#M1	#P1	#M1	-1	es werden die Plätze in Magazin #M1 beginnend ab Magazin #M1 und darin Platz #P1 durchsucht
#M1	#P1	#M2	-1	es werden die Plätze beginnend ab Magazin #M1 und darin Platz #P1 bis einschließlich Magazin #M2 durchsucht
#M1	-1	#M2	#P2	es werden die Plätze beginnend ab Magazin #M1 bis einschließlich Magazin #M2 und darin Platz #P2 durchsucht
#M1	-1	#M2	-1	es werden die Plätze beginnend ab Magazin #M1 bis einschließlich Magazin #M2 durchsucht
-1	-1	-1	-1	es werden alle Magazinplätze durchsucht

**Hinweis**

Vor und nach diesem PI-Dienst ist der PI-Dienst MMCSEM mit entsprechendem Parameter "WVar1" für diesen PI-Dienst aufzurufen. Näheres siehe PI-Dienst MMCSEM.

Parametrierung			
Signal	Typ	Wertebereich	Bedeutung
PIService	ANY	PI.TMFPBP	Leerplatzsuche
Unit	INT	1 ... max. TOA	TOA
WVar1	INT		MagazinNummer_Von: Magazinnummer des Magazins, ab dem der Suchbereich beginnen soll
WVar2	INT		PlatzNummer_Von: Platznummer des Platzes in dem Magazin MagazinNummer_Von, ab dem der Suchbereich beginnen soll
WVar3	INT		MagazinNummer_Bis: Magazinnummer des Magazins, bei dem der Suchbereich enden soll
WVar4	INT		PlatzNummer_Bis: Platznummer des Platzes in dem Magazin MagazinNummer_Bis, bei dem der Suchbereich enden soll
WVar5	INT		MagazinNummer_Bezug:
WVar6	INT		PlatzNummer_Bezug:
WVar7	INT	0, 1 .. 7	Anzahl der benötigten Halbplätze nach links
WVar8	INT	0, 1 .. 7	Anzahl der benötigten Halbplätze nach rechts
WVar9	INT	0, 1 .. 7	Anzahl der benötigten Halbplätze nach oben
WVar10	INT	0, 1 .. 7	Anzahl der benötigten Halbplätze nach unten
WVar11	INT		Nummer des gewünschten Platztyps
WVar12	INT	0: default 1: vorwärts 2: rückwärts 3: symmetrisch	spezifiziert die gewünschte Suchrichtung 0: Leerplatz Suchstrategie wie in \$TC_MAMP2 eingestellt

## 2.12.7 FB 9: M zu N Bedieneinheitenumschaltung

### Funktionsbeschreibung

Dieser Baustein ermöglicht die Umschaltung mehrerer **Bedieneinheiten** (Bedientafeln, MMC Bedienhandgeräte und/oder Maschinensteuertafeln MSTT), die über ein Bussystem mit einer /mehreren Steuerungsbaugruppen NCU verbunden sind.

**Literatur:**

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Mehrere Bedientafelfronten an mehreren NCUs, Dezentrale Systeme (B3).

Die **Schnittstelle** zwischen den einzelnen Bedieneinheiten und der NCU (PLC) ist die M : N Schnittstelle im Datenbaustein DB 19; siehe /FB2/ Funktionshandbuch Erweiterungsfunktionen, Kapitel "Datenlisten, Signalbeschreibung".

Der FB 9 arbeitet mit den Signalen dieser Schnittstellen.

Folgende **Grundfunktionen** werden neben der Initialisierung, der Lebenszeichenüberwachung und der Fehlerrouinen vom Baustein für die Bedieneinheitenumschaltung ausgeführt:

Tabellarische Übersicht der Funktionen:	
Grundfunktion	Bedeutung
MMC klopft an	MMC hat den Wunsch an eine NCU online zu gehen
MMC kommt	MMC verbindet sich mit einer NCU
MMC geht	MMC trennt sich von einer NCU
Verdrängung	MMC muss Verbindung zu einer NCU lösen
Bedienfocuswechsel im Serverbetrieb	Bedienfocus von einer auf die andere NCU umschalten
Aktiver/Passiver Bedienmodus	Bedienen und beobachten/nur beobachten
MSTT-Umschaltung	MSTT kann optional mit dem MMC umgeschaltet werden

### Kurzbeschreibung einiger wichtiger Funktionen

Aktiver/Passiver Bedienmodus:

Ein Online-MMC kann zwei verschiedene Bedienmodi haben:

Aktiv-Modus: Bediener kann bedienen und beobachten

Passiv-Modus: Bediener kann beobachten (nur Kopfzeile MMC)

Nach Umschaltung auf eine NCU fordert sie in der PLC der Online-NCU zuerst den aktiven Bedienmodus an. Falls zwei MMC's zu einer Zeit an einer NCU online sind, ist eine der beiden immer im aktiven, die andere im passiven Bedienmodus. Der Bediener kann den aktiven Bedienmodus an der passiven Bedieneinheit per Tastendruck anfordern.

### MSTT-Umschaltung

Zusammen mit dem MMC kann optional eine ihm zugeordnete MSTT mit umgeschaltet werden. Voraussetzung ist, dass die MSTT-Adresse im Parameter "mstt\_adress" der Konfigurationsdatei NETNAMES.INI der Bedientafel eingetragen und "MCPEnable" = TRUE gesetzt ist. Die MSTT der passiven MMC ist deaktiviert. Dadurch gibt es an einer NCU immer nur eine aktive MSTT.

### Hochlaufbedingung

Um zu verhindern, dass bei einem Neustart der NCU, die davor zuletzt angewählte MSTT aktiviert wird, muss beim Aufruf von FB1 in OB100 der Eingangsparameter "MCP1BusAdr" =255 (Adresse 1. MSTT) und "MCP1STOP" = TRUE (1.MSTT ausschalten) gesetzt werden.

### Freigaben

Wenn von einer MSTT auf eine andere umgeschaltet wird, dann bleiben eventuell eingeschaltete Vorschub- sowie Achsfreigaben erhalten.

---

#### Hinweis

Die zur Zeit der Umschaltung betätigten Tasten wirken bis zur Aktivierung der neuen MSTT (vom MMC, die anschließend aktiviert wird) weiter. Auch die Overridestellungen für Vorschub, Spindel bleiben erhalten. Um die betätigten Tasten zu deaktivieren, ist bei fallender Flanke des Signals DB10, ... DBX104.0 (MSTT 1 bereit) das Eingangsabbild der Maschinensteuer-Signale auf nicht betätigte Signalpegel zu legen. Die Overridestellungen sollten unverändert bleiben.

Maßnahmen zur Deaktivierung der Tasten sind im PLC-Anwenderprogramm zu realisieren, siehe Beispiel "Override-Umschaltung".

Der Aufruf ist nur im zyklischen Programm OB 1 zulässig.

---



## Deklaration der Funktion

```
FUNKTION_BLOCK FB 9
VAR_INPUT
    Quit:          BOOL;          //Quittierung Alarme
    OPMixedMode:   BOOL:= FALSE;  //Mischbetrieb mit nicht M zu N fähigen OP
                                //deaktiviert
    ActivEnable:   BOOL:= TRUE;   //wird nicht unterstützt.
                                //Bedienfeldumschaltung Verriegelung über
                                //MMCx_SHIFT_LOCK im DB 19
    MCPEnable:     BOOL:= TRUE;   //MSTT Umschaltung aktivieren
END_VAR
VAR_OUTPUT
    Alarm1:        BOOL;          //Alarm: Fehler HMI-Busadresse, Bustyp!
    Alarm2:        BOOL;          //Alarm: Keine Bestätigung MMC_1 offline!
    Alarm3:        BOOL;          //Alarm: MMC_1 geht nicht offline!
    Alarm4:        BOOL;          //Alarm: Keine Bestätigung MMC_2 offline!
    Alarm5:        BOOL;          //Alarm: MMC_2 geht nicht offline!
    Alarm6:        BOOL;          //Alarm: Anklopf-MMC geht nicht online!
    Report:        BOOL;          //Meldung: Lebenszeichenüberwachung MMC
    ErrorMMC:      BOOL;          //Fehlererkennung MMC
END_VAR
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion MzuN

Formalparameter der Funktion MzuN			
Signal	Art	Typ	Bemerkung
Quit	E	BOOL	Quittierung Alarme
OPMixedMode	E	BOOL	Mischbetrieb mit einem nicht M zu N fähigen OP
ActivEnable	E	BOOL	Funktion wird nicht unterstützt. Bedienfeldumschaltung Verriegelung über MMCx_SHIFT_LOCK im DB 19
MCPEnable	E	BOOL	MSTT Umschaltung aktivieren <b>TRUE</b> = MSTT wird mit Bedienfeld umgeschaltet. <b>FALSE</b> = MSTT wird nicht mit Bedienfeld umgeschaltet. Hiermit kann eine MSTT fest verbunden werden. Siehe auch MMCx_MSTT_SHIFT_LOCK im DB 19
Alarm1	A	BOOL	Alarm: Fehler MMC-Busadresse, Bustyp!
Alarm2	A	BOOL	Alarm: Keine Bestätigung MMC 1 offline!
Alarm3	A	BOOL	Alarm: MMC 1 geht nicht offline!
Alarm4	A	BOOL	Alarm: Keine Bestätigung MMC 2 offline!
Alarm5	A	BOOL	Alarm: MMC 2 geht nicht offline!
Alarm6	A	BOOL	Alarm: Anklopf-MMC geht nicht online!
Report	A	BOOL	Meldung: Lebenszeichenüberwachung MMC
ErrorMMC	A	BOOL	Fehlererkennung MMC

#### Hinweis

Der Baustein ist vom Anwenderprogramm aufzurufen. Hierbei ist ein Instanz-DB mit beliebiger Nummer vom Anwender beizustellen. Der Aufruf ist nicht Multi-Instanzfähig.

## Aufrufbeispiel für FB 9

```
CALL    FB 9, DB 109 (
    Quit           :=Fehler_Quitt,           //z.B. MSTT-RESET
    OPMixedMode    :=FALSE,
    ActivEnable    :=TRUE,                   //
    MCPEnable      :=TRUE,                   //Freigabe MSTT Umschaltung
    Alarm1         :=DB2.dbx188.0,           //Fehlermeldung 700.100
    Alarm2         :=DB2.dbx188.1,           // Fehlermeldung 700.101
    Alarm3         :=DB2.dbx188.2,           //Fehlermeldung 700.102
    Alarm4         :=DB2.dbx188.3,           //Fehlermeldung 700.103
    Alarm5         :=DB2.dbx188.4,           //Fehlermeldung 700.104
    Alarm6         :=DB2.dbx188.5,           //Fehlermeldung 700.105
    Report         :=DB2.dbx192.0);          //Betriebsmeldung 700.132
```

---

### Hinweis

Der Eingangsparameter "MCPEnable" muss TRUE sein, um die MSTT Umschaltung zu ermöglichen. Der Defaultwert dieser Parameter ist so geschaltet und muss beim Aufruf der Funktion nicht zusätzlich beschaltet werden.

---

## Alarm, Fehler

Die Ausgangsparameter "Alarm1" bis "Alarm6" und "Report" können für die Alarm- und Fehlermeldungen des MMC in den Bereichen des DB 2 übergeben werden.

Konnte eine MMC-Funktion nicht ausgeführt werden (bei der die Fehlermeldung nicht angezeigt werden kann), wird dies am Zustandsparameter "ErrorMMC" mit logisch 1 angezeigt (z.B. Fehler im Hochlauf, wenn keine Verbindung aufgebaut wird).

## Aufrufbeispiel für FB 1 (Aufruf im OB 100)

```
CALL "RUN_UP" , "gp_par" (
    MCPNum         := 1,
    MCP1In         := P#E 0.0,
    MCP1Out        := P#A 0.0,
    MCP1StatSend   := P#A 8.0,
    MCP1StatRec    := P#A 12.0,
    MCP1BusAdr     := 255,                       //Adresse 1. MSTT
    MCP1Timeout    := S5T#700MS,
    MCP1Cycl       := S5T#200MS,
    MCP1Stop      := TRUE,                       //MSTT ausgeschaltet
    NCCyclTimeout  := S5T#200MS,
    NCRunupTimeout := S5T#50S);
```

### Beispiel für Override-Umschaltung

```
//verwendete Hilfsmerker M100.0, M100.1, M100.2, M100.3
//Flanke positiv von MCP1Ready muss Override prüfen und Maßnahmen für Aktivierung
//MSTT Baustein einleiten
//Dieses Beispiel gilt für Vorschub-Override;
//für Spindel-Override sind Nahtstellen- und Eingangsbyte auszutauschen.
U   DB10.DBX  104.0;    //MCP1Ready
FN  M         100.0;    //Flankenmerker 1
SPBN weil;
S   M         100.2;    //Hilfsmerker 1 setzen
R   M         100.3;    //Hilfsmerker 2 rücksetzen

//Override speichern
    L DB21.DBB 4;        //Nahtstelle Vorschub-Override
    T EB 28;            //Zwischenspeicher (freies Eingangs- oder Merkerbyte)
weil:
U   M         100.2;    //Umschaltung erfolgt
O   DB10.DBX  104.0;    //MCP1Ready
SPBN weil2;
U   DB10.DBX  104.0;    //MCP1Ready
FP  M         100.1;    //Flankenmerker 2
SPB weil2;
U   M         100.2;    //Umschaltung erfolgt
R   M         100.2;    //Hilfsmerker 1 rücksetzen
SPB weil2;
U   M         100.3;    //Vergleich ist erfolgt
SPB MCP;                //MSTT-Programm aufrufen
//gespeicherten Override auf Nahtstelle der umgeschalteten MSTT lenken
//bis die Override-Werte übereinstimmen
L   EB28;                //Zwischenspeicher auf
T DB21.DBB 4;            //Nahtstelle Override lenken
L EB 3;                  //Override-Eingangsbyte für Vorschub
<>i;                      //Übereinstimmung?
SPB weil2;                //nein, Absprung
S   M100.3;              //ja, Hilfsmerker 2 setzen
//nach Übereinstimmung der Override-Werte MSTT-Programm wieder aufrufen
MCP: CALL "MCP_IFM" (     //FC 19
    BAGNo      := B#16#1,
    ChanNo     := B#16#1,
    SpindleIFN := B#16#0,
    o
    FeedHold   := M 101.0,
    SpindleHol := M 101.1);
    d
weil2: NOP                0;
```

## 2.12.8 FB 10: Sicherheits-Relais (SI-Relais)

### Funktionsbeschreibung

Der SPL-Baustein "Sicherheitsrelais" für "Safety Integrated" ist das Äquivalent auf der PLC zu der gleichnamigen NC-Funktion. Der Standard-SPL-Baustein "Sicherheitsrelais" ist für die Anforderungen einer NOT-HALT Realisierung mit sicherer programmierbarer Logik ausgelegt. Er kann aber auch für weitere ähnliche Anforderungen, z.B. einer Schutztür-Ansteuerung, genutzt werden. Die Funktion enthält 3 Eingangsparameter (In1, In2, In3). Bei Schalten eines dieser Parameter auf den Wert 0, wird der Ausgang Out0 unverzüglich abgeschaltet und die Ausgänge Out1, Out2 und Out3 über die parametrisierten Zeitwerte (Parameter TimeValue1, TimeValue2, TimeValue3) verzögert abgeschaltet. Die Ausgänge werden unverzüglich wieder eingeschaltet, wenn die Eingänge In1 bis In3 den Wert 1 annehmen und an einem der Quittierungseingänge Quit1, Quit2 ein positiver Flankenwechsel erkannt wird. Um die Ausgänge in Grundstellung (Werte = 0) nach Neueinschalten zu bringen, ist der Parameter "FirstRun" wie folgt zu beschalten. Der Parameter "FirstRun" muss beim 1. Durchlauf nach Hochlauf der Steuerung über ein remanentes Datum (Merkerbit, Bit im Datenbaustein) auf den Wert TRUE geschaltet sein. Das Datum kann z.B. im OB 100 vorbesetzt werden. Der Parameter wird nach dem erstmaligen Durchlauf des FB 10 auf den Wert FALSE zurückgesetzt. Für jeden Aufruf mit eigener Instanz ist für den Parameter "FirstRun" ein separates Datum zu verwenden.

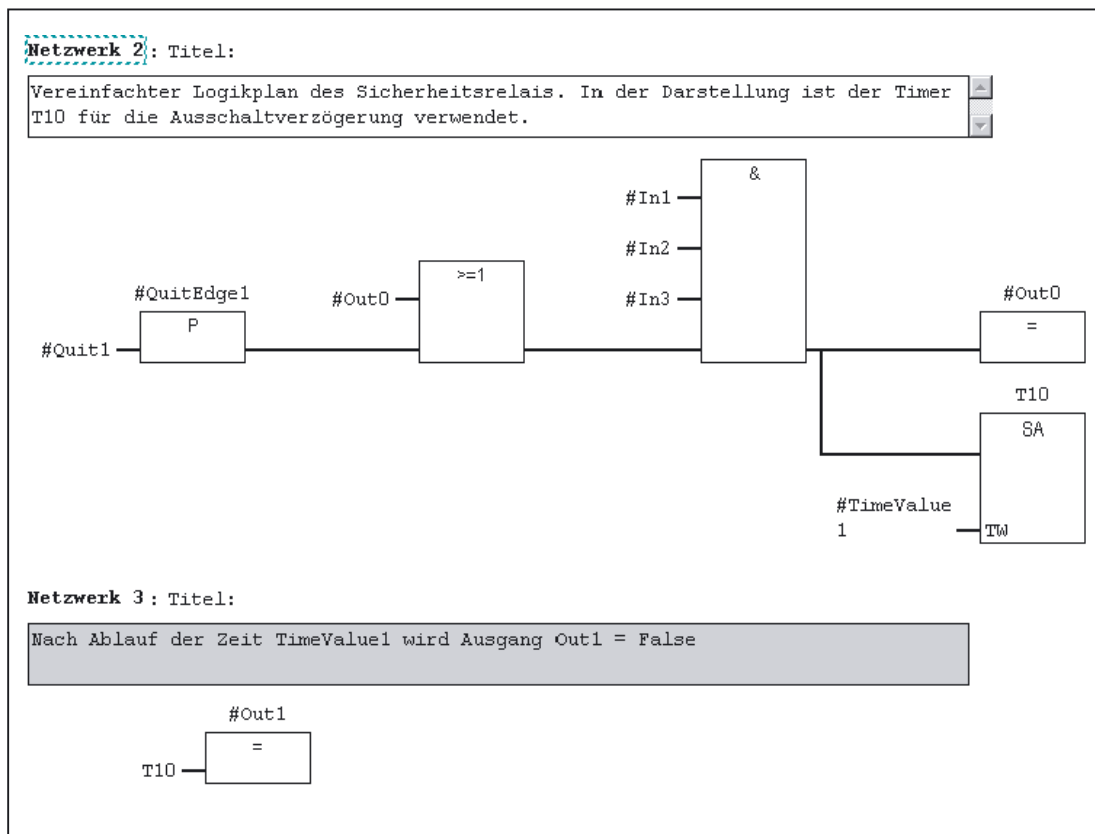
Der entsprechende NCK-SPL-Baustein ist beschrieben in:

#### **Literatur:**

/FBSI/ Funktionsbeschreibung Safety Integrated

### Vereinfachtes Prinzipschaltbild im Funktionsplan

In der folgenden Darstellung sind nur ein Quittiereingang Quit1 und ein ausschaltverzögerter Ausgang Out1 dargestellt. Schaltung für Quit2 und die weiteren ausschaltverzögerten Ausgänge sind identisch realisiert. In dem Funktionsplan fehlt auch der Parameter FirstRun. Die Wirkungsweise ist oben beschrieben.



## Deklaration der Funktion

```
FUNCTION_BLOCK FB 10
VAR_INPUT
    In1 : BOOL      := TRUE;           //Input 1
    In2 : BOOL      := TRUE;           //Input 2
    In3 : BOOL      := TRUE;           //Input 3
    Quit1:          : BOOL;            //Quit 1 Signal
    Quit2:          : BOOL;            //Quit 2 Signal
    TimeValue1:     TIME := T#0ms;     //TimeValue for Output 1
    TimeValue2:     TIME := T#0ms;     //TimeValue for Output 2
    TimeValue3:     TIME := T#0ms;     //TimeValue for Output 3
END_VAR
VAR_OUTPUT
    Out0           : BOOL;             //Output without Delay
    Out1           : BOOL;             //Delayed Output to False by Timer 1
    Out2           : BOOL;             //Delayed Output to False by Timer 2
    Out3           : BOOL;             //Delayed Output to False by Timer 3
END_VAR
VAR_INOUT
    FirstRun       : BOOL;             //TRUE by User after 1. Start of SPL
END_VAR
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion SI-Relais

Formalparameter der Funktion SI-Relais			
Signal	Art	Typ	Bemerkung
In1	E	BOOL	Eingang 1
In2	E	BOOL	Eingang 2
In3	E	BOOL	Eingang 3
Quit1	E	BOOL	QuittierEingang 1
Quit2	E	BOOL	QuittierEingang 2
TimeValue1	E	TIME	Zeit-Wert 1 für Ausschaltverzögerung
TimeValue2	E	TIME	Zeit-Wert 2 für Ausschaltverzögerung
TimeValue3	E	TIME	Zeit-Wert 3 für Ausschaltverzögerung
Out0	A	BOOL	Ausgang unverzögert
Out1	A	BOOL	Ausgang verzögert mit TimeValue1
Out2	A	BOOL	Ausgang verzögert mit TimeValue2
Out3	A	BOOL	Ausgang verzögert mit TimeValue3
FirstRun	E/A	BOOL	Aktivierung der Grundstellung

---

#### Hinweis

Der Baustein ist vom Anwenderprogramm zyklisch aufzurufen ab dem Starten des SPL-Programms. Hierbei ist ein Instanz-DB mit beliebiger Nummer vom Anwender beizustellen. Der Aufruf ist Multi-Instanzfähig.

---



## 2.12.9 FB 11: Bremsentest

### Funktionsbeschreibung

Die Funktionsprüfung der Bremsenmechanik sollte für alle Achsen verwendet werden, die durch eine Haltebremse gegen Bewegungen im ungeregelten Betrieb gesichert werden müssen. Der Hauptanwendungsfall sind dabei die sogenannten "hängenden Achsen". In seinem PLC-Anwenderprogramm kann der Maschinenhersteller zu einem geeigneten Zeitpunkt (Richtwert alle 8h, wie auch beim SI-Teststopp) die Bremse schließen und vom Antrieb ein zusätzliches Moment / eine zusätzliche Kraft zu der Gewichtskraft der Achse aufbringen lassen. Im fehlerfreien Zustand kann die Bremse das notwendige Bremsmoment / die Bremskraft aufbringen, die Achse wird sich kaum bewegen. Im Fehlerfall wird ein Verlassen des parametrierten Überwachungsfensters für den Positionswert erkannt. In diesem Fall wird durch den Lagereger ein Absacken der Achse verhindert und die Funktionsprüfung der Bremsenmechanik negativ quittiert.

Die notwendige Parametrierung von NC und Antrieb ist beschrieben in:

#### **Literatur:**

/FBSI/ Funktionsbeschreibung Safety Integrated

Der Start des Bremsentests muss immer im Stillstand der Achse erfolgen. Für den gesamten Zeitraum des Bremsentests sind die Freigabesignale der parametrierten Achse auf Freigabe zu setzen (z.B. die Signale Reglersperre, Vorschubfreigabe). Weiterhin ist das Signal an Achse/Spindel DB31, ... DBX28.7 (PLC kontrolliert Achse) während der gesamten Dauer des Tests auf Zustand 1 vom Anwenderprogramm zu setzen.

Vor Aktivierung des Signals DB31, ... DBX28.7 (PLC kontrolliert Achse) ist die Achse als "neutrale Achse" zu schalten z.B. ist das DB31, ... DB8.0-8.3 (NC-Achse Kanal zuordnen) auf Kanal 0 zu setzen sowie das DB31, ... DB8.4 (Aktiviersignal bei Veränderung dieses Bytes) zu setzen.

Die Rückmeldung

- über den aktuellen Zustand kann im Byte DB31, ... DBB68 abgefragt werden.
- der NC über das Signal DB31, ... DBX63.1 (PLC kontrolliert Achse) ist abzuwarten bevor der Baustein gestartet wird. Die Richtung, in die der Antrieb sein Moment / seine Kraft aufbringt, wird von der PLC durch eine "Verfahrbewegung" (z.B. über den FC 18) vorgegeben.

Der Zielpunkt dieser Verfahrbewegung muss gefahrlos erreicht werden können (keine Kollision), falls die Bremse das notwendige Moment / die Kraft nicht aufbringen kann.

---

#### **Hinweis**

##### **Hinweis zu FC 18**

Falls FC 15, FC 16 oder FC 18 für die gleiche Achse im weiteren Anwenderprogramm aufgerufen werden, müssen die Aufrufe gegeneinander verriegelt werden. Dies kann z.B. über einen gemeinsamen Aufruf dieser Funktion mit einer verriegelten gemeinsamen Datenschnittstelle für die Parameter des FC 18 erfolgen. Eine 2. Möglichkeit ist der Mehrfachaufruf des FC 18, wobei der inaktive FC 18 programmäßig nicht durchlaufen wird. Eine Verriegelung für Mehrfachnutzung ist vorzusehen.

---

Der Bremsentest gliedert sich in folgende Schritte:

Bremsentest-Gliederung		
Schritt	Erwartete Rückmeldung	Überwachungs-Zeitwert
Bremsentest starten	DBX 71.0 = 1	TV_BTactiv
Bremse schließen	Bclosed = 1	TV_Bclose
Fahrbefehl ausgeben	DBX 64.6 Or DBX 64.7	TV_FeedCommand
Prüfung Fahrbefehl ausgegeben	DBX62.5 = 1	TV_FXSreached
Haltezeit abwarten	DBX62.5 = 1	TV_FXShold
Abwahl Bremsen Test / Bremsen öffnen	DBX71.0 = 0	TV_BTactiv
Test ok ausgeben		

### Deklaration der Funktion

```

Function_BLOCK FB 11
VAR_INPUT
    Start          : BOOL;           //Start for Braketest
    Quit           : BOOL;           //Quit Error
    Bclosed        : BOOL;           //brake closed input (single channel - PLC)
    Axis           : INT;            //testing axisno.
    TimerNo        : TIMER;          //Timer from User
    TV_BTactiv     : S5TIME;         //TimeValue -> braketest activ
    TV_Bclose      : S5TIME;         //TimeValue -> close Brake
    TV_FeedComman : S5TIME;         //TimeValue -> force FeedCommand
    d
    TV_FXSreached : S5TIME;         //TimeValue -> Fixed stop reached
    TV_FXShold     : S5TIME;         //TimeValue -> test brake
END_VAR
VAR_OUTPUT
    CloseBrake     : BOOL;           //Signal Close brake
    MoveAxis       : BOOL;           //do move axis
    Done           : BOOL;
    Error          : BOOL;
    State          : BYTE;           //Errorbyte
END_VAR
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion Bremsentest:

<b>Formalparameter der Funktion Bremsentest</b>			
<b>Signal</b>	<b>Art</b>	<b>Typ</b>	<b>Bemerkung</b>
Start	E	BOOL	Start des Bremsentests
Quit	E	BOOL	Fehler Quittierung
Bclosed	E	BOOL	Rückmelde-Eingang ob Bremse schließen angesteuert ist (einkanalig - PLC)
Axis	E	INT	<b>Achsnnummer der zu testenden Achse</b>
TimerNo	E	TIMER	Timer aus Anwenderprogramm
TV_BTactiv	E	S5TIME	Überwachungs-Zeitwert -> Bremsentest aktiv, Prüfung des Achs-Signals DBX71.0
TV_Bclose	E	S5TIME	Überwachungs-Zeitwert -> Bremse schließen. Prüfung des Eingangssignals Bclosed nachdem Ausgang CloseBrake gesetzt wurde.
TV_FeedCommand	E	S5TIME	Überwachungs-Zeitwert -> Fahrbefehl ausgegeben. Prüfung Fahrbefehl nachdem MoveAxis gesetzt wurde.
TV_FXSreched	E	S5TIME	Überwachungs-Zeitwert -> Festanschlag erreicht
TV_FXShold	E	S5TIME	Überwachungs-Zeitwert -> Bremse testen
CloseBrake	A	BOOL	Anforderung für Bremse schließen
MoveAxis	A	BOOL	Anforderung Verfahrbewegung anstoßen
Done	A	BOOL	Test erfolgreich beendet
Error	A	BOOL	Fehler aufgetreten.
State	A	BYTE	Fehler Status

## Fehlererkennungen

State	Bedeutung
0	Kein Fehler
1	Startbedingungen nicht gegeben, z.B. Achse nicht in Regelung / Bremse zu / Achsensperre steht an
2	Bei Anwahl Bremsentest erfolgt keine Rückmeldung der NC im Signal "Bremsentest aktiv"
3	Keine Rückmeldung "Bremse eingefallen" durch Eingangssignal BClosed
4	Keine Fahrbefehlausgabe (z.B. Bewegung der Achse wurde nicht gestartet)
5	Festanschlag wird nicht erreicht -> Achs-RESET wurde ausgelöst.
6	Verfahrssperre/Anfahrt zu langsam -> Festanschlag kann nicht erreicht werden. Überwachungszeit TV FXSreached abgelaufen.
7	Bremse hält überhaupt nicht (Endposition wird erreicht) / Anfahrgeschwindigkeit zu groß
8	Bremse bricht während der Haltezeit auf
9	Fehler bei Bremsentest-Abwahl
10	interner Fehler
11	Signal "PLC kontrolliert Achse" nicht eingeschaltet vom Anwenderprogramm

---

### Hinweis

Der Baustein ist vom Anwenderprogramm aufzurufen. Hierbei ist ein Instanz-DB mit beliebiger Nummer vom Anwender beizustellen. Der Aufruf ist Multi-Instanzfähig.

---

### Aufrufbeispiel für FB 11:

```
UN      M      111.1; //Anforderung Bremse schließen Z-Achse von FB
=       A      85.0;  //Ansteuerung Bremse Z-Achse
AUF                    "Axis3"; //Bremsentest Z-Achse
O       E      73.0;  //Anstoß für den Bremsentest Z-Achse
O       M      110.7; //Bremsentest läuft
FP      M      110.0;
UN      M      111.4; //Fehler aufgetreten
S       M      110.7; //Bremsentest läuft
S       M      110.6; //nächster Schritt
S       DBX    8.4;   //Neutrale Achse anfordern

U       DBX    68.6;  //Rückmeldung Achse ist neutral
U       M      110.6;
FP      M      110.1;
R       M      110.6;
S       M      110.5; //nächster Schritt
R       DBX    8.4;
S       DBX    28.7; //PLC-kontrollierte Achse anfordern

U       DBX    63.1;  //Rückmeldung Achse wird von PLC kontrolliert
U       M      110.5;
FP      M      110.2;
R       M      110.5;
S       M      111.0; //Start Bremsentest für FB

CALL FB 11 , DB 211 (//Bremsentestbaustein
    Start           := M      111.0, //Start des Bremsentests
    Quit            := E      3.7,  //Fehlerquittierung mit RESET-Taste
    Bclosed         := E      54.0,  //Rückmeldung Bremse schließen
                                //angesteuert
    Axis            := 3,           //Achsnnummer der zu testenden
                                //Achse Z-Achse
    TimerNo         := T      110,  //Timernummer
    TV_BTactiv      := S5T#200MS,  //Überwachungs-Zeitwert:
                                //Bremsentest aktiv DBX71.0
    TV_Bclose       := S5T#1S,     //Überwachungs-Zeitwert:
                                //Bremse geschlossen
    TV_FeedCommand  := S5T#1S,     //Überwachungs-Zeitwert:
                                //Fahrbefehl ausgegeben
    TV_FXSreache    := S5T#1S,     //Überwachungs-Zeitwert:
                                //Festanschlag erreicht
    TV_FXShold      := S5T#2S,     //Überwachungs-Zeitwert: Testzeit Bremse
    CloseBrake      := M      111.1, //Anforderung Bremse schließen
```

```

MoveAxis      := M    111.2, //Anforderung Verfahrbewegung anstoßen
Done          := M    111.3, //Test erfolgreich beendet
Error        := M    111.4, //Fehler aufgetreten
State        := MB   112); //Fehler Status

AUF           "Axis3" //Bremsentest Z-Achse
;

O    M        111.3; //Test erfolgreich beendet
O    M        111.4; //Fehler aufgetreten
FP   M        110.3;
R    DBX     28.7; //Anforderung PLC-kontrollierte Achse
UN   DBX     63.1; //Rückmeldung Achse wird von PLC kontrolliert
U    M        111.0; //Start Bremsentest für FB
U    M        110.7; //Bremsentest läuft
FP   M        110.4;
R    M        111.0; //Start Bremsentest für FB
R    M        110.7; //Bremsentest läuft

CALL "SpinCtrl" (//Z-Achse verfahren
    Start      := M    111.2, //Start des Verfahrbewegung
    Stop       := FALSE,
    Funct      := B#16#5, //Mode: Achsbetrieb
    Mode       := B#16#1, //Verfahren: Inkrementell
    AxisNo     := 3, //Achsnnummer der zu verfahrenenden
                //Achse Z-Achse
    Pos        := - //Verfahrweg: minus 5 mm
                5.000000e+000,
    FRate      := 1.000000e+003, //Vorschub: 1000 mm/min
    InPos      := M    113.0, //Position erreicht
    Error      := M    113.1, //Fehler aufgetreten
    State      := MB   114); //Fehler Status

```

## 2.12.10 FB 29: Diagnose Signalrekorder und Datentrigger

### Signalrekorder

Mit dem FB "Diagnose" besteht die Möglichkeit verschiedene Diagnosen am PLC Anwenderprogramm durchzuführen. Ein Diagnosefall ist das Protokollieren von Signalzuständen und Signaländerungen. Bei diesem Diagnosefall wird die Funktionsnummer 1 dem Parameter "Func" zugeordnet. Bis zu 8 Signale der Parameter "Signal\_1" bis "Signal\_8" werden bei Änderung von einem dieser Signale in einem Ringpuffer aufgezeichnet. Zusätzlich werden noch die aktuellen Informationen der Parameter "Var1" als BYTE-Wert sowie "Var2", "Var3" als INTEGER-Werte im Ringpuffer mit abgelegt.

Weiterhin wird die Anzahl der vergangenen OB 1 Zyklen als Zusatzinformation im Ringpuffer abgelegt. Hiermit ist eine graphische Auswertung von Signalen und Werten im OB 1 Zyklusraster möglich. Beim ersten Aufruf des FB "Diagnose" innerhalb eines neuen PLC-Zyklus muss der Parameter "NewCycle" auf TRUE parametrieren sein. Bei mehrfachen Aufrufen des FB "Diagnose" in einem OB 1 Zyklus muss beim zweiten und folgenden Aufruf der Parameter "NewCycle" FALSE sein. Hiermit wird verhindert, dass eine neue Anzahl von OB 1 Zyklen berechnet wird.

Der Ringpuffer wird vom Anwender zur Verfügung gestellt muss eine ARRAY-Struktur wie im Quellcode angegeben haben. Die Anzahl der ARRAY Elemente ist beliebig. Empfohlen wird eine Größe von 250 Elementen. Über den Parameter "ClearBuf" wird der Ringpuffer gelöscht und der Zeiger BufAddr (Ein-/Ausgangsparameter) auf den Anfang gesetzt. Der zugehörige Instanz-DB zum FB ist ein DB aus dem Anwenderbereich und ist an dem FB "Diagnose" mit dem Parameter "BufDB" zu übergeben.

### Datentrigger

Die Funktion Datentrigger soll eine Triggerung auf bestimmte Werte (auch Bits) an einer beliebigen zulässigen Speicherzelle ermöglichen. Hierbei wird die zu triggernde Zelle mit einer Bitmaske (Parameter "AndMask") vor dem Vergleich des Parameters "TestVal" in dem Diagnose-Baustein "verundet".

---

#### Hinweis

Die Funktion steht als Quelle im Quellcontainer der Grundprogramm-Bibliothek mit Namen Diagnose.awl. zur Verfügung. In diesem Quellbaustein sind auch Instanz-DB und Ringpuffer DB definiert. Weiterhin ist der Aufruf der Funktion dargestellt. Hierbei sind die DB Nummern und auch der Aufruf anzupassen.

---

## Deklaration der Funktion

```
FUNCTION_BLOCK FB 29
VAR_INPUT
Func          : INT;          //Function number: 0 = No Function,
                               //1 = Signalrecorder, 2 = Datatrigger
    Signal_1   : BOOL;        //Start for Braketest
    Signal_2   : BOOL;
    Signal_3   : BOOL;
    Signal_4   : BOOL;
    Signal_5   : BOOL;
    Signal_6   : BOOL;
    Signal_7   : BOOL;
    Signal_8   : BOOL;
    NewCycle   : BOOL;
    Var1       : BYTE;
    Var2       : INT;
    Var3       : INT;
    BufDB      : INT;
    ClearBuf   : BOOL;
    DataAdr    : POINTER;     //Area pointer to testing word
    TestVal    : WORD;        //Value for triggering
    AndMask    : WORD;        //AND- Mask to the testing word
END_VAR
VAR_OUTPUT
    TestIsTrue : BOOL;
END_VAR
VAR_IN_OUT
    BufAddr    : INT;
END_VAR
```



## Struktur Ringpuffer

```
TITLE =
                                     //Ring buffer-DB for FB 29
VERSION : 1.0

STRUCT
  Feld : ARRAY [0 .. 249 ] OF STRUCT //can be any size of this struct

  Cycle : INT;                       //Delta cycle to last storage in buffer
  Signal_1 : BOOL;                   //Signal names same as FB 29
  Signal_2 : BOOL;
  Signal_3 : BOOL;
  Signal_4 : BOOL;
  Signal_5 : BOOL;
  Signal_6 : BOOL;
  Signal_7 : BOOL;
  Signal_8 : BOOL;
  Var1 : BYTE;
  Var2 : WORD;
  Var3 : WORD;
  END_STRUCT;
END_STRUCT;
BEGIN
END_DATA_BLOCK
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion Diagnose:

Formalparameter der Funktion Diagnose				
Signal	Art	Typ	Wertebereich	Bemerkung
Func	E	INT	0, 1, 2	Funktion 0: Abschalten 1: Signalrekorder 2: Datentrigger
<b>Parameter für Funktion 1</b>				
Signal_1 bis Signal_8	E	BOOL		Bit-Signale, die auf Änderung geprüft werden
NewCycle	E	BOOL		Siehe obige Beschreibung "Signalrekorder"
Var1	E	BYTE		Zusatzwert
Var2	E	INT		Zusatzwert
Var3	E	INT		Zusatzwert
BufDB	E	INT		Ringpuffer DB-Nr.
ClearBuf	E	BOOL		Löschen des Ringpuffer DB mit Rücksetzen des Zeiger BufAddr
BufAddr	E/A	INT		Zielbereich für gelesene Daten
<b>Parameter für Funktion 2</b>				
DataAdr	E	POINTER		Zeiger auf zu prüfendes Wort
TestVal	E	WORD		Vergleichswert
AndMask	E	WORD		siehe Beschreibung
TestIsTrue	A	BOOL		Ergebnis des Vergleichs

### Projektierungsschritte

- Funktion des Diagnosebausteins auswählen.
- Festlegung der geeigneten Daten für die Aufzeichnung als Signalrekorder bzw. der Datentriggerung.
- Geeignete Stelle(n) im Anwenderprogramm suchen für den Aufruf des FB Diagnose.
- Erzeugung eines Datenbausteins für den Ringpuffer, siehe Aufrufbeispiel.
- Aufruf des FB Diagnose im Anwenderprogramm mit Parametern.

Bei der Funktion 1 ist ein Löschen des Ringpuffers mit dem Parameter "ClearBuf" sinnvoll. Nach Beendigung der Aufzeichnung mit Funktion 1 wird der Ringpuffer-DB über STEP 7 mit der Funktion "Datenbaustein öffnen in Datenansicht" ausgelesen. Die Inhalte des Ringpuffers-DB können nun analysiert werden.

## Aufrufbeispiel

```
FUNCTION FC 99: VOID
TITLE =
VERSION : 0.0

BEGIN
NETWORK
TITLE = NETWORK

CALL FB 29 , DB 80 (
Func                := 1,
    Signal_1        := M      100.0,
    Signal_2        := M      100.1,
    Signal_3        := M      100.2,
    Signal_4        := M      100.3,
    Signal_5        := M      10.4,
    Signal_6        := M      100.5,
    Signal_7        := M      100.6,
    Signal_8        := M      100.7,
    NewCycle        := TRUE,
    Var1            := MB      100,
    BufDB           := 81,
    ClearBuf        := M      50.0);
END_FUNCTION
```

### 2.12.11 FC 2: GP\_HP Grundprogramm, zyklischer Teil

#### Funktionsbeschreibung

Die komplette Bearbeitung der NCK-PLC-Nahtstelle erfolgt im zyklischen Betrieb. Um die Grundprogrammlaufzeit gering zu halten, werden nur die Steuer-/Statussignale zyklisch übertragen, die Hilfs- und G-Funktionsübergabe wird nur auf Anforderung vom NCK bearbeitet.

#### Deklaration

```
FUNCTION FC 2: VOID
//keine Parameter
```

## Aufrufbeispiel

Das Grundprogramm muss - zeitlich gesehen - **vor der** Bearbeitung des Anwenderprogramms durchlaufen werden. Deshalb ist es als erstes im OB 1 aufzurufen.

Das vorliegende Beispiel enthält die Standard-Deklarationen für den OB 1 und die Aufrufe für das Grundprogramm (FC 2), den Transfer der MCP-Signale (FC 19) sowie die Erfassung von Fehler- und Betriebsmeldungen (FC 10).

```
ORGANIZATION_BLOCK OB 1
VAR_TEMP
    OB1_EV_CLASS :          BYTE;
    OB1_SCAN_1   :          BYTE;
    OB1_PRIORITY :          BYTE;
    OB1_OB_NUMBR :          BYTE;
    OB1_RESERVED_1 :        BYTE;
    OB1_RESERVED_2 :        BYTE;
    OB1_PREV_CYCLE :        INT;
    OB1_MIN_CYCLE :        INT;
    OB1_MAX_CYCLE :        INT;
    OB1_DATE_TIME :        DATE_AND_TIME;
END_VAR
BEGIN
CALL FC 2;                               //Aufruf Grundprogramm als 1. FC
//HIER ANWENDERPROGRAMM EINFUEGEN
CALL FC 19(                               //MCP-Signale an Nahtstelle
    BAGNo :=                B#16#1,        //BAG Nr. 1
    ChanNo :=                B#16#1,        //Kanal Nr. 1
    SpindleIFNo :=          B#16#4,        //Spindel Interface Nummer = 4
    FeedHold :=             m22.0,        //Vorschub Halt Signal
                                        //selbsthaltend
    SpindleHold :=          db2.dbx151.0);  //Spindel Halt selbsthaltend
                                        //in Meldungs-DB
CALL FC 10(                               //Fehler- und Betriebsmeldungen
    ToUserIF :=            TRUE,          //Signale aus DB2 an Nahtstelle
                                        //übertragen
    Quit :=                E6.1);         //Quittierung der Fehlermeldungen
                                        //über E 6.1
END_ORGANIZATION_BLOCK
```

## 2.12.12 FC 3: GP\_PRAL Grundprogramm, alarmgesteuerter Teil

### Funktionsbeschreibung

Im alarmgesteuerten Teil des Grundprogramms erfolgt die Bearbeitung der satzsynchronen Übergaben von NCK an PLC (Hilfs- und G-Funktionen). Bei den **Hilfsfunktionen** wird zwischen normalen und schnellen Hilfsfunktionen unterschieden.

Schnelle Hilfsfunktionen

- eines NC-Satzes werden zwischengespeichert und die Übernahme an die NC quittiert. Zu Beginn des nächsten OB 1-Zyklus werden diese an die Anwendernahtstelle übergeben.
- die sofort hintereinander programmiert sind, gehen nicht für das Anwenderprogramm verloren. Hierfür sorgt ein Mechanismus im Grundprogramm.

Normale Hilfsfunktionen

werden erst dann an die NC quittiert, wenn an diesen eine Zyklusdauer angestanden hat. Damit hat der Anwender bei Bedarf die Möglichkeit, eine Einlesesperre an die NC zu geben.

Die **G-Funktionen** werden sofort ausgewertet und an die Anwendernahtstelle übergeben.

### NC-Prozessalarme

Bei Auslösung des Interrupts durch die NC (ist in jedem IPO-Takt möglich), wird ein Bit im Lokaldatum des OB 40 ("GP\_IRFromNCK") vom Grundprogramm nur dann gesetzt, wenn FB 1 Parameter "UserIR": = TRUE ist. Bei anderen Ereignissen (Prozessalarme durch die Peripherie), ist dieses Datum nicht gesetzt. Mit dieser Information kann in die zugehörige Interrupt-Routine im Anwenderprogramm verzweigt werden, um die notwendigen Aktionen einzuleiten.

Einer schneller auftragsgesteuerten Bearbeitung vom Anwenderprogramm für die Maschine, stehen ab dem Softwarestand 3.2 folgende Funktionen von der NC in der Interruptbearbeitung (OB 40 Programmteil) für das PLC-Anwenderprogramm zur Verfügung:

- Ausgewählte **Hilfsfunktionen**
- **Werkzeugwechsel-Funktion** bei der Option Werkzeugverwaltung
- **Position erreicht** bei den Positionierachsen, Teilungsachsen und Spindeln bei Aktivierung über PLC
- Satzübergabe zur FM (Funktion in Vorbereitung)

Die oben aufgeführten Funktionen müssen durch das Anwenderprogramm in dem OB 40 ausgewertet werden, um schnelle Reaktionen an der Maschine einzuleiten. Hier kann z.B. bei der Programmierung eines T-Befehls an einer Drehmaschine das Revolver-Schaltwerk aktiviert werden.

Weitere Einzelheiten zur Programmierung von Prozessalarmen (Zeitverzögerung, Unterbrechbarkeit, usw. sind in der entsprechenden SIMATIC-Dokumentation nachzulesen.

## Hilfsfunktionen

Generell gilt, dass schnelle oder quittierende Hilfsfunktionen unabhängig von einer Zuordnung mit oder ohne Interruptsteuerung bearbeitet werden.

Durch Grundprogrammparameter im FB 1 kann ausgewählt werden, welche Hilfsfunktionen (T, H, DL) vom Anwenderprogramm nur interruptgesteuert bearbeitet werden sollen.

Die nicht über Interrupt zugeordneten Funktionen werden wie bisher erst durch das zyklische Grundprogramm zur Verfügung gestellt. Hierbei stehen die Änderungssignale der Funktionen einen PLC-Zyklus an.

Wenn die Auswahl für die Hilfsfunktionsgruppen (T, H, DL) mit Interrupt-Steuerung getroffen ist, dann kann für die ausgewählten Funktionen nur noch eine Interruptbearbeitung durch das Anwenderprogramm erfolgen.

Für das Anwenderprogramm wird kanalorientiert ein Bit in dem Lokaldatum "GP\_AuxFunction" gesetzt (wenn "GP\_AuxFunction[1]" gesetzt ist, dann steht für den 1. Kanal eine Hilfsfunktion bereit).

Im zugehörigen Kanal-DB stehen Änderungssignal und Funktionswert für den Anwender zur Verfügung. Das Änderungssignal dieser interruptgesteuerten Funktion wird im zyklischen Grundprogrammteil nach Ablauf von mindestens einem vollem OB 1 Zyklus (max. ca. zwei OB 1 Zyklen) wieder auf Null zurückgesetzt.

## Werkzeugwechsel

Bei der Option Werkzeugverwaltung werden der Werkzeugwechsel-Befehl für Revolver und der Werkzeugwechsel in die Spindel durch einen Interrupt unterstützt. Hierzu wird das Lokaldatenbit "GP\_TM" im OB 40 gesetzt. Damit kann das PLC-Anwenderprogramm den DB der Werkzeugverwaltung (DB 72 bzw. DB 73) auf die Werkzeugwechselfunktion abprüfen und den Wechselvorgang anstoßen.

## Position erreicht

In der Bitstruktur "GP\_InPosition" der Lokaldaten des OB 40 ist maschinen- achsspezifisch (jedes Bit entspricht einer Achse/Spindel, z.B. GP\_InPosition[5] entspricht der 5. Achse) aufgebaut.

Wenn eine Funktion über FC 15 (Positionierachse), FC 16 (Teilungsachse) oder FC 18 (Spindelsteuerung) für eine Achse oder Spindel aktiviert wurde, kann das zugehörige "GP\_InPosition"-Bit eine unverzögerte Auswertung des Signals "InPos" der oben aufgeführten FCs erreicht werden. Hierdurch können z.B. Klemmungen für eine Teilungsachse unverzögert aktiviert werden.

## Deklaration

```
FUNCTION FC 3: VOID
//keine Parameter
```

## Aufrufbeispiel

Das Grundprogramm muss - zeitlich gesehen - **vor der** Bearbeitung weiterer alarmgesteuerter Anwenderprogramme durchlaufen werden. Deshalb ist es als erstes im OB 40 aufzurufen.

Das vorliegende Beispiel enthält die Standard-Deklarationen für den OB 40 und den Aufruf für das Grundprogramm.

```
ORGANIZATION_BLOCK OB 40
VAR_TEMP
    OB40_EV_CLASS :          BYTE;
    OB40_STRT_INF :          BYTE;
    OB40_PRIORITY :          BYTE;
    OB40_OB_NUMBR :          BYTE;
    OB40_RESERVED_1 :        BYTE;
    OB40_MDL_ID :            BYTE;
    OB40_MDL_ADDR :          INT;
    OB40_POINT_ADDR :        DWORD;
    OB40_DATE_TIME :         DATE_AND_TIME;

//Belegung durch das Grundprogramm
GP_IRFromNCK : BOOL;          //Interrupt durch NCK für Anwender
GP_TM : BOOL;                 //Werkzeugverwaltung
GP_InPosition : ARRAY [1..3] OF BOOL; //Achsenorientiert für Positionier.-,
//Teilungsachsen, Spindeln
GP_AuxFunction : ARRAY [1..10] OF BOOL; //Kanalorientiert für Hilfsfunktionen
GP_FMBlock : ARRAY [1..10] OF BOOL; //Kanalorientiert für Satzübergabe zur FM
// (in Vorbereitung)

//ab hier dürfen weitere Lokaldaten des Anwenders definiert werden
END_VAR
BEGIN
    CALL FC 3;
    //HIER ANWENDERPROGRAMM EINFUEGEN
END_ORGANIZATION_BLOCK
```

## 2.12.13 FC 7: TM\_REV Transfer-Baustein für Werkzeugwechsel mit Revolver

### Funktionsbeschreibung

Der Anwender ruft nach erfolgtem Wechsel eines Revolvers diesen Baustein FC TM\_REV auf. Hierzu ist in dem Parameter "ChgdRevNo" die Revolvernummer (entsprechend der Schnittstellen-Nummer im DB 73) mitzuteilen. Mit Aufruf dieses Bausteins wird das zugehörige "Schnittstelle aktiv"-Bit im Datenbaustein DB 73, Wort 0 vom FC 7 zurückgesetzt, nachdem der Parameter "Ready" = TRUE zurückgemeldet wird.

Der Baustein FC TM\_REV darf nur mit Parameter "Start" = TRUE gestartet werden, wenn für diesen Transfer eine Aktivierung der zugehörigen Schnittstelle (DB 73, Wort 0) durch die Werkzeugverwaltung vorliegt.

Wenn dieser Auftrag korrekt ausgeführt wurde, enthält der **Ausgangsparameter "Ready"** den Wert TRUE.

Damit muss der Anwender den **Parameter "Start"** = FALSE setzen bzw. den Baustein nicht mehr aufrufen.

Falls der **Parameter "Ready"** = FALSE ist, muss der Fehlercode im **Parameter "Error"** interpretiert werden.

Wenn der Fehlercode = 0 ist, dann ist dieser Auftrag im nächsten PLC-Zyklus zu wiederholen ("Start" bleibt TRUE). Es bedeutet, dass der Transferauftrag noch nicht beendet ist, siehe Beispiel FC 7 Aufruf und Impulsdiagramm.

Der Parameter "Start" benötigt keine Flanke für einen Folgeauftrag.



### Warnung

Ein Abbruch eines Transfers (z.B. durch ein externes Signal RESET) ist nicht zulässig. Der Parameter "Start" muss in jedem Fall so lange 1-Signal führen, bis die Parameter "Ready" bzw. "Error"  $\neq 0$  sind.

---

Bei Fehlercode  $\neq 0$  liegt eine Falschparametrierung vor.

---

### Hinweis

Weitere Informationen zur Werkzeugverwaltung (auch in Bezug zur PLC) sind in der Funktionsbeschreibung Werkzeugverwaltung enthalten. Weiterhin stehen noch PI-Dienste für die Werkzeugverwaltung über den FB 4, FC 8 und FC 22 zur Verfügung.

---



## Revolverschalten von Hand

Wird über eine Handbedienung der Revolver verdreht, so ist es notwendig die Werkzeugverwaltung zu informieren. Über die Funktion asynchroner Transfer des FC 8 müssen die geänderten Positionen des Revolvers mitgeteilt werden. Dies darf nur einmalig bei der ersten Handverdrehung in dieser Sequenz erfolgen. Folgende Parametrierung des asynchronen Transfers ist dann erforderlich über FC 8:

TaskIdent = 4

TaskIdentNo = Kanal

NewToolMag = Magazinnummer des Revolvers

NewToolLoc = Ursprünglicher Platz des Werkzeugs

OldToolMag = Magazinr. Zwischenspeicher (Spindel) = 9998

OldToolLoc = Zwischenspeichernummer der Spindel

Status = 1

Mit dieser Maßnahme wird auch erreicht, dass der gleiche T Befehl an die Werkzeugverwaltungs-Nahtstelle wiederholt wird, wenn das vorhergehende T noch einmal programmiert wird.

## Deklaration der Funktion

### AWL-Darstellung

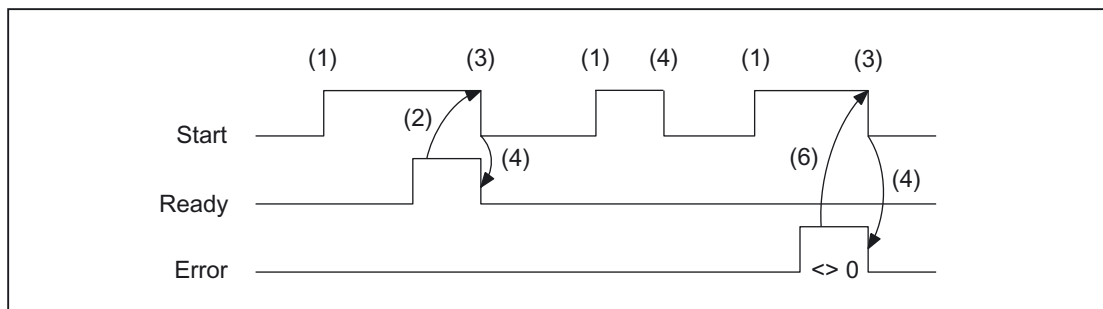
```
FUNCTION FC 7 :          VOID
//NAME :TM_REV
VAR_INPUT
    Start :              BOOL;
    ChgdRevNo :          BYTE;
END_VAR
VAR_OUTPUT
    Ready :              BOOL;
    Error :              INT;
END_VAR
BEGIN
END_FUNCTION
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion TM\_REV.

Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		1 = Transfer wird gestartet.
ChgdRevNo	E	BYTE	1..	Nummer der Revolver-Schnittstelle
Ready	A	BOOL		1 = Transfer abgeschlossen
Error	A	INT	0..3	Fehler-Rückmeldung 0 : kein Fehler aufgetreten 1: kein Revolver vorhanden 2: unzulässige Revolvernummer in Parameter "ChgdRevNo" 3: unzulässiger Auftrag ("Signal "Schnittstelle (SS) aktiv" des angewählten Revolvers = "FALSE")

### Impulsdiagramm



- (1) Funktionsanstoß durch positive Flanke
- (2) Positive Quittung: WZV Transfer wurde ausgeführt
- (3) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC
- (5) Dieser Signalverlauf ist unzulässig. Der Auftrag ist generell zu beenden, da die neuen Werkzeug-Positionen der Werkzeugverwaltung im NCK mitzuteilen sind.
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter Error

## Aufrufbeispiel

```
CALL FC 7(                                     //Werkzeugverwaltungs Transfer Baustein für Revolver
Start :=                                       m 20.5,           //Start := "1 " => Anstoß des Transfers
ChgdRevNo :=                                  DB61.DBB 1,
Ready :=                                       m 20.6,
Error :=                                       DB61.DBW 12);
u m 20.6;                                       //Ready abfragen
r m 20.5;                                       //Start zurücksetzen
spb m001;                                       //Springe, wenn alles in Ordnung
l db61.dbw 12;                                   //Fehler-Information
ow w#16#0;                                       //Fehler auswerten
spn fehl;                                       //Springe auf Fehlerbehandlung, wenn <> 0
m001 :                                         //Beginn des weiteren Programms
fehl :
r m 20.5;                                       //Start zurücksetzen, wenn Fehler vorliegt
```

## 2.12.14 FC 8: TM\_TRANS Transfer-Baustein für Werkzeugverwaltung

### Funktionsbeschreibung

Der Anwender ruft diesen Baustein FC TM\_TRANS bei Positionsänderungen der Werkzeuge oder bei Statusänderungen des Transfervorgangs auf. Mit dem Parameter "TaskIdent" wird für den Baustein FC 8 der Transferauftrag an der Werkzeugverwaltungs-Schnittstelle angegeben:

1. für Be-/Entladestellen,
2. für Spindel-Wechselstellen,
3. für Revolver-Wechselstellen als Transferkennung,
4. Asynchroner Transfer,
5. Asynchroner Transfer mit Platzreservierung.

Die Schnittstellen-Nummer wird im Parameter "TaskIdentNo" mitgeteilt.

Beispiel für Beladestelle 5:

Parameter "TaskIdent":= 1 und "TaskIdentNo":= 5.

Weiterhin werden für diesen Transfer die **aktuellen** Werkzeugpositionen und Statusinformationen, siehe Status-Auflistung des Parameters "Status" im nachfolgenden Text, übermittelt.

---

### Hinweis

Der FC 8 teilt dem NCK die aktuellen Positionen vom alten Werkzeug mit.

Dem NCK ist bekannt, wo sich das alte und das neue Werkzeug bis zur Positionsänderung befunden haben.

---

Bei einem Transfer ohne ein sogenanntes "altes Werkzeug" (z.B. beim Beladen), wird den Parametern "OldToolMag", "OldToolLoc" der Wert 0 zugeteilt.

Der Transfer-Baustein darf nur mit den Parameter "Start" = TRUE gestartet werden, wenn für diesen Transfer eine Aktivierung der zugehörigen Schnittstelle (DB 71, DB 72, DB 73 im Wort 0) durch die Werkzeugverwaltung vorliegt.

Wenn dieser Auftrag korrekt ausgeführt wurde, enthält der **Ausgangsparameter "Ready"** den Wert TRUE.

Damit muss der Anwender den **Parameter "Start"** = FALSE setzen bzw. den Baustein nicht mehr aufrufen.

Falls der **Parameter "Ready"** = FALSE ist, muss der Fehlercode im **Parameter "Error"** interpretiert werden, siehe Beispiel FC 8 Aufruf und Impulsdiagramm.

Wenn der Fehlercode = 0 ist, dann ist dieser Auftrag im nächsten PLC-Zyklus zu wiederholen ("Start" bleibt TRUE). Es bedeutet, dass der Transferauftrag noch nicht beendet ist.

Wenn der **Parameter "Status"** einen Wert kleiner 100 durch den Anwender erhält, wird die zugehörige Schnittstelle im Datenbaustein DB 71 bzw. DB 72 oder DB 73, Wort 0 deaktiviert (Vorgang beendet). Das entsprechende Bit für die Schnittstelle wird auf 0 gesetzt durch den FC 8.

Der Parameter "Start" benötigt keine Flanke für einen Folgeauftrag. Dies bedeutet, dass sofort bei Erhalten von "Ready" = TRUE neue Parameter zugeordnet werden können mit "Start" = TRUE.

### **Asynchroner Transfer**

Für eine eigenständige Mitteilung der PLC von den Positionsänderungen eines Werkzeugs zur Werkzeugverwaltung (z.B. Spannungsunterbrechung bei einem aktiven Kommando oder selbständige Positionsveränderungen durch PLC) wird dieser FC 8 Transfer-Baustein mit der "TaskIdent": = 4 oder 5 aufgerufen. Zu diesem Aufruf muss keine Schnittstellen-Aktivierung durch die Werkzeugverwaltung vorliegen.

Bei dem Parameter "TaskIdent" = 5 wird zusätzlich zur Positionsveränderung noch eine Platzreservierung von der Werkzeugverwaltung vorgenommen. Diese Platzreservierung erfolgt aber nur, wenn das Werkzeug aus einem realen Magazin in einen Zwischenspeicher transportiert wurde.

Im Parameter "TaskIdentNo" ist ein zugehöriger NC-Kanal zu parametrieren.

In den Parametern "OldToolMag", "OldToolLoc" wird die bisherige Position des Werkzeugs angegeben und in den Parametern "NewToolMag", "NewToolLoc" wird die aktuelle Position dieses Werkzeugs angegeben. Hierbei ist der Status = 1 anzugeben.

Bei Status 5 verbleibt das angegebene Werkzeug auf dem Platz "OldToolMag", "OldToolLoc". Dieser Platz muss ein Zwischenspeicher sein (z.B. Spindel). In den Parametern "NewToolMag", "NewToolLoc" ist das reale Magazin mit Platz anzugeben, wobei sich der Platz an der Position des Zwischenspeichers befindet. Dieses Verfahren ist immer dann anzuwenden, wenn der Werkzeugverwaltung mitgeteilt werden soll, wo sich ein bestimmter Platz des Magazins befindet. Dieses Verfahren dient zum Abgleich bei Suchstrategien.

---

#### **Hinweis**

Ein Abbruch eines Transfers (z.B. durch ein externes Signal RESET) ist nicht zulässig. Der Parameter "Start" muss in jedem Fall so lange 1-Signal führen, bis die Parameter "Ready" bzw. "Error"  $\neq$  0 sind.

---

Bei Fehlercode  $\neq$  0 liegt eine Falschparametrierung vor.

---

#### **Hinweis**

Weitere Informationen zur Werkzeugverwaltung (auch in Bezug zur PLC) sind in der Funktionsbeschreibung Werkzeugverwaltung enthalten. Weiterhin stehen noch PI-Dienste für die Werkzeugverwaltung über den FB 4, FC 7 und FC 22 zur Verfügung.

---

## Deklaration der Funktion

### AWL-Darstellung

```

FUNCTION FC 8 : VOID
//NAME :TM_TRANS
VAR_INPUT
    Start :                BOOL;
    TaskIdent :            BYTE;
    TaskIdentNo :         BYTE;
    NewToolMag :          INT;
    NewToolLoc :          INT;
    OldToolMag :          INT;
    OldToolLoc :          INT;
    Status :              INT;
END_VAR
VAR_OUTPUT
    Ready :               BOOL;
    Error :               INT;
END_VAR
BEGIN
END_FUNCTION
  
```

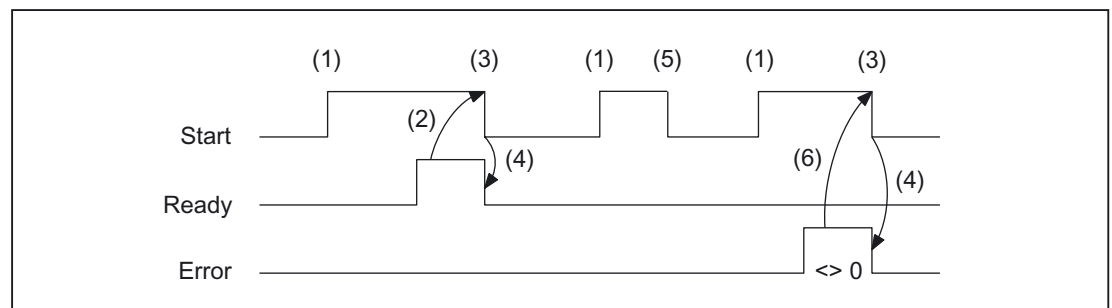
## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion TM\_TRANS:

Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		1 = Transfer wird gestartet.
TaskIdent	E	BYTE	1..5	Kennung der Schnittstelle bzw. des Auftrags 1: Be-/Entladestelle 2: Spindel-Wechselstelle 3: Revolver-Wechselstelle 4: asynchroner Transfer 5: asynchroner Transfer mit Platzreservierung
TaskIdentNo	E	BYTE	1..	Nummer der zugehörigen Schnittstelle bzw. Kanalnummer. Im oberen Nibble kann beim asyn. Transfer die Schnittstellennr. angegeben werden (z.B. B#16#12, 1. Schnittstelle, 2. Kanal).
NewToolMag	E	INT	1, 0..	aktuelle Magazinnummer des neuen Werkzeugs -1: Werkzeug verbleibt auf seinem bisherigen Platz. NewToolLoc = beliebiger Wert. Nur bei TaskIdent = 2 zulässig
NewToolLoc	E	INT	0 ...max. Platznummer	aktuelle Platznummer des neuen Werkzeugs

Signal	Art	Typ	Wertebereich	Bemerkung
OldToolMag	E	INT	-1, 0..	aktuelle Magazinnummer des auszuwechselnden Werkzeugs -1: Werkzeug verbleibt auf seinem bisherigen Platz. OldToolLoc = beliebiger Wert. Nur bei TaskIdent = 2 zulässig.
OldToolLoc	E	INT	max. Platznummer	aktuelle Platznummer des auszuwechselnden Werkzeugs
Status	E	INT	1...7, 103...105	Status-Information des Transfervorgangs
Ready	A	BOOL		1= Transfer abgeschlossen
Error	A	INT	0..65535	Fehler-Rückmeldung 0: kein Fehler aufgetreten 1: unbekannte "TaskIdent" 2: unbekannte "TaskIdentNo" 3: unzulässiger Auftrag ("Signal "Schnittstelle (SS) aktiv" des angewählten Revolvers = "FALSE") andere Werte: Die Zahl entspricht der Fehlermeldung der WZV im NCK, die durch diesen Transfer verursacht wurde.

### Impulsdiagramm



- (1) Funktionsanstoß durch positive Flanke
- (2) Positive Quittung: WZV Transfer wurde ausgeführt
- (3) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC
- (5) Dieser Signalverlauf ist unzulässig. Der Auftrag ist generell zu beenden, da die neuen Werkzeug-Positionen der Werkzeugverwaltung im NCK mitzuteilen sind.
- (6) Negative Quittung: Fehler aufgetreten, Fehler-Code im Ausgangs-Parameter Error

## Status-Auflistung

### Status = 1:

**Der WZV-Auftrag ist beendet worden** (Be-/Ent-/Umladen, Wechsel vorbereiten, Wechseln).

Die Parameter "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc" des FC 8 Bausteins sind auf die tatsächlichen Positionen der beteiligten Werkzeuge zu parametrieren. Außer beim Wechsel vorbereiten sind dies die in der Regel die angegebenen Werkzeug-Zielpositionen der zugehörigen WZV-Schnittstelle, siehe auch "Erläuterungen der Formalparameter".

1. Beim Be-/Ent-/Umladen ist das Werkzeug auf der geforderten Zieladresse angekommen. Wenn das Bit in der Nahtstelle im DB 71.DBX (n+0).3 "Positionieren zur Beladestelle" gesetzt ist, dann darf für den Abschluss der Funktion nicht der Status 1 verwendet werden. Für einen korrekten Abschluss ist der Status 5 zu verwenden.
2. Beim "Wechsel vorbereiten" ist die Bereitstellung des neuen Werkzeuges erfolgt. Dieses kann z.B. auf einem Zwischenspeicher (Greifer) abgelegt sein. Gegebenenfalls wurde auch das Ziel (Magazin, Platz) des alten Werkzeuges zur Wechsellage positioniert, nachdem das neue Werkzeug in einem Zwischenspeicher abgelegt wurde. Das alte Werkzeug verbleibt aber noch in der Spindel. Somit ist die Vorbereitung zum Wechseln abgeschlossen. Nach dieser Quittierung kann das Kommando "Wechseln" empfangen werden. Die Positionen in den Parametern "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc" entsprechen den aktuellen Positionen der Werkzeuge.
3. Beim "Wechseln" (in Spindel bzw. beim Revolver) sind die in der Schnittstelle angesprochenen Werkzeuge auf den geforderten Zieladressen angekommen. Der Vorgang des Werkzeugwechsels ist damit abgeschlossen.

### Status = 2:

**Das "neue" Werkzeug kann nicht bereitgestellt werden.**

Dieser Status ist nur bei dem Befehl "Wechsel vorbereiten" zulässig. Wenn dieser Status angewendet wird, soll von der PLC ein Wechsel mit dem vorgeschlagenen Werkzeug verhindert werden. Durch die WZV im NCK erfolgt das Sperren dieses vorgeschlagenen (neuen) Werkzeuges. Anschließendes erfolgt ein neues Kommando der Werkzeugverwaltung mit einem Duplo-Werkzeug. Die Positionen in den Parametern "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc" entsprechen den ursprünglichen Positionen der Werkzeuge.

### Status = 3:

**Ein Fehler ist aufgetreten.**

Es darf keine Änderung der Werkzeugpositionen erfolgt sein. Wenn sich die Magazinpositionen der Werkzeuge zwischenzeitlich verändert hatten, ist dieses z.B. vorher mit dem Status = 105 über den FC 8 Transfer-Baustein mitzuteilen. Nur dann werden die Positionen von der Werkzeugverwaltung berücksichtigt.

### Status = 4:

**Das "alte" Werkzeug sollte besser auf der in den Parametern "OldToolMag", "OldToolLoc" angegebenen Magazinposition untergebracht werden.**

Dieser Status ist nur bei der Werkzeugwechsel-Vorbereitung (Wechsel in die Spindel) zulässig. Nachdem dieser Status an die Werkzeugverwaltung im NCK gegeben wurde, versucht die Werkzeugverwaltung beim nachfolgenden Kommando die angegebene Magazin-Position zu berücksichtigen. Dieses erfolgt aber nur dann, wenn diese Position frei ist. Die Parameter "NewToolMag" und "NewToolLoc" werden nicht berücksichtigt.



**Status = 5:**  
**Der Vorgang ist beendet worden.**

Das "neue" Werkzeug ist **an** der in den Parametern "NewToolMag", "NewToolLoc" angegebenen Position. Hierbei befindet sich das angegebene Werkzeug nicht wirklich in dieser Position, sondern ist weiterhin im gleichen Magazinplatz. Dieser Magazinplatz wurde jedoch zu dieser Position (z.B. Wechselstelle) hin positioniert. Dieser Status darf nur bei Revolver, Ketten- und Scheibenmagazinen verwendet werden. Der Status dient der Werkzeugverwaltung zum Abgleich der aktuellen Position eines Magazins und zur Verbesserung der Suchstrategie für nachfolgende Befehle. Dieser Status ist nur beim Beladen, Entladen, Umladen und bei der Wechsel-Vorbereitung erlaubt.

Die Parameter "OldToolMag" und "OldToolLoc" müssen mit den Daten eines Zwischenspeichers parametrisiert sein.

- **Beladen, Umladen:**

Beim Beladen oder Umladen ist im NCK schon ein Platz für das Werkzeug reserviert worden. Im Anschluss muss der Maschinen-Bediener das Werkzeug auf dem Zielplatz einsetzen. Achtung: Nach einem erneuten Einschalten der Steuerung ist die Platzreservierung aufgehoben.

- **Wechsel-Vorbereitung:**

Die noch ausstehenden Werkzeugbewegungen erfolgen erst mit dem Werkzeug-Wechsel.

- **Positionieren zur Beladestelle:**

Wenn das Bit in der Nahtstelle im DB 71.DBX (n+0).3 "Positionieren zur Beladestelle" gesetzt ist, dann darf für den Abschluss der Funktion nur der Status 5 (nicht Status 1) verwendet werden.

**Status = 6:**  
**Der WZV-Auftrag ist beendet worden.**

Dieser Status hat die gleiche Funktion wie der Status 1, aber zusätzlich wird eine Reservierung des Quellplatzes vorgenommen. Dieser Status ist nur beim Umladen erlaubt. Das Kommando wird beendet und der Quellplatz des Werkzeugs wird reserviert, falls der Zielplatz in einem Zwischenspeichermagazin liegt.

**Status = 7:**  
**Wiederholung des "Werkzeug Vorbereiten" Befehls anstossen.**

Dieser Status ist nur bei dem "Wechsel vorbereiten"-Befehl zulässig. Der Status soll angewendet werden, wenn das "neue" Werkzeug seine Position verändert hat (z. B. über einen asynchronen Befehl des "neuen" Werkzeugs). Nach "Ready = 1" vom FC 8 erfolgt eine automatische Wiederholung des "Wechsel vorbereiten" mit dem gleichen Werkzeug. Für die automatische Wiederholung wird eine neue Werkzeugsuche durchgeführt. Die Positionen in den Parametern "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc" müssen den ursprünglichen Positionen der Werkzeuge entsprechen.

**Status = 103:**  
**Das "neue" Werkzeug kann eingesetzt werden.**

Dieser Status ist nur bei der Wechsel-Vorbereitung zulässig, wenn die PLC das neue Werkzeug ablehnen darf (z.B. bei MD20310 \$MC\_TOOL\_MANAGEMENT\_MASK, Bit 4=1 für Möglichkeit, geänderte Parameter vom PLC noch einmal anzufordern). Die Positionen der Werkzeuge sind unverändert geblieben. Dieser Status ist damit notwendig, wenn der Vorlauf im NCK ohne einen unnötigen Stopp der Bearbeitung fortgesetzt werden soll.

**Literatur:**  
/FBW/ Funktionsbeschreibung Werkzeugverwaltung.

**Status = 104:**

**Das "neue" Werkzeug ist an der in den Parametern "NewToolMag", "NewToolLoc" angegebenen Position.**

Dieser Status ist nur zulässig, wenn das Werkzeug sich noch im Magazin auf dem gleichen Platz befindet. Das "alte" Werkzeug ist auf der in den Parametern "OldToolMag", "OldToolLoc" angegebenen Position (Zwischenspeicher). Hierbei befindet sich das neue Werkzeug nicht wirklich in dieser Position, sondern ist weiterhin im gleichen Magazinplatz. Dieser Magazinplatz wurde jedoch zu dieser Position (z.B. Wechselstelle) hin positioniert. Dieser Status darf nur bei Revolver, Ketten- und Scheibenmagazinen beim "Werkzeug-Wechsel Vorbereiten" verwendet werden. Der Status dient der Werkzeugverwaltung zum Abgleich der aktuellen Position eines Magazins, zur Verbesserung der Suchstrategie für nachfolgende Befehle.

**Status = 105:**

**Der angegebene Zwischenplatz ist erreicht von allen beteiligten Werkzeugen** (Standardfall, wenn der Vorgang noch nicht beendet ist).

Die Werkzeuge sind in den angegebenen Werkzeug-Positionen (Parameter "NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc").

## Status-Definition

Generell gilt für den Quittungsstatus, dass die Statusinformationen 1 bis 7 zu einer Beendigung des Kommandos führen. Wenn eine dieser Statusinformationen dem FC 8 mitgeteilt wird, wird das "Schnittstelle Aktiv-Bit" der im FC 8 angegebenen Schnittstelle auf "0" zurückgesetzt (siehe auch Nahtstellen Listen DB71 bis DB73). Damit ist der Vorgang abgeschlossen. Anders verhält es sich bei den Statusinformationen 103 bis 105. Bei Übergabe einer dieser Statusinformationen zum FC 8 bleibt das "Schnittstelle Aktiv-Bit" dieser Schnittstelle auf "1". Eine weitere Bearbeitung ist notwendig durch das Anwenderprogramm in der PLC (z.B. Fortführen der Magazinpositionierung). Diese Statusinformation dient in der Regel zum Übermitteln von veränderten Positionen von einem oder beiden Werkzeugen, ohne dass der Vorgang abgeschlossen ist.

## Aufrufbeispiel

```
CALL FC 8(           //Werkzeugverwaltungs Transfer Baustein
  Start :=          m 20.5,           //Start := "1 " => Anstoß des Transfers
  TaskIdent :=      DB61.DBB 0,
  TaskIdentNo :=    DB61.DBB 1,
  NewToolMag :=     DB61.DBW 2,       //aktuelle Position neues Werkzeug
  NewToolLoc :=     DB61.DBW 4,
  OldToolMag :=     DB61.DBW 6,       //aktuelle Position altes Werkzeug
  OldToolLoc :=     DB61.DBW 8,
  Status :=        DB61.DBW 10,      //Status
  Ready :=         m 20.6,
  Error :=         DB61.DBW 12);

u m 20.6;           //Ready abfragen
r m 20.5;           //Start zurücksetzen
spb m001;          //springe wenn alles in Ordnung
l DB61.dbw 12;     //Fehler Information
ow w#16#0;        //Fehler auswerten
spn fehl;         //Springe auf Fehlerbehandlung

m001 :             //normaler Zweig

fehl :             //Fehlerbehandlung
r m 20.5 :        //Start zurücksetzen
```

## 2.12.15 FC 9: ASUP Start von asynchronen Unterprogrammen

### Funktionsbeschreibung

Mit dem FC ASUP können beliebige Funktionen in der NC ausgelöst werden. Voraussetzung dafür, dass ein ASUP von der PLC gestartet werden kann, ist dessen Anwahl und Parametrierung durch ein NC-Programm oder durch den FB 4 (PI-Dienst ASUP). Hierbei muss Kanal- und Interruptnummer übereinstimmen zu den Parametern im FC 9. Ein derart vorbereitetes ASUP kann von der PLC zu einem beliebigen Zeitpunkt gestartet werden. Das in dem betreffenden Kanal laufende NC-Programm wird durch das ASUP unterbrochen. In einem Kanal kann dabei zu einem Zeitpunkt nur ein ASUP gestartet werden. Sollte in **einem** PLC Zyklus für zwei FC 9 der Start-Parameter auf logisch 1 gesetzt werden, werden die ASUP in Aufrufreihenfolge gestartet.

Der Start-Parameter muss vom Anwender auf logisch 0 gesetzt werden, wenn das ASUP beendet (Done) oder ein Fehler aufgetreten ist.

Zur Auftragsbearbeitung benötigt jeder FC ASUP einen eigenen WORD-Parameter "Ref" aus dem globalen Anwender-Speicherbereich. Dieser wird intern verwendet und darf vom Anwender nicht verändert werden. Der Parameter "**Ref**" wird im ersten OB 1-Zyklus mit dem Wert 0 initialisiert, aus diesem Grund **muss jeder FC 9 absolut aufgerufen** werden. Alternativ kann der Parameter "Ref" mit dem Wert 0 initialisiert werden durch den Anwender im Anlauf. Damit sind auch bedingte Aufrufe möglich. Ein bedingter Aufruf muss bei Aktivierung durch den Parameter "Start" = 1 solange erfolgen, bis der Parameter "Done" einen Zustandswechsel von 1 nach 0 durchführt.

---

### Hinweis

Der FB 4 Aufruf muss beendet sein bevor der FC 9 gestartet werden darf. Der FC 9 kann nicht gestartet werden, wenn "NOT-HALT" gesetzt ist. Der Start des FC 9 darf auch nicht erfolgen, wenn der Kanal-RESET aktiv ist.

---

## Deklaration der Funktion

```

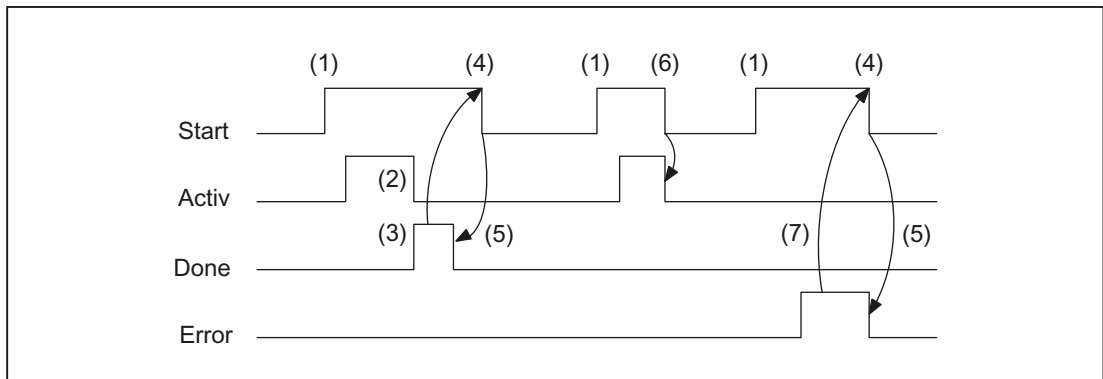
FUNCTION FC 9 : VOID
//NAME :ASUP
VAR_INPUT
    Start :    BOOL;
    ChanNo :  INT;
    IntNo  :   INT;
END_VAR
VAR_OUTPUT
    Activ :    BOOL;
    Done  :    BOOL;
    Error  :    BOOL;
    StartErr :  BOOL;
    :
END_VAR
VAR_IN_OUT
    Ref :      WORD;
END_VAR
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion ASUP.

Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		
ChanNo	E	INT	1 - 10	Nr. des NC-Kanals
IntNo	E	INT	1 - 8	Interrupt-Nr.
Activ	A	BOOL		1 = aktiv
Done	A	BOOL		1 = ASUP beendet
Error	A	BOOL		1 = Interrupt ist ausgeschaltet
StartErr	A	BOOL		1 = Interruptnummer nicht vergeben oder gelöscht
Ref	E/A	WORD	globale Variable (MW,DBW,..)	1 Wort je FC 9 (für interne Verwendung)

## Impulsdiagramm



- (1) Funktionsanstoß
- (2) ASUP aktiv
- (3) Positive Quittung: ASUP beendet
- (4) Zurücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (5) Signalwechsel durch FC
- (6) Nicht zulässig. Wird Funktionsanstoß vor Erhalt der Quittung rückgesetzt, werden die Ausgangssignale nicht aktualisiert, ohne Einfluss auf Ablauf der angestoßenen Funktion
- (7) Negative Quittung: Fehler aufgetreten

## Aufrufbeispiel

```
CALL FC 9(           //Start eines asynchronen Unterprogramms
              //im Kanal 1 Interruptnummer 1
              Start :=      E 45.7,
              ChanNo :=     1,
              IntNo :=      1,
              Activ :=      M 204.0,
              Done :=       M204.1,
              Error :=      M 204.4,
              StartErr :=   M 204.5,
              Ref :=        MW 200);
```

## 2.12.16 FC 10: AL\_MSG Fehler- und Betriebsmeldungen

### Funktionsbeschreibung

Mit dem FC AL\_MSG werden die in DB 2 eingetragenen Signale ausgewertet und als kommende und gehende Fehler- und Betriebsmeldungen auf dem MMC zur Anzeige gebracht.

Die kommenden Signale (positive Flanke) werden sowohl bei Fehler- als auch bei Betriebsmeldungen sofort zur Anzeige gebracht.

Gehende Signale (negative Flanke) werden nur bei Betriebsmeldungen sofort gelöscht; bei Fehlermeldungen werden erst mit dem Parameter "Quit" die nicht mehr anstehenden Meldungen gelöscht, d.h. Fehleranzeigen bleiben - auch wenn die Signale nicht mehr anstehen - auf dem MMC solange erhalten, bis sie vom Anwender quittiert wurden.

Über den Parameter "ToUserIF" kann die Übertragung der Sammelsignale für Vorschub-, Einlese- und NC-Start-Sperre sowie Vorschub-Halt an vorhandene Achs-, Spindel- und Kanal- Nahtstellen veranlasst werden. Die Sammelsignale werden unabhängig von der Quittierung eines Alarms direkt aus den Zustandsinformationen des DB 2 in die Anwendernahtstelle transferiert.

1. Wenn der Parameter "ToUserIF" := FALSE ist, erfolgt keine Übertragung der Signale an die Anwendernahtstelle. Der Anwender muss in diesem Fall durch sein PLC-Programm dafür sorgen, dass diese Signale in der Nahtstelle beeinflusst werden.
2. Wenn der Parameter "ToUserIF" := TRUE ist, erfolgt eine Übertragung aller oben aufgeführten Signale an die Anwendernahtstelle als jeweils ein Sammelsignal. Das PLC-Programm des Anwenders kann folglich die oben aufgeführten Signale nur über den DB 2 beeinflussen in Verbindung mit einer Meldungs- oder Alarm-Ausgabe. Es erfolgt ein Überschreiben der jeweiligen Information in der Anwendernahtstelle.

Alternativ zu dem unter Punkt 2 beschriebenen Verhalten kann der Anwender eine Beeinflussung des Sperr- und Halt-Signals ohne Meldungsausgabe vornehmen, indem er nach dem Aufruf des FC AL\_MSG die Nahtstellensignale mit einem Sperr- oder Halt-Signalzustand beeinflusst.

Folgende Programmsequenz soll das verdeutlichen:

```
CALL FC 10(  
    ToUserIF    TRUE,  
    :=  
    Quit :=     e 6.1);  
  
u m 50.0;           //Vorschub Sperre für Kanal 1  
auf db 21;  
s dbx 6.0;         //Setzen der Sperrbedingung, Rücksetzen erfolgt über  
                   den  
                   //FC AL_MSG, wenn M 50.0 das Signal "0" führt.
```

Die Fehler- und Betriebsmeldungen werden vom Anwender im Datenbaustein DB 2 versorgt, damit auch eine Meldung auch am MMC angezeigt werden kann.

**Hinweis**

Im DB 2 muss ein "1-Signal" für mehrere OB 1 Zyklen anstehen, damit eine Meldung auch an der Bedienoberfläche angezeigt werden kann.

---

**Deklaration der Funktion**

**AWL-Darstellung**

```

FUNCTION FC 10:          VOID
  //NAME:                AL_MSG
VAR_INPUT
  ToUserIF :            BOOL;
  Quit :              BOOL;
END_VAR
END_FUNCTION
  
```

**Erläuterung der Formalparameter**

Die folgende Tabelle zeigt alle Formalparameter der Funktion AL\_MSG.

Signal	Art	Typ	Wertebereich	Bemerkung
ToUserIF	E	BOOL		1 = Übertragung der Signale an Anwenderschnittstelle je Zyklus
Quit	E	BOOL		1 = Quittierung von Fehlermeldungen

**Aufrufbeispiel**

```

CALL FC 10(
  ToUserIF := TRUE, //Fehler- und Betriebsmeldungen
  //Signale aus DB 2 werden an
  //Nahtstelle übertragen
  Quit := E6.1); //Quittierung der Fehlermeldung wird über
  //Eingang E6.1 durchgeführt
  
```



## 2.12.17 FC 12: AUXFU Aufrufchnittstelle für Anwender bei Hilfsfunktionen

### Funktionsbeschreibung

Der FC AUXFU wird generell ereignisgesteuert im Grundprogramm aufgerufen, wenn neue Hilfsfunktionen für den im Eingangs-Parameter übergebenen Kanal bereitstehen. Dieser FC kann durch den PLC-Anwender für seine Hilfsfunktion-Bearbeitung mit Programm-Anweisungen erweitert werden, um das zyklische Polling der Kanal-DBs bei Hilfsfunktionen zu vermeiden. Über diesen Mechanismus wird eine auftragsgesteuerte Bearbeitung der Hilfsfunktionen ermöglicht. Dieser FC wird als leerer Baustein in kompilierter Form im Grundprogramm geliefert. Hierbei wird der Parameter "Chan" vom Grundprogramm mit der Kanalnummer versorgt. Somit kennt der PLC-Anwender den Kanal in dem neue Hilfsfunktionen bereitstehen. Über die Änderungssignale der Hilfsfunktionen in diesem Kanal können die neuen Hilfsfunktionen ermittelt werden.

### Deklaration der Funktion

```
FUNCTION FC 12: VOID //Ereignissteuerung der Hilfsfunktionen
VAR_INPUT
    Chan : BYTE;
END_VAR
BEGIN
    BE;
END_FUNCTION
```

### Erläuterungen der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion AUXFU:

Signal	Art	Typ	Wertebereich	Bemerkung
Chan	E	BYTE	0 bis 9	Nr. des NC-Kanals -1

### Beispiel

```
FUNCTION FC 12: VOID //Ereignissteuerung der Hilfsfunktionen
VAR_INPUT
    Chan : BYTE; //Parameter wird vom Grundprogramm versorgt
END_VAR
VAR_TEMP
    ChanDB: INT;
END_VAR
BEGIN
    L Chan; //KanalIndex-Nr, (0,1,2,...)
    + 21; //Kanal DB Offset
```

```
T ChanDB; //Kanal DB-Nr. speichern
AUF DB[ChanDB]; //Kanal-DB wird indirekt geöffnet
//Jetzt werden die Änderungssignale der Hilfsfunktionen abgefragt, usw.
    BE;
END_FUNCTION
```

## 2.12.18 FC 13: BHGDisp Displaysteuerung für Bedienhandgerät

### Funktionsbeschreibung

Dieser Baustein übernimmt die Displaysteuerung des Bedienhandgerätes (BHG). Die Information, die auf dem Display erscheinen soll, ist in dem Stringdatum ChrArray mit 32 Zeichen abgelegt. Hierzu ist beim Erstellen des Datenbausteins für diesen String eine Festtext-Zuweisung von 32 Zeichen notwendig. Variable Anteile innerhalb dieses Strings können mit dem optionalen Zahlen-Wandler eingefügt werden. Für den Zahlenwandler ist der Parameter "Convert" auf TRUE zu setzen. Die Variable, die angezeigt werden soll, ist über den Pointer Addr referenziert. Im Parameter "DataType" ist die Formatbeschreibung dieses Parameters enthalten (siehe Parameter-Tabelle). Die Anzahl der Bytes der Variablen ist an die Formatbeschreibung gekoppelt. Die rechtsbündige Adresse innerhalb des Strings wird durch den Parameter "StringAddr" angegeben. Die Anzahl der geschriebenen Zeichen ist aus der Parametertabelle ersichtlich. Durch den Parameter "Row" = 0 kann die Displayausgabe unterdrückt werden (z.B. wenn mehrere Variablen in einem oder mehreren PLC-Zyklen im String eingetragen werden sollen, ohne dass eine Displayausgabe erfolgt).

### Signale

Von den Ausgangssignalen des BHG wird das Byte 1 und die Zeichenvorgaben durch den Baustein beschaltet. Diese dürfen nicht durch das PLC-Anwenderprogramm beschrieben werden.

### Zusätzliche Parameter

Im Anlauf-OB 100 sind am FB 1, DB 7 die Pointer-Parameter für die Ein- und Ausgangsdaten des Bedienhandgerätes zu parametrieren. Der Parameter "BHGIn" entspricht den Eingangsdaten der PLC vom Bedienhandgerät (Empfangsdaten der PLC). Der Parameter "BHGOut" entspricht den Ausgangsdaten der PLC zum Bedienhandgerät (Sendedaten der PLC). Diese beiden Pointer sind auf den jeweiligen Datenbereichs-Anfang einzustellen, der auch bei MPI-Kopplung im SDB210 parametrier ist.

---

### Hinweis

Wenn der Zahlenwandler zur Informationsanzeige genutzt wird, dann ist es zur Verringerung der PLC-Zykluszeit zweckmäßig, die Wandlung nicht in jedem PLC-Zyklus durchzuführen. Es empfiehlt sich hierzu das Eingangssignal vom BHG zur PLC "Quittung Ziffernanzeige" (EB m+5.7) für den Parameter "Convert" zu verwenden.

Damit ist sichergestellt, dass die aktuellste Zahleninformation zur Anzeige kommt.

---

## Deklaration der Funktion

### AWL-Darstellung

```
DATA_BLOCK "strdat"           //In Symboldatei ist die Datenbaustein-Nummer
                               definiert

    STRUCT
    disp :      STRING [32]:= 'Zeile 1 Zeile 2 ';      //32 Zeichen sind definiert
    END_STRUCT;

BEGIN
END_DATA_BLOCK

FUNCTION FC 13: VOID
VAR INPUT
    Row :      BYTE;           //Displayzeile (siehe Tabelle)
    ChrArray : STRING;         //mindestens String[32] übergeben
    Convert :  BOOL;          //Zahlenwandlung aktivieren
    Addr :    POINTER;        //Zeigt auf Variable, die konvertiert wird
    DataType : BYTE;          //Datentyp der Variablen
    StringAddr INT;           //rechtsbündige Stringadresse (1...32)
    :
    Digits :  BYTE;           //Anzahl Nachkommastellen (1...3)
END VAR
VAR OUTPUT
    Error :    BOOL;          //Konvertierungs- oder Stringfehler
END VAR
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion BHGDisp:

Signal	Art	Typ	Wertebereich	Bemerkung
Row	E	BYTE	0-3	Displayzeile 0: keine Displayausgabe 1: Zeile 1 2: Zeile 2 3: Zeile 1 und Zeile 2
ChrArray	E	STRING	>= string[32]	Hierin ist der komplette Displayinhalt abgelegt
Convert	E	BOOL		Aktivierung der Zahlenwandlung
Addr	E	POINTER		Zeigt auf die Variable, die konvertiert werden soll
DataType	E	BYTE	1-8	Datentyp der Variable 1: BOOL, 1 Zeichen 2: BYTE, 3 Zeichen 3: CHAR, 1 Zeichen 4: WORD, 5 Zeichen 5: INT, 6 Zeichen 6: DWORD, 7 Zeichen 7: DINT, 8 Zeichen 8: REAL, 9 Zeichen (siehe Parameter Digits)
StringAddr	E	INT	1-32	Adresse innerhalb der Variablen ChrArray
Digits	E	BYTE	1-4	nur relevant bei Datentyp REAL mit Vorzeichen (VZ) 1: 6.1 Stellen ohne VZ 2: 5.2 Stellen ohne VZ 3: 4.3 Stellen ohne VZ 4: 3.4 Stellen ohne VZ Stellenangaben sind ohne das VZ auszuweisen
Error	A	BOOL		Konvertierungsfehler, Zahlenüberlauf oder StringAddr fehlerhaft

## Wertebereiche

Wertebereiche von Datentypen	
Datentyp	darstellbarer Zahlenbereich
BOOL	0, 1
BYTE	0 bis 255
WORD	0 bis 65535
INT	- 32768 bis + 32767
DWORD	0 bis 9999999
DINT	- 9999999 bis + 9999999
REAL (Digits := 1)	- 999999.9 bis + 999999.9
REAL (Digits := 2)	- 99999.99 bis + 99999.99
REAL (Digits := 3)	- 9999.999 bis + 9999.999
REAL (Digits := 4)	- 999.9999 bis + 999.9999

## Aufrufbeispiel

```
CALL FC 13 (
  Row :=          MB 26,
  ChrArray :=    "strdat".dis //DB mit Namen strdat in der Symboltabelle,
                                     p, //Datenelement disp ist als String[32]
                                     //und komplett zugewiesen mit Zeichen
  Convert :=     M 90.1,
  Addr :=        P#M 20.0, //zu konvertierende Zahl
  DataType :=    MB 28, //Datentyp der Variablen
  StringAddr :=  MW 30,
  Digits :=      B#16#3, //3 Nachkommastellen
  Error :=       M 90.2);
```

## 2.12.19 FC 15: POS\_AX Positionierung von Linear- und Rundachsen

### Funktionsbeschreibung

(Nicht für Neuanwendungen verwenden, Funktion ist ab Software 3.6 im FC 18 integriert)

Mit dem FC POS\_AX können NC-Achsen in jeder Betriebsart auch von der PLC verfahren werden.

#### Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2).

Um die NC-Achsen über die PLC zu verfahren, muss die Verfahrkontrolle für die PLC aktiviert werden. Dieses kann z.B. durch Aufruf des FC "POS\_AX" mit Aktivieren des Parameters "Start" erreicht werden. Hierbei fordert der FC "POS\_AX" die Kontrolle der Achse von der NC an. Die NC meldet in der zugehörigen Achs-Nahtstelle DB 31, ... im Byte 68 den Status dieser Achse zurück, siehe /Listen/ (Buch 2), Nahtstellensignale power line.

Nach Beendigung ("InPos" ist TRUE, "Start" wechselt auf Null) wird die Kontrolle der Achse vom FC POS AX in einen neutralen Zustand geschaltet. Alternativ kann auch das PLC-Anwenderprogramm vor Aufruf des FC "POS\_AX" die Kontrolle für die PLC anfordern.

Hierdurch kann bei mehrfach hintereinander folgenden Aufrufen dieser Funktion ein besseres Reaktionsverhalten der Achsen erreicht werden, da der Umschaltvorgang im FC entfällt.

Die Aktivierung durch das PLC-Anwenderprogramm wird in der zugehörigen Achs-Nahtstelle im Byte 8 durchgeführt.

Nach Rückgabe der Kontrolle kann die Achse vom NC-Programm wieder programmiert werden.

---

### Hinweis

Bei Rundachsen kann bei Absolut-Positionierung durch die Programmierung eines negativen Vorschubwertes auf dem kürzesten Weg positioniert werden. Im inkrementellen Betrieb (Parameter "IC" := TRUE) kann durch das Vorzeichen des Parameters "Pos" die Verfahrrichtung bestimmt werden:

Positives Vorzeichen bewirkt das Fahren in Plus-Richtung.

Negatives Vorzeichen bewirkt das Fahren in Minus-Richtung.

Nach dem Aufruf des FC steht im ACCU1 ein Fehlerhinweis des NCK (nicht, wenn die Ausgangsparameter einem Datenbaustein zugeordnet sind). Im Regelfall ist dies der Wert 0 (Bedeutung: kein Fehler). Die Interpretation anderer Zahlenwerte ist aus der nachfolgenden Tabelle ersichtlich.

Der FC 15 ist zyklisch aufzurufen, bis das Signal "Activ" einen Flankenwechsel von 1 nach 0 liefert. Erst wenn das Signal "Activ" den Wert 0 geliefert hat, ist ein weiterer Start für diese Achse möglich (mindestens ein PLC-Zyklus muss mit dem nächsten Start gewartet werden). Dies gilt auch bei Veränderung der Zuordnung im Datenbyte 8.

Ein Abbrechen der Funktion ist nicht möglich über den Parameter "Start", sondern nur durch die axialen Nahtstellensignale (z.B. Restweglöschen). Ebenso liefert die axiale Nahtstelle Statussignale der Achse zurück, die gegebenenfalls auszuwerten sind (z.B. Genauhalt, Fahrbefehl).

---



### Warnung

Wenn mehrere Bausteinaufrufe (FC 15, FC 16, FC 18) für die gleiche Achse/Spindel im PLC-Anwenderprogramm programmiert wurden, dann ist eine Verriegelung dieser Funktionen durch bedingte Aufrufe im Anwenderprogramm notwendig.  
Der bedingte Aufruf eines gestarteten (Parameter Start oder Stopp = TRUE) ist solange zyklisch aufzurufen, bis ein Zustandswechsel des Ausgangs-Parameters "Activ" bzw. "InPos" von 1 nach 0 erfolgt.

### Fehlerkennungen

Konnte eine Funktion nicht ausgeführt werden, wird dies am Zustandsparameter "Error" mit "logisch 1" angezeigt. Die Fehlerursache ist am Bausteinausgang "State" kodiert.

Auflistung der Fehlerkennungen siehe Tabelle im Kapitel Bausteinbeschreibung, "FC 18: SpinCTRL Spindelsteuerung".

### Deklaration der Funktion

```
FUNCTION FC 15: VOID                                //POS_AX
VAR_INPUT
    Start :          BOOL;
    AxisNo :         INT;
    IC :            BOOL;
    Inch :          BOOL;
    HWheelOv :      BOOL;
    Pos :           REAL;
    FRate :         REAL;
END_VAR
VAR_OUTPUT
    InPos :          BOOL;
    Activ :          BOOL;
    StartErr :      BOOL;
    Error :          BOOL;
END_VAR
```

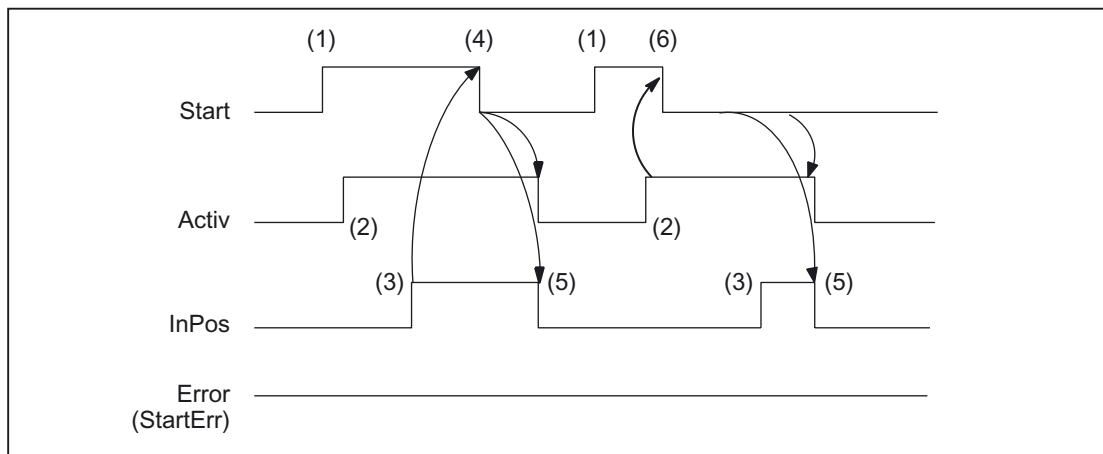
### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion POS\_AX:

Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		
AxisNo	E	BYTE	1 - 31	Nr. der zu verfahrenen Achse
IC	E	BOOL		0 = absolut 1 = inkrementell
Inch	E	BOOL		0 = mm 1 = inch
HWheelOv	E	BOOL		1 = Handradüberlagerung
Pos	E	REAL	± 0,1469368 E -38 bis ± 0,1701412 E +39	Position der Linearachse: mm Rundachse: Grad
FRate	E	REAL	± 0,1469368 E -38 bis ± 0,1701412 E +39	Vorschub der Linearachse: mm/Min Rundachse: Grad/Min
InPos	A	BOOL		1 = Position erreicht
Activ	A	BOOL		1 = aktiv
StartErr	A	BOOL		Achse kann nicht gestartet werden
Error	A	BOOL		Fehler beim Verfahren <sup>1)</sup>

<sup>1)</sup> Fehlerauswertung durch Anwender im PLC

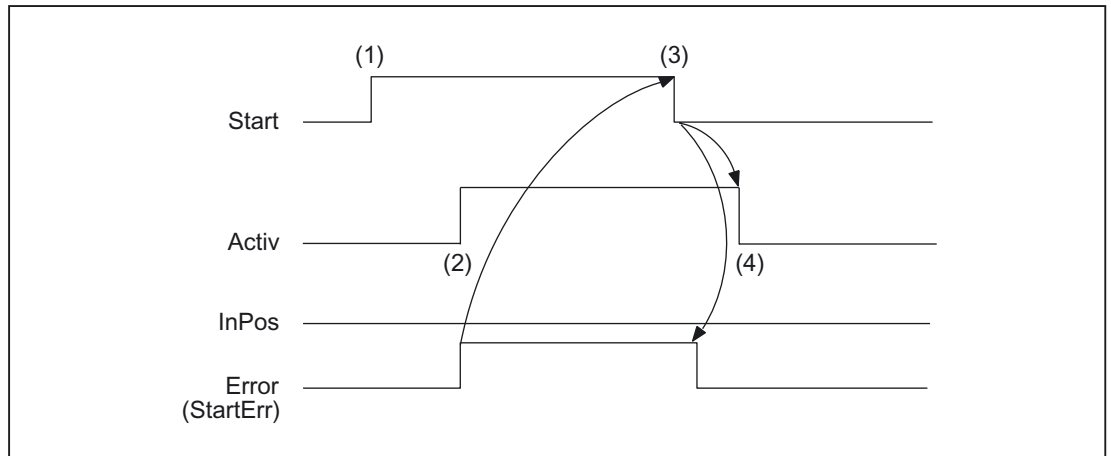
### Impulsdiagramm



- (1) Funktionsanstoß
- (2) Positionierachse aktiv
- (3) Positive Quittung: Position erreicht
- (4) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (5) Signalwechsel durch FC
- (6) Rücksetzen vom Funktionsanstoß nach Erhalt des Aktiv-Signals



## Impulsdiagramm (Fehlerfall)



- (1) Funktionsanstoß durch positive Flanke
- (2) Negative Quittung: Fehler aufgetreten
- (3) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC

## Aufrufbeispiel

```
CALL FC 15 (  
    Start := TRUE,  
    AxisNo := 5,  
    IC := #inkr, //z.B. lokale Variable  
    Inch := FALSE,  
    HWheelOv := FALSE,  
    Pos := MD160,  
    FRate := MD164,  
    InPos := A 36.0,  
    Activ := A 36.1,  
    StartErr := A 36.2,  
    Error := A 36.3);
```

## 2.12.20 FC 16: PART\_AX Positionierung von Teilungsachsen

### Funktionsbeschreibung

(Nicht für Neuanwendungen verwenden, Funktion ist ab Software 3.6 im FC 18 integriert)

Mit dem FC PART\_AX können NC-Achsen, die per Maschinendatum als "Teilungsachsen" definiert sind, auch von der PLC verfahren werden.

#### Literatur:

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Teilungsachsen (T1)

Um die Teilungsachsen über die PLC zu verfahren, muss die Verfahrkontrolle für die PLC aktiviert werden. Dieses kann z.B. durch Aufruf des FC "PART\_AX" mit Aktivieren des Parameters "Start" erreicht werden. Hierbei fordert der FC "PART\_AX" die Kontrolle der Achse von der NC an. Die NC meldet in der zugehörigen Achs-Nahtstelle DB 31, ... im Byte 68 den Status dieser Achse zurück, siehe /Listen/ (Buch 2), Nahtstellensignale power line.

Nach Beendigung ("InPos" ist TRUE, "Start" wechselt auf Null) wird die Kontrolle der Achse/Spindel vom FC PART\_AX in einen neutralen Zustand geschaltet. Alternativ kann auch das PLC-Anwenderprogramm vor Aufruf des FC "PART\_AX" die Kontrolle für die PLC anfordern.

Hierdurch kann ein besseres Reaktionsverhalten der Spindel/Achse bei mehrfach hintereinander folgenden Aufrufen dieser Funktion erreicht werden, da der Umschaltvorgang im FC entfällt.

Die Aktivierung durch das PLC-Anwenderprogramm wird in der zugehörigen Achs-Nahtstelle im Byte 8 durchgeführt.

Nach Rückgabe der Kontrolle kann die Achse vom NC-Programm wieder programmiert werden.

---

### Hinweis

Nach dem Aufruf des FC steht im ACCU1 ein Fehlerhinweis des NCK (nicht, wenn die Ausgangsparameter einem Datenbaustein zugeordnet sind). Im Regelfall ist dies der Wert 0 (Bedeutung: kein Fehler). Die Interpretation anderer Zahlenwerte ist aus der Tabelle für Fehlerkennungen im FC 15 ersichtlich.

Der FC 16 ist zyklisch aufzurufen, bis das Signal "Activ" einen Flankenwechsel von 1 nach 0 liefert. Erst wenn das Signal "Activ" den Wert 0 geliefert hat, ist ein weiterer Start für diese Achse möglich (mindestens ein PLC-Zyklus muss mit dem nächsten Start gewartet werden). Dies gilt auch bei Veränderung der Zuordnung im Datenbyte 8.

Ein Abbrechen der Funktion ist nicht möglich über den Parameter "Start", sondern nur durch die axialen Nahtstellensignale (z.B. Restweglöschen).

Ebenso liefert die axiale Nahtstelle Statussignale der Achse zurück, die gegebenenfalls auszuwerten sind (z.B. Genauhalt, Fahrbefehl).

---



### Warnung

Wenn mehrere Bausteinaufrufe (FC 15, FC 16, FC 18) für die gleiche Achse/Spindel im PLC-Anwenderprogramm programmiert wurden, dann ist eine Verriegelung dieser Funktionen durch bedingte Aufrufe im Anwenderprogramm notwendig. Der bedingte Aufruf eines gestarteten (Parameter "Start" oder "Stop" = TRUE) ist solange zyklisch aufzurufen bis ein Zustandswechsel des Ausgangs-Parameters "Activ" bzw. "InPos" von 1 nach 0 erfolgt.

### Deklaration der Funktion

```
FUNCTION FC 16: VOID                                //PART_AX
VAR_INPUT
  Start :          BOOL;
  AxisNo :        INT;
  IC :           BOOL;
  DC :           BOOL;
  Minus :         BOOL;           //Bewegung in negativer Richtung
  Plus :          BOOL;          //Bewegung in positiver Richtung

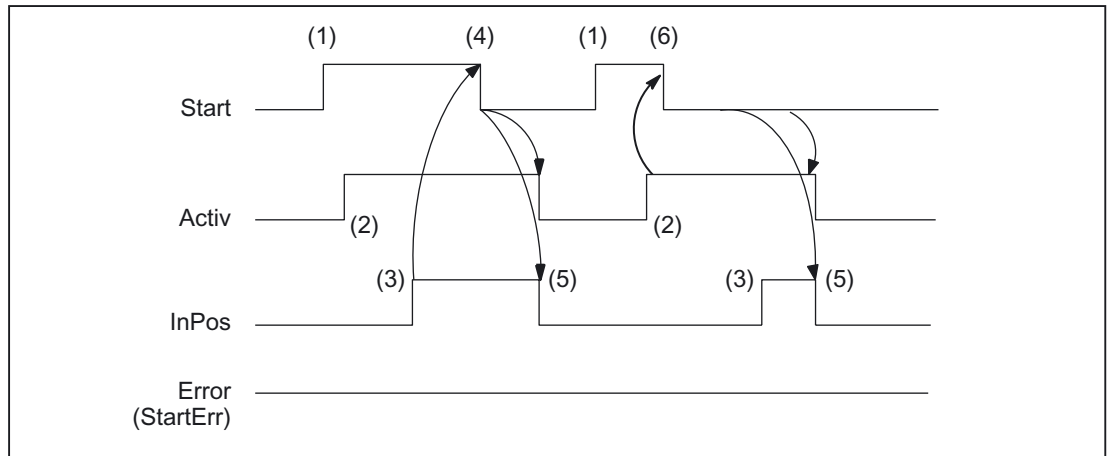
  Pos :           INT;
  FRate :         REAL;
END_VAR
VAR_OUTPUT
  InPos :         BOOL;
  Activ :         BOOL;
  StartErr :     BOOL;
  Error :        BOOL;
END_VAR
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion PART\_AX:

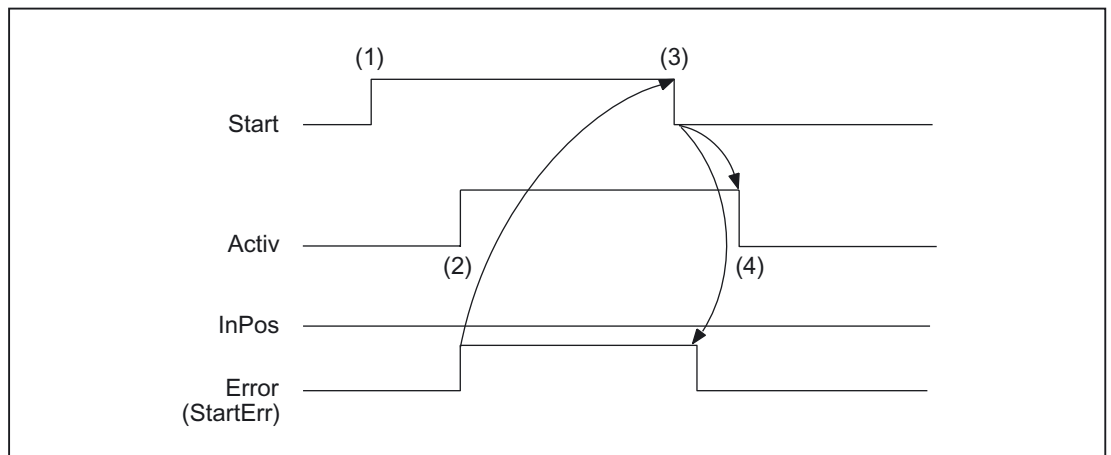
Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		
AxisNo	E	INT	1 - 31	Nr. der zu verfahrenen Achse
IC	E	BOOL		Richtungsvorgabe 0 = absolut 1 = inkrementell
DC	E	BOOL		0 = vorgegebene Richtung 1 = kürzester Weg wenn DC = 1 müssen Parameter IC, Minus, Plus = 0 sein
Minus	E	BOOL		0 : Rundachsbewegung wie Linearachse 1: Bewegung in negativer Richtung bei Rundachsen
Plus	E	BOOL		0 : Rundachsbewegung wie Linearachse 1: Bewegung in positiver Richtung bei Rundachsen
Pos	E	INT	0 bis + 32767	Teilungs-Positions-Nr.
FRate	E	REAL	$\pm 0,1469368 \text{ E } -38$ bis $\pm 0,1701412 \text{ E } +39$	Vorschub der Linearachse: mm/Min Rundachse: Grad/Min
InPos	A	BOOL		1 = Position erreicht
Activ	A	BOOL		1 = aktiv
StartErr	A	BOOL		Achse kann nicht gestartet werden
Error	A	BOOL		Fehler beim Verfahren <sup>1)</sup>
<sup>1)</sup> Fehlerauswertung durch Anwender im PLC				

## Impulsdiagramm



- (1) Funktionsanstoß durch positive Flanke
- (2) Positionierachse aktiv
- (3) Positive Quittung: Position erreicht
- (4) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (5) Signalwechsel durch FC
- (6) Rücksetzen vom Funktionsanstoß nach Erhalt des Aktiv-Signals

## Impulsdiagramm (Fehlerfall)



- (1) Funktionsanstoß durch positive Flanke
- (2) Negative Quittung: Fehler aufgetreten
- (3) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC

### Aufrufbeispiel

```
CALL FC 16 ( //Positionierung einer Teilungsachse
            Start := E72.4,
            AxisNo := 6,
            IC := FALSE,
            DC := #kurz, //z.B. lokale Variable
            Minus := FALSE,
            Plus := FALSE,
            Pos := MW 168,
            FRate := MD164,
            InPos := A 36.4,
            Activ := A 36.5,
            StartErr := A 36.6,
            Error := A 36.7);
```

### 2.12.21 FC 17: YDelta Stern-/Dreieck Umschaltung

#### Funktionsbeschreibung

Mit dem Baustein für die Stern-/Dreieck Umschaltung wird eine definierte Umschaltlogik zeitlich so gesteuert, dass diese Umschaltung auch bei laufender Spindel in beiden Richtungen vorgenommen werden kann. Der Baustein ist nur für digitale Hauptspindelantriebe verwendbar und muss für jede Spindel getrennt aufgerufen werden.

Die Umschaltung erfolgt über 2 getrennte Schütze und wird in 4 Schritten abgearbeitet:

Schritt 1:	Löschen des Nahtstellensignals DB31, ... DBX21.5 (Motoranwahl erfolgt) im zugehörigen Achs-DB und Anmelden des Umschaltvorgangs über A mit DB31, ... DBX21.3 (Motoranwahl).
Schritt 2:	Sobald die Rückmeldung NST DB31, ... DBX93.7 (Impulse freigegeben) = 0 und die Quittierung der angemeldeten Motoranwahl vom Antrieb vorliegt, wird das bisher angezogene Schütz ausgeschaltet.
Schritt 3:	nach der vom Anwender im Parameter "TimeVal" parametrisierten Zeit wird das andere Schütz eingeschaltet.
Schritt 4:	Nach einer erneuten Zeit wird die Umschaltung an den Antrieb mit NST DB31, ... DBX21.5 (Motoranwahl erfolgt) gemeldet.

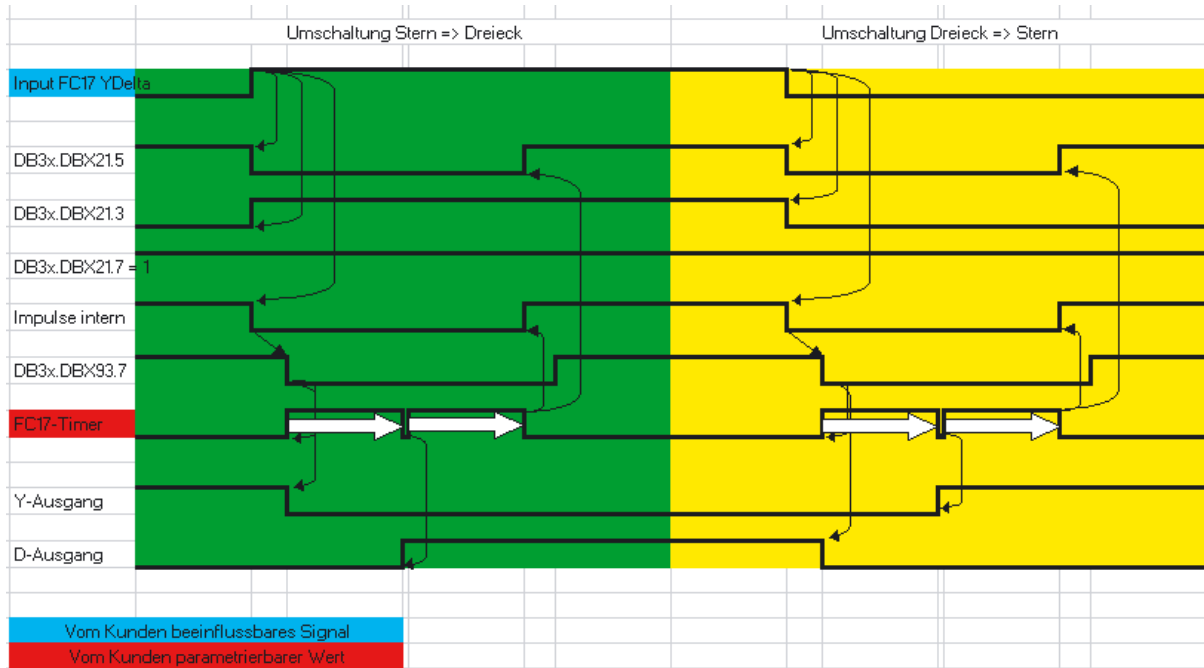


Bild 2-29 Stern-/Dreieck-Umschaltung

Weitere Erläuterungen zu Motordrehzahlanpassungen entnehmen Sie bitte:

**Literatur:**

/FB1/ Funktionshandbuch Grundfunktionen; Spindeln (S1);  
 Kapitel "Projektierbare Getriebearpassungen"  
 /FB1/ Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsys.,  
 Regelung (G2)

**Fehlermeldung**

Wenn Parameter "SpindleIFNo" nicht im zulässigen Bereich liegt, erfolgt Stopp der PLC mit Ausgabe der Alarmmeldungs-Nummer 401702.

**Besonderheiten**

Bei Parametrierung des "TimeVal" mit dem Wert 0 wird ein Standardwert von 100 ms verwendet. Bei einem Wert kleiner 50 ms wird der Minimalwert von 50 ms eingestellt.

Der Baustein ist absolut aufzurufen.

---

**Hinweis**

Eine Umschaltung erfolgt nicht, wenn die Spindel in einer Achs-Betriebsart wie z.B. M70, SPOS ist.

---

## Randbedingungen

Mit der Stern-/Dreieck Umschaltung digitaler Hauptspindelantriebe wird ein Vorgang angestoßen der auch reglungstechnische Abläufe beinhaltet. Da die Regelung die automatische Stern-/Dreieck Umschaltung unterstützt, ergeben sich einige Randbedingungen, die zu beachten sind:

- Aufgrund des automatischen Wegschaltens der Impulse im Antrieb werden gleichzeitig mit dem NST DB31, ... DBX93.7 (Impulse freigegeben) die NST DB31, ... DBX61.7 (Stromregler aktiv) und DB31, ... DBX61.6 (Drehzahlregler aktiv) weggeschaltet.
- Wird bei drehender Spindel und eingeschaltetem Lageregler der Spindel mit NST DB31, ... DBX61.5 (Lageregler aktiv) von Stern nach Dreieck umgeschaltet, so führt dies zum Alarm 25050 "Konturüberwachung".
- Eine angestoßene Stern-/Dreieck Umschaltung mit FC 17 kann nicht vom Anwender z.B. durch betriebsmäßiges Warten auf eine erfolgreich durchgeführte Umschaltung der Stern-/Dreieck Schütze verzögert werden. Dieses Signalspiel kann vom Anwender durch eine PLC-Logik realisiert werden.

## Deklaration der Funktion

### AWL-Darstellung

```
VAR_INPUT
  YDelta :          BOOL;           //Stern = 0, Dreieck = 1
  SpindleIFNo :    INT;            //Maschinenachs-Nummer
  TimeVal :        S5TIME;        //Zeitwert
  TimerNo :        INT;            //Timer des Anwenders für Umschaltzeit
END_VAR
VAR_OUTPUT
  Y :              BOOL;           //Stern-Schütz
  Delta :          BOOL;           //Dreieck-Schütz
END_VAR
VAR_IN_OUT
  Ref :           WORD;            //Zustandswort des Bausteins (Instanz)
END_VAR
```



## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion YDelta:

Signal	Art	Typ	Wertebereich	Bemerkung
YDelta	E	BOOL		= Stern = Dreieck Die Umschaltflanke des Signals löst die Umschaltung aus.
SpindleIFNo	E	INT	1..	Nummer der Achsschnittstelle, die als Spindel deklariert ist.
TimeVal	E	S5time	0..	Umschaltzeit
TimerNo	E	INT	10..	Zeitglied für die Programmierung der Wartezeit.
Y	A	BOOL		Ansteuerung des Stern-Schütz
Delta	A	BOOL		Ansteuerung des Dreieck-Schütz
Ref	E/A	WORD		Instanz für Zustandsinformationen. Interne Verwendung

## Aufrufbeispiel

```
CALL FC 17 (
  YDelta :=      e 45.7,           //Stern Dreieck
  SpindleIFNo := 4,
  TimeVal :=     S5T#150ms,
  TimerNo :=     10,              //Timer 10
  Y :=          a 52.3,          //Stern-Schütz
  Delta :=      a 52.4,          //Dreieck-Schütz
  Ref :=        mw 50);         //Instanz
```

## 2.12.22 FC 18: SpinCtrl Spindelsteuerung

### Funktionsbeschreibung

Mit dem FC SpinCtrl können Spindeln und Achsen von der PLC gesteuert werden.

#### Literatur:

/FB1/ Funktionshandbuch Grundfunktionen; Spindeln (S1)

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Positionierachsen (P2)

/FB2/ Funktionshandbuch Erweiterungsfunktionen; Teilungsachsen (T1)

Der Baustein unterstützt dabei die Funktionen:

- Spindel positionieren
- Spindel drehen
- Spindel pendeln
- Teilungsachsen
- Positionierachsen

Jede Funktion wird durch die positive Flanke des entsprechenden Anstoß-Signals (Start, Stop) angestoßen. Das Anstoß-Signal muss solange auf logisch "1" bleiben, bis die Funktion durch InPos="1" oder Error = "1" positiv oder negativ quittiert wurde. Die Ausgangsparameter werden mit Rücksetzen des jeweiligen Anstoß-Signals und dem Ende der Funktion abgelöscht.

Um die Spindel/Achsen über die PLC zu beeinflussen, muss sie für die PLC aktiviert werden. Dieses kann z.B. durch Aufruf des FC "SpinCtrl" mit Aktivieren des Parameters "Start" bzw. "Stop" erreicht werden. Hierbei fordert der FC "SpinCtrl" die Kontrolle der Spindel/Achse von der NC an.

Die NC meldet in der zugehörigen Spindel-/Achs-Nahtstelle DB 31, ... im Byte 68 den Status dieser Spindel/Achse zurück siehe /Listen/ (Buch 2), Nahtstellensignale power line.

Nachdem die Achse / Spindel der PLC gehört, kann über die zugehörige Achsnahtstelle der Fahrbefehl für den Aktivzustand ausgewertet werden.

Nach Beendigung ("InPos" ist TRUE, "Start" wechselt auf Null) wird die Kontrolle der Achse/Spindel vom FC "SpinCtrl" in einen neutralen Zustand geschaltet. Alternativ kann auch das PLC-Anwenderprogramm vor Aufruf des FC "SpinCtrl" die Kontrolle für die PLC anfordern.

Hierdurch kann ein besseres Reaktionsverhalten der Spindel/Achse bei mehrfach hintereinander folgenden Aufrufen dieser Funktion erreicht werden, da der Umschaltvorgang im FC entfällt.

Die Aktivierung durch das PLC-Anwenderprogramm wird in der zugehörigen Spindel-Nahtstelle im Byte 8 durchgeführt.

Nach Rückgabe der Kontrolle kann die Spindel vom NC-Programm wieder programmiert werden.

### **Hinweis**

#### **Aufrufhinweis:**

Der FC 18 ist zyklisch aufzurufen, bis das Signal "InPos", bzw. bei Fehler das Signal "Error", einen Flankenwechsel von 1 nach 0 liefert. Erst wenn das Signal "InPos"/"Error" den Wert 0 geliefert hat, ist ein weiterer "Start" oder "Stop" für diese Spindel/Achse möglich (mindestens ein PLC-Zyklus muss mit dem nächsten "Start" oder "Stop" gewartet werden). Dies gilt auch bei Veränderung der Zuordnung im Datenbyte 8 der axialen Nahtstelle.

#### **Abbruch:**

Ein Abbrechen der Funktion ist nicht möglich über den Parameter "Start" oder "Stop", sondern nur durch die axialen Nahtstellensignale (z. B. Restweglöschen). Ebenso liefert die axiale Nahtstelle Statussignale der Achse zurück, die gegebenenfalls auszuwerten sind (z.B. Genauhalt, Fahrbefehl).

#### **InPos bei Spindel - Drehen/Pendeln:**

Bei der Funktion "Spindel drehen" und auch bei "Spindel pendeln" ist die Bedeutung des Parameters "InPos" wie folgt festgelegt:  
Soll Drehzahl wird ausgegeben --> Funktion wurde ohne Fehler gestartet.  
Das Erreichen der geforderten Spindeldrehzahl muss über die Spindelnahtstelle ausgewertet werden.

#### **Gleichzeitigkeit:**

Mehrere Achsen können zeitgleich oder auch zeitversetzt durch die Bausteine FC 15, 16, 18 verfahren werden. Die Obergrenze ist über die max. Achszahl begrenzt. Der NCK wickelt die Funktionsanforderung der PLC (FC 15, 16, 18) über eigenständige Schnittstellen je Achse / Spindel ab.

#### **Achssperre:**

Bei gesetzter Achssperre bewegt sich eine über FC 18 gesteuerte Achse nicht. Es wird nur ein simulierter Istwert erzeugt. (Verhalten wie bei NC-Programmierung).

---



### **Warnung**

Wenn mehrere Bausteinaufrufe (FC 15, FC 16, FC 18) für die gleiche Achse/Spindel im PLC-Anwenderprogramm programmiert wurden, dann ist eine Verriegelung dieser Funktionen durch bedingte Aufrufe im Anwenderprogramm notwendig. Der bedingte Aufruf eines gestarteten (Parameter "Start" oder "Stop" = TRUE) ist solange zyklisch aufzurufen bis ein Zustandswechsel des Ausgangs-Parameters Activ bzw. InPos von 1 nach 0 erfolgt.

---

## Funktionen

### 1. Spindel positionieren:

Folgende Signale sind relevant:

Start :	Anstoß-Signal
Funct :	"1" = Spindel positionieren
Mode :	Positionierungs-Mode 1, 2, 3, 4
AxisNo :	Nummer der Maschinenachse
Pos :	Position
FRate :	Positioniergeschwindigkeit, wenn FRate = 0, wird Wert aus MD35300: SPIND_POSCTRL_VELO (Lageregeleinschaltdrehzahl) genommen
InPos :	wird bei Erreichen der Position mit "Genauhalt Fein" auf "1" gesetzt
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

### 2. Spindel drehen:

Folgende Signale sind relevant:

Start :	Anstoß-Signal für Start drehen
Stop :	Anstoß-Signal für Stop drehen
Funct :	"2" = Spindel drehen
Mode :	Positionierungs-Mode 5 (Drehrichtung M4) Positionierungs-Mode < >5 (Drehrichtung M3)
AxisNo :	Nummer der Maschinenachse
FRate :	Spindel-Drehzahl
InPos :	Funktion wurde ohne Fehler gestartet
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

### 3. Spindel pendeln:

Folgende Signale sind relevant:

Start:	Anstoß-Signal für Start pendeln
Stop:	Anstoß-Signal für Stop pendeln
Funct:	"3" = Spindel pendeln
AxisNo:	Nummer der Maschinenachse
Pos:	Sollgetriebestufe
InPos:	Solldrehzahl wird ausgegeben
Error:	Bei Positionierfehler = "1"
State:	Fehlercode

Die Pendeldrehzahl stammt aus dem Maschinendatum:  
 MD35400 SPIND\_OSCILL\_DES\_VELO

MD35010 GEAR_STEP_CHANGE_ENABLE = 0	Funktion	MD35010 GEAR_STEP_CHANGE_ENABLE = 1	Funktion
Pos = 0	Pendeln	Pos = 0	
Pos = 1	Pendeln	Pos = 1	Pendeln mit Getriebestufenwechsel M41
Pos = 2	Pendeln	Pos = 2	Pendeln mit Getriebestufenwechsel M42
Pos = 3	Pendeln	Pos = 3	Pendeln mit Getriebestufenwechsel M43
Pos = 4	Pendeln	Pos = 4	Pendeln mit Getriebestufenwechsel M44
Pos = 5	Pendeln	Pos = 5	Pendeln mit Getriebestufenwechsel M45

#### 4. Teilungsachsen verfahren:

Folgende Signale sind relevant:

Start :	Anstoß-Signal
Funct :	"4" = Teilungsachse

---

#### Hinweis

Bei  
Funct: "4" = Teilungsachse

Die Modulowandlung ist vergleichbar mit Teilungsposition anfahren über POS[AX] = CIC (Wert) im Teileprogramm.

---

Mode :	Positionierungs-Mode 0, 1, 2, 3, 4
AxisNo :	Nummer der Maschinenachse
Pos :	Teilungs-Position
FRate :	Positioniergeschwindigkeit; wenn FRate = 0, wird Wert aus Maschinendatum POS_AX_VELO genommen (Einheit wie im Maschinendatum eingestellt)
InPos :	wird bei Erreichen der Position mit "Genauhalt Fein" auf "1" gesetzt
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

**5. bis 8. Achsen positionieren:**

Folgende Signale sind relevant:

Start :	Anstoß-Signal
Funct :	"5 bis 8" = Achsen positionieren
Mode :	Positionierungs-Mode 0, 1, 2, 3, 4
AxisNo :	Nummer der Maschinenachse
Pos :	Position
FRate :	Positioniergeschwindigkeit; wenn FRate = 0, wird Wert aus Maschinendatum POS_AX_VELO genommen (Einheit wie im Maschinendatum eingestellt)
InPos :	wird bei Erreichen der Position mit "Genauhalt Fein" auf "1" gesetzt
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

**9. Spindel drehen mit automatischer Getriebestufenwahl:**

Folgende Signale sind relevant:

Start :	Anstoß-Signal für Start drehen
Stop :	Anstoß-Signal für Stop drehen
Funct :	"9" = Spindel drehen mit Getriebestufenwahl
Mode :	Positionierungs-Mode 5 (Drehrichtung M4)
	Positionierungs-Mode < > 5 (Drehrichtung M3)
AxisNo :	Nummer der Maschinenachse
FRate :	Spindel-Drehzahl
InPos :	Solldrehzahl wird ausgegeben
Error :	Bei Positionierfehler = "1"
State :	Fehlercode

**10./11. Spindel drehen mit konstanter Schnittgeschwindigkeit:**

Voraussetzung zur Ausführung ist die Aktivierung der Funktion "konstante Schnittgeschwindigkeit" durch das NC-Programm.

Folgende Signale sind relevant:

Start :	Anstoß-Signal für Start drehen
Stopp :	Anstoß-Signal für Stopp drehen
Funct :	"B#16#0A = Spindel drehen mit konstanter Schnittgeschwindigkeit (m/min)
Funct :	"B#16#0B = Spindel drehen mit konstanter Schnittgeschwindigkeit (feet/min)
Mode :	Positionierungs-Mode 5 (Drehrichtung M4)
	Positionierungs-Mode <>5 (Drehrichtung M3)
AxisNo :	Nummer der Maschinenachse
FRate :	Schnittgeschwindigkeit
InPos :	Solldrehzahl wird ausgegeben
Error :	Bei Positionsfehler = "1"
State :	Fehlercode

## Deklaration der Funktion

```
FUNCTION FC 18: VOID                                     //SpinCtrl
VAR_INPUT
    Start :          BOOL;
    Stop  :          BOOL;
    Funct :          BYTE;
    Mode  :          BYTE;
    AxisNo :         INT;
    Pos   :          REAL;
    FRate :          REAL;
END_VAR
VAR_OUTPUT
    InPos :          BOOL;
    Error :          BOOL;
    State :          BYTE;
END_VAR
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion SpinCtrl.

Signal	Art	Typ	Wertebereich	Bemerkung
Start	E	BOOL		Start Spindel-Steuerung von PLC
Stop	E	BOOL		Stop Spindel-Steuerung von PLC
Funct	E	BYTE	1 bis B#16#0B	1: Spindel positionieren 2: Spindel drehen 3: Spindel pendeln 4: Teilungsachse 5: PosAchse metrisch 6: PosAchse inch 7: PosAchse metrisch mit Handradüberlagerung 8: PosAchse inch mit Handradüberlagerung 9: Spindel drehen mit automatischer Getriebestufenwahl A: Spindel drehen mit konstanter Schnittgeschwindigkeit (m/min) B: Spindel drehen mit konstanter Schnittgeschwindigkeit (feet/min)
Mode	E	BYTE	0 bis 5	0: Pos auf Absolutposition 1: Pos inkrementell 2: Pos kürzester Weg 3: Pos absolut, positive Anfahrriichtung 4: Pos absolut, negative Anfahrriichtung 5: Drehrichtung wie M4
AxisNo	E	INT	1 - 31	Nr. der zu verfahrenen Achse, Spindel
Pos	E	REAL	± 0,1469368 E -38 bis ± 0,1701412 E +39	Rundachse: Grad Teilungsachse: Teilungsposition Linearachse: mm oder inch
FRate	E	REAL	± 0,1469368 E -38 bis ± 0,1701412 E +39	Rundachse und Spindel: Umdr./Min siehe unterhalb der Tabelle zum Thema FRate
InPos	A	BOOL		1 = Position erreicht, bzw. Funktion ausgeführt
Error	A	BOOL		1 = Fehler
State	A	BYTE	0 bis 255	Fehlercode

### FRate

Die Vorschubgeschwindigkeit im FC 18 kann auch angegeben werden als:

1. Schnittgeschwindigkeit in der Einheit m/min bzw. ft/min
2. konstante Scheibenumfangsgeschwindigkeit in m/s bzw. ft/s

Voraussetzung für diese alternativen Geschwindigkeiten ist eine Aktivierung dieser Funktion durch das NC-Programm. Rückmeldungen für die erfolgreiche Aktivierung sind in der Achsnahtstelle im Byte 84 zu finden.

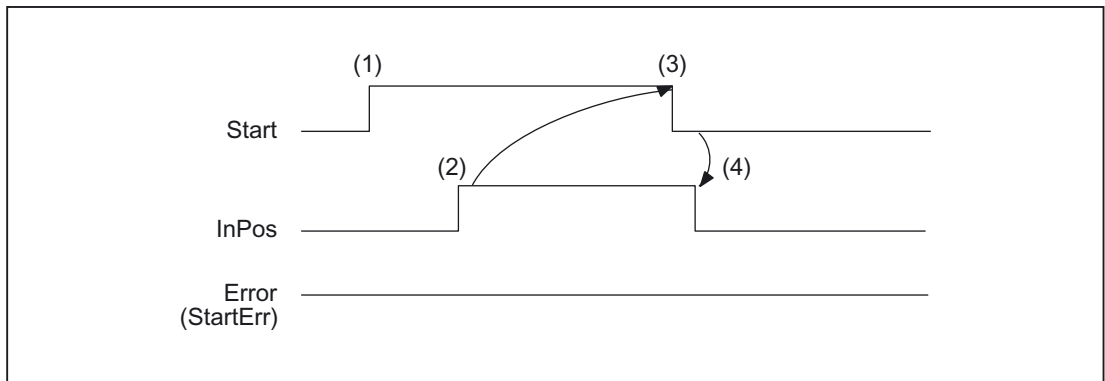


## Fehlerkennungen

Konnte eine Funktion nicht ausgeführt werden, wird dies am Zustandsparameter "Error" mit "logisch 1" angezeigt. Die Fehlerursache ist am Bausteinausgang "State" kodiert.

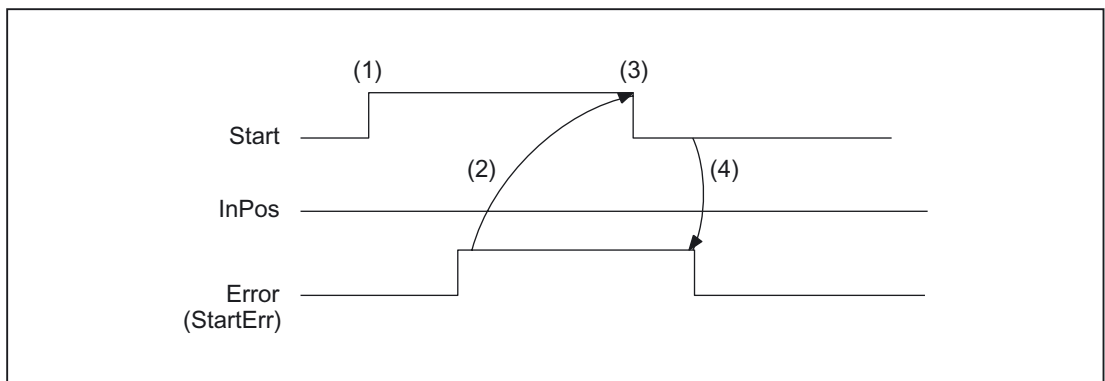
State	Bedeutung
Fehler, die durch PLC-Handling auftreten	
1	es wurden mehrere Funktionen der Achse/Spindelgleichzeitig aktiviert
20	es wurde eine Funktion gestartet, ohne dass die Position erreicht wurde
30	Die Achse/Spindel wurde vor Ende der Bewegung an die NC abgegeben
40	NCK interner Fehler
Fehler, die durch Behandlung des NCK auftreten. Die Alarmnummern sind im Diagnosehandbuch SINUMERIK 840D/840Di/810D beschrieben.	
100	entspricht Alarmnummer: 16830
105	entspricht Alarmnummer: 16770
106	entspricht Alarmnummer: 22052
107	entspricht Alarmnummer: 22051
108	entspricht Alarmnummer: 22050
109	entspricht Alarmnummer: 22055
110	Geschwindigkeit/Drehzahl ist negativ
111	Soll-Drehzahl ist Null
112	ungültige Getriebestufe
115	programmierte Position wurde nicht erreicht
117	in der NC ist G96/G961 <b>nicht aktiv</b>
118	in der NC ist G96/G961 <b>noch aktiv</b>
120	keine Teilungsachse
121	Teilungspositions-Fehler
125	DC (kürzester Weg) nicht möglich
126	Absolutwert Minus nicht möglich
127	Absolutwert Plus nicht möglich
130	Softwareendschalter plus
131	Softwareendschalter Minus
132	Arbeitsfeldbegrenzung Plus
133	Arbeitsfeldbegrenzung Minus
System- oder andere schwere Alarme:	
200	entspricht Systemalarmnummer: 450007

## Impulsdiagramm



- (1) Funktionsanstoß durch positive Flanke durch Start oder Stop
- (2) Positive Quittung: Funktion ausgeführt / Position erreicht
- (3) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC

## Impulsdiagramm (Fehlerfall)



- (1) Funktionsanstoß durch positive Flanke durch Start oder Stop
- (2) Negative Quittung: Fehler aufgetreten
- (3) Rücksetzen vom Funktionsanstoß nach Erhalt der Quittung
- (4) Signalwechsel durch FC

## Aufrufbeispiele

### 1. Spindel positionieren:

```
//positive Quittung setzt Start zurück:
U M112.0;           // InPos
R M 100.0;         // Start
//negative Quittung , nach Fehlerauswertung (State: MB114) mit T12 Start rücksetzen
U M113.0;         // Error
U E 6.4;          //Taste T12
R M 100.0;         //Start
//Starten mit T13
U E 6.3;          //Taste T13
UN M 112.0;       //neuen Start erst wenn InPos bzw. Error = 0
UN M 113.0;
S M 100..0;

CALL FC 18 (
    Start :=          M100.0,
    Stop :=           FALSE,
    Funct :=          B#16#1,           //Spindel positionieren
    Mode :=           B#16#2,           //kürzester Weg
    AxisNo :=         5,
    Pos :=            MD104,
    FRate :=          MD108,
    InPos :=          M112.0,
    Error :=          M113.0,
    State :=          MB114);
```

### 2. Start Spindel drehen:

```
CALL FC 18 (
    Start :=          M100.0,
    Stop :=           FALSE,
    Funct :=          B#16#2,           //Spindel drehen
    Mode :=           B#16#5,           //Drehrichtung wie M4
    AxisNo :=         5,
    Pos :=            0.0,
    FRate :=          MD108,
    InPos :=          M112.0,
    Error :=          M113.0,
    State :=          MB114);
```

### 3. Start Spindel pendeln:

```
CALL FC 18 (  
    Start :=      M100.0,  
    Stop  :=      FALSE,  
    Funct :=      B#16#3,      //Spindel pendeln  
    Mode  :=      B#16#0,  
    AxisNo :=     5,  
    Pos   :=      0.0,  
    FRate :=      MD108,  
    InPos :=      M112.0,  
    Error :=      M113.0,  
    State :=      MB114);
```

### 4. Teilungsachse verfahren

```
CALL FC 18 (  
    Start :=      M100.0,  
    Stop  :=      FALSE,      //nicht genutzt  
    Funct :=      B#16#4,      //Teilungsachse verfahren  
    Mode  :=      B#16#0,      //absolut positionieren  
    AxisNo :=     4,  
    Pos   :=      MD104,      //Vorgabe in REAL: 1.0;2.0;..  
    FRate :=      MD108,  
    InPos :=      M112.0,  
    Error :=      M113.0,  
    State :=      MB114);
```

### 5. Achsen positionieren

```
CALL FC 18 (  
    Start :=      M100.0,  
    Stop  :=      FALSE,      //nicht genutzt  
    Funct :=      B#16#5,      //Achsen positionieren  
    Mode  :=      B#16#1,      //inkrementell positionieren  
    AxisNo :=     6,  
    Pos   :=      MD104,  
    FRate :=      MD108,  
    InPos :=      M112.0,  
    Error :=      M113.0,  
    State :=      MB114);
```

## 2.12.23 FC 19: MCP\_IFM Übertragung der MSTT-Signale an die Nahtstelle

### Funktionsbeschreibung

Mit dem FC MCP\_IFM (M-Variante) werden von der Maschinensteuertafel (MSTT) an die entsprechenden Signale der NCK-/PLC-Nahtstelle übertragen:

- Betriebsarten
- Achsanwahlen
- WKS/MKS-Umschaltung
- Verfahrstasten
- Overrides
- Schlüsselschalter

Im Grundprogramm (FC 2) werden weiterhin die Handradanwahlen, Betriebsarten und weitere Bediensignale von der Bedientafel (BT) bzw. vom MMC an die NCK-/PLC-Nahtstelle so übertragen, dass bei den Betriebsarten eine wahlweise Anwahl von der MSTT oder von der BT her möglich ist.

Die Übertragung der HMI-Signale an die Nahtstelle kann durch Setzen des Parameters im FB 1 (DB 7) "MMCToIF" auf den Wert "FALSE" abgeschaltet werden.

Für **Vorschub-Override**, **Achsfahr**- und **INC-Tasten** gelten abhängig von der aktiven Betriebsart bzw. vom angewählten Koordinatensystem folgende Festlegungen:

- **Vorschub-Override:**
  - Der Vorschub-Override wird auf die Nahtstelle des angewählten Kanals und auf die Nahtstelle der Achsen transferiert.
  - Die Vorschub-Override-Signale werden zusätzlich zum Nahtstellenbyte "Eilgangkorrektur" (DBB 5) an den NC-Kanal übergeben, wenn das MMC-Signal "Vorschubkorrektur für Eilgang wirksam" gesetzt ist (Ausnahme: Schalterstellung "Null"). Weiterhin wird mit diesem MMC-Signal auch "Eilgangkorrektur wirksam" gesetzt.
- **Maschinenfunktion INC- und Achsfahrtasten:**
  - Bei angewähltem MKS werden die Signale auf die Nahtstelle der angewählten Maschinenachsetransferiert.
  - Bei angewähltem WKS werden die Signale auf die Geo-Achs-Nahtstelle des parametrisierten Kanals transferiert.
  - Bei Umschaltung zwischen MKS und WKS erfolgt generell eine Abwahl der bis dahin angewählten Achse.

Die **Handrad-Anwahlsignale von MMC** werden decodiert und in der zugehörigen Maschinen-Achs-Nahtstelle oder in der Geo-Achs-Nahtstelle des jeweiligen Handrads aktiviert (nur wenn im FB 1 der Parameter "HWheelMMC:= TRUE" ist).

Die Ansteuerung der zugehörigen LEDs der Maschinensteuertafel wird aus der Rückmeldung entsprechender Anwahlen abgeleitet.

Vorschub- und Spindel-Start/Stop werden nicht an die Nahtstelle übertragen, sondern als Signal "FeedHold" bzw. "SpindleHold" selbsthaltend ausgegeben. Der Anwender kann diese Signale mit weiteren Signalen verknüpfen, die zu Vorschub- oder Spindel-Halt führen sollen (dies kann z.B. über die entsprechenden Eingangssignale des FC 10: AL\_MSG erfolgen). Zusätzlich werden die zugehörigen LEDs mit angesteuert.

Bei Ausfall der Maschinensteuertafel werden die Signale, die von dieser kommen, mit Null vorbesetzt; ebenso die Ausgangssignale "FeedHold" und "SpindleHold".

In einem PLC-Zyklus sind Mehrfachaufrufe des FC 19 bzw. auch FC 24, FC 25, FC 26 zulässig. Hierbei steuert der erste Aufruf im Zyklus die LED-Anzeigen an. Weiterhin werden im ersten Aufruf alle Aktionen des parametrisierten Bausteins durchgeführt. Bei den weiteren Aufrufen findet nur noch eine reduzierte Bearbeitung von Kanal und BAG-Schnittstelle statt. Die Geometrieachsen werden nur im ersten Aufruf des Bausteins im Zyklus mit Richtungsvorgaben versorgt.

Die Einzelsatz-An-/Abwahl wird nur durch den ersten Aufruf im Zyklus beeinflusst.

Die zweite Maschinensteuertafel kann bearbeitet werden, wenn der Parameter "BAGNo" um B#16#10 erhöht ist. Bei der Parametrierung ist die BAG Nummer in dem unteren Nibble (untere 4 Bits) enthalten.

"BAGNo" = 0 oder B#16#10 bedeutet keine Bearbeitung der BAG Signale.

"ChanNo" = 0 bedeutet keine Bearbeitung der Kanal-Signale.

Die INC-Anwahlen werden nur noch in die BAG-Schnittstelle übertragen. Hierdurch ergeben sich Laufzeit-Verbesserungen. Die Aktivierung für diese Vorgabe findet über den DB10 ... DBX57.0 (INC-Eingänge im BAG-Bereich aktiv) durch diesen Baustein einmalig nach Hochlauf statt.

Weiterhin können zwei Maschinensteuertafeln parallel durch diesen Baustein bearbeitet werden. Hierbei ist der Aufruf des Bausteins für die 2. Maschinensteuertafel im OB 1 Zyklus zeitlich hinter den Aufruf für die 1. MSTT zu setzen. Eine Unterstützung von 2 MSTT ist in den Maschinensteuertafel-Bausteinen bis zu bestimmten Grenzen vorhanden (nicht unterstützt werden vom Standard Achsnummern 10 bis 31, gegenseitige Verriegelungen der Achsanwahlen bei 2 MSTT).

## Flexible Achskonfiguration

Ab dem Softwarestand 6 ist eine Flexibilität bezüglich der Zuordnung Achsanwahlen bzw. Richtungstasten von Maschinen-Achsnummern verfügbar.

Der Einsatz von 2 Maschinensteuertafeln, die gleichzeitig betrieben werden sollen, wird durch die MSTT Bausteine insbesondere für den Anwendungsfall 2 Kanäle, 2 BAGs besser unterstützt. Bei den Achs-Tabellen der jeweiligen MSTT ist zu beachten, dass die Achsnummern auch in der parametrisierten BAG Nummer des MSTT-Bausteins angegeben sind.

Für diese Flexibilität existieren Tabellen für Achsnummern im DB 10.

Für die 1. Maschinensteuertafel (MSTT) beginnt die Tabelle ab dem Byte 8

(symbolischer Name: MCP1AxisTbl[1..22]) und

für die 2. Maschinensteuertafel (MSTT) ab dem Byte 32

(symbolischer Name: MCP2AxisTbl[1..22]). Hier sind die Maschinen-Achsnummern byteweise einzutragen.

Der Eintrag 0 ist in der Achstabelle zulässig. Eine Prüfung auf eine unzulässige Achsnummer findet nicht statt und kann bei Falscheintrag zum PLC Stopp führen.

Für den **FC 19** kann auch eine Begrenzung der **maximal möglichen Achsanwahlen** erfolgen.

Diese Obergrenze wird für die

1. Maschinensteuertafel im DB10, ... DBW30 (symbolischer Name: MCP1MaxAxis) bzw. für die
2. Maschinensteuertafel im DB10, ... DBW54 (symbolischer Name: MCP2MaxAxis) für die jeweilige MSTT eingestellt werden.

Der voreingestellte Wert ist 0, damit wirkt die konfigurierte maximale Achsanzahl. Die Achsnummern und die Begrenzung können auch dynamisch angepasst werden. Danach muss eine erneute Achsanwahl bei FC 19 erfolgen. Während des Fahrens von Achsen über die jeweiligen Richtungstasten darf keine Umschaltung der Achsnummern erfolgen. Voreingestellt ist der Kompatibilitätsmodus mit den Achsnummern **1 bis 9** für beide MSTT und die Begrenzung auf die konfigurierte Achsanzahl.

## Deklaration der Funktion

```

FUNCTION FC 19: VOID
//NAME :                               MCP_IFM

VAR_INPUT
    BAGNo :                               BYTE;
    ChanNo :                               BYTE;
    SpindleIFNo :                           BYTE;
END_VAR

VAR_OUTPUT
    FeedHold :                               BOOL;
    SpindleHold :                             BOOL;
END_VAR

BEGIN
END_FUNCTION
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "MCP\_IFM":

Signal	Art	Typ	Wertebereich	Bemerkung
BAGNo	E	BYTE	0 - b#16# und b#16#10 - b#16#1A	BAG-Nr., in die die Betriebsartensignale übertragen werden. BAGNo >= b#16#10 bedeutet Zugriff auf die 2. Maschinensteuertafel.
ChanNo	E	BYTE	0 - B#16#0A	Kanal-Nr. für die Kanalsignale
SpindleIFNo	E	BYTE	0 - 31 (B#16#1F)	Nummer der Achsschnittstelle, die als Spindel deklariert ist.
FeedHold	A	BOOL		Vorschub Halt von MSTT, selbsthaltend
SpindleHold	A	BOOL		Spindel Halt von MSTT, selbsthaltend

## Anwahlsignale der MSTT auf die Anwender-Nahtstelle

### Schlüsselschalter

Quelle: MSTT - Schalter	Ziel: Nahtstellen-DB
Stellung 0	DB10.DBX56.4
Stellung 1	DB10.DBX56.5
Stellung 2	DB10.DBX56.6
Stellung 3	DB10.DBX56.7

### Betriebsarten und Maschinenfunktionen

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (Parameter BAGNo)
AUTOMATIC	DB11, ... DBX0.0
MDA	DB11, ... DBX0.1
JOG	DB11, ... DBX0.2
REPOS	DB11, ... DBX1.1
REF	DB11, ... DBX1.2
TEACH IN	DB11, ... DBX1.0
INC 1 ... 10 000, INC Var. (ab SW 5)	DB11, ... DBB2, Bit 0 bis 5

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (Parameter ChanNo)
INC 1 ... 10 000, INC Var. (bis SW 4)	DB21, ... DBB13, Bit 0 bis 5
INC 1 ... 10 000, INC Var. (bis SW 4)	DB21, ... DBB17, Bit 0 bis 5
INC 1 ... 10 000, INC Var. (bis SW 4)	DB21, ... DBB21, Bit 0 bis 5

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (alle Achs-DBs)
INC 1 ... 10 000, INC Var. (bis SW 4)	DB31, ... DBB5, Bit 0 bis 5



### Richtungstasten Eilgangüberlagerung

Die Übertragung ist abhängig von der angewählten Achse. Für nicht angewählte Achsen werden die zugehörigen Nahtstellenbits gelöscht.

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (Parameter ChanNo)
Richtungstaste +	DB21, ... DBX12.7
Richtungstaste -	DB21, ... DBX12.6
Eilgangüberlagerung	DB21, ... DBX12.5
Richtungstaste +	DB21, ... DBX16.7
Richtungstaste -	DB21, ... DBX16.6
Eilgangüberlagerung	DB21, ... DBX16.5
Richtungstaste +	DB21, ... DBX20.7
Richtungstaste -	DB21, ... DBX20.6
Eilgangüberlagerung	DB21, ... DBX20.5

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (alle Achs-DB)
Richtungstaste +	DB31, ... DBX4.7
Richtungstaste -	DB31, ... DBX4.6
Eilgangüberlagerung	DB31, ... DBX4.5

### Override

Quelle: MSTT - Schalter	Ziel: Nahtstellen-DB (Parameter ChanNo)
Vorschuboverride	DB21, ... DBB4

Quelle: MSTT - Schalter	Ziel: Nahtstellen-DB (alle Achs-DB)
Vorschuboverride	DB31, ... DBB0 (angewählte Achsnummer). Durch Vorschuboverride der 1. MSTT werden alle Achsen beaufschlagt.
Spindeloverride	DB31, ... DBB19 (Parameter SpindleIFNo)

### Kanalsignale

Quelle: MSTT - Tasten	Ziel: Nahtstellen-DB (Parameter ChanNo)
NC-Start	DB21, ... DBX7.1
NC-Stop	DB21, ... DBX7.3
RESET	DB21, ... DBX7.7
Einzelstart	DB21, ... DBX0.4

### Vorschub, Spindelsignale

Quelle: MSTT - Tasten	Ziel: Ausgangsparameter vom FC
Vorschub Halt Vorschub Freigabe	Parameter: "FeedHold" speichernd verknüpft, LEDs werden angesteuert
Spindel Halt Spindel Freigabe	Parameter: "SpindleHold" speichernd verknüpft, LEDs werden angesteuert

### Rückmeldungen der Anwender-Nahtstelle zur Ansteuerung von Anzeigen

### Betriebsarten und Maschinenfunktionen

Ziel: MSTT - LED	Quelle: Nahtstellen-DB (Parameter BAGNo)
AUTOMATIC	DB11, ... DBX6.0
MDA	DB11, ... DBX6.1
JOG	DB11, ... DBX6.2
REPOS	DB11, ... DBX7.1
REF	DB11, ... DBX7.2
TEACH IN	DB11, ... DBX7.0

Ziel: MSTT - LED	Quelle: Nahtstellen-DB (Parameter BAGNo)
INC 1..10000, INC Var.	DB11, ... DBB8, Bit0 bis Bit5

## Kanalsignale

Ziel: MSTT - LED	Quelle: Nahtstellen-DB (Parameter ChanNo)
NC-Start	DB21, ... DBX35.0
NC-Stop	DB21, ... DBX35.2 oder DB21, ... DBX35.3
Einzelstart	DB21, ... DBX0.4

---

### Hinweis

LEDs der Richtungstasten werden durch Betätigen der Richtungstasten angesteuert. Achsanwahl- und WKS/MKS-LED werden durch Betätigen der jeweiligen Taster angesteuert.

---

## Aufrufbeispiel

```
CALL FC 19(           //Maschinensteuertafel-M-Variante Signale an Nahtstelle
    BAGNo :=         B#16#1,           //BAG Nr. 1
    ChanNo :=        B#16#1,           //Kanal Nr. 1
    SpindleIFNo :=   B#16#4,           //Spindel Schnittstellen- Nummer = 4
    FeedHold :=      m22.0,           //Vorschub Halt Signal selbsthaltend
    SpindleHold :=   db2.dbx151.0);    //Spindel Halt selbsthaltend in
                                           //Meldungs-DB
```

Mit dieser Parametrierung werden die Signale an die 1. BAG, den 1. Kanal und an alle Achsen übertragen. Zusätzlich wird der Spindeloverride in die 4. Achs/Spindel-Nahtstelle übertragen. Das Vorschub Halt Signal wird an Merker 22.0 und das Spindel Halt Signal an den Datenbaustein DB 2, Datenbit 151.0 übergeben.

## Rangierung der Achsanwahlen

Um eine flexible Zuordnung der Achsanwahltasten zur jeweiligen Achse oder Spindel zu ermöglichen, ist **kein Umschreiben oder Neuschreiben** des FC 19 erforderlich.

Folgender Lösungsvorschlag erfüllt die gewünschte Flexibilität.

1. Vor dem Aufruf des FC 19 wird die Information (VKE) der neu definierten Achswahltaste auf die Anwahl der über die Achsnummer gekennzeichneten Taste gelegt.
2. Nach dem Aufruf des FC 19 wird die Information (VKE) der über die Achsnummer gekennzeichneten LED Information auf die LED der neuen Achswahltaste gelegt und anschließend wird das VKE der bisherigen Achs-LED gelöscht.

### Beispiel:

Die Spindel ist als 4. Achse definiert und soll über die Achstaste 9 angewählt werden.

```
AWL-Ausschnitt

u e                                     //Anwahl 9. Achse
5.2;

= e                                     //auf Anwahl 4. Achse
4.2;

CALL FC 19(                             //Signale an Nahtstelle
    BAGNo :=      B#16#1,                //BAG Nr. 1
    ChanNo :=     B#16#1,                //Kanal Nr. 1
    SpindleIFNo := B#16#4,                //Spindel Schnittstellen-Nummer = 4

    FeedHold :=   m30.0,                 //Vorschub Halt Signal selbsthaltend
    SpindleHold := m30.1);               //Spindel Halt selbsthaltend

u a                                     //LED 4. Achse
2.5;

= a                                     //LED 9. Achse
3.3;

= a                                     //LED 4. Achse ausschalten
2.5;
```

## 2.12.24 FC 21: Transfer Datenaustausch PLC-NCK

### Funktionsbeschreibung

Mit Aufruf des Bausteins Transfer werden Daten zwischen PLC und NCK entsprechend des angewählten Funktionscodes ausgetauscht. Die Daten werden sofort bei Aufruf des FC 21 übertragen, nicht erst am Zyklusbeginn.

Die Aktivierung des Bausteins erfolgt über das "Enable" - Signal.  
Der FC 21 wird nur bei "Enable" = "1" durchlaufen.

Folgende Funktionen für den Datenaustausch zwischen PLC und NCK werden unterstützt:

1. Signale Synchronaktionen an NCK - Kanal
2. Signale Synchronaktionen von NCK - Kanal
3. Schneller Datenaustausch PLC-NCK (Funktion lesen im NCK)
4. Schnellen Datenaustausch PLC-NCK (Funktion schreiben im NCK)
5. Steuerungssignale an NCK - Kanal aktualisieren
6. Steuerungssignale an Achsen aktualisieren (Datenbyte 2 der Anwender Nahtstelle)
7. Steuerungssignale an Achsen aktualisieren (Datenbyte 4 der Anwender Nahtstelle)

### Deklaration der Funktion

#### AWL-Darstellung

```
VAR_INPUT
    Enable :          BOOL;
    Funct  :          BYTE;
    S7Var  :          ANY;
    IVar1  :          INT;
    IVar2  :          INT;
END_VAR
VAR_OUTPUT
    Error  :          BOOL;
    ErrCode :        INT;
END_VAR
```

### Erläuterungen der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "Transfer":

Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		1 = FC 21 aktiv
Funct	E	BYTE	1.. 7	1: Synchronaktionen an Kanal 2: Synchronaktionen von Kanal 3: Daten lesen 4: Daten schreiben 5: Steuersignale an Kanal 6, 7: Steuersignale an Achse
S7Var	E	ANY	S7-Datenbereich	Abhängig von "Funct"
IVAR1	E	INT	0..	Abhängig von "Funct"
IVAR2	E	INT	1..	Abhängig von "Funct"
Error	A	BOOL		
ErrCode	A	INT		Abhängig von "Funct"

### Funktionen

**1: Signale Synchronaktionen an Kanal:**

**2: Signale Synchronaktionen von Kanal:**

Synchronaktionen können von PLC verriegelt bzw. freigegeben werden.  
 Der Datenbereich liegt in der Anwender-Nahtstelle DB21 bis DB30.DBB 300..307 (an Kanal) und DBB 308..315 (von Kanal). Der Parameter "S7Var" wird bei dieser Funktion nicht ausgewertet, muss aber mit einem Aktualparameter belegt werden (siehe Aufrufbeispiel). Die Daten werden sofort während des FC 21-Durchlaufs an/von NC übergeben.

Folgende Signale sind relevant:

Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		1 = FC 21 aktiv
Funct	E	BYTE	1, 2	1: Synchronaktionen an Kanal 2: Synchronaktion von Kanal
S7Var	E	ANY	S7-Datenbereich	nicht genutzt
IVAR1	E	INT	1..MaxKanal	Kanal-Nummer
Error	A	BOOL		
ErrCode	A	INT		1 : "Funct" ungültig 10: Kanal-Nr. ungültig

## Aufrufbeispiel

```
FUNCTION FC 100: VOID
VAR_TEMP
    myAny :          ANY;
END_VAR
BEGIN
NETWORK
...
//Synchronaktionen mit ID3, ID10 und ID31 deaktivieren im NC - Kanal 1:
    SYAK :           AUF DB21;
    SET;
    S DBX300.2;      //ID3
    S DBX301.1;      //ID10
    S DBX303.6;      //ID31
    L B#16#1;
    T MB11;
    SPA TRAN;

//Synchronaktionen von NCK - Kanal:
SYVK :              L B#16#2;
                   T MB11;
TRAN: CALL FC 21 (
                   Enable :=      M 10.0,          // wenn True, FC21 aktiv
                   Funct :=      MB 11,
                   S7Var :=      #myAny,          //nicht genutzt
                   IVAR1 :=      1,                //Kanal-Nr
                   IVAR2 :=      0,
                   Error :=      M 10.1,
                   ErrCode :=     MW 12);
...
END_FUNCTION
```

## Funktionen

### 3, 4: Schneller Datenaustausch PLC-NCK

## Allgemeines

Für den schnellen Austausch von Informationen zwischen PLC und NCK existiert ein eigener interner Datenbereich. Die Dimension des internen Datenfeldes ist auf 1024 Bytes festgelegt. Die Zugriffe (Lesen/Schreiben) von PLC aus erfolgen über FC 21. Die Belegung dieses Bereiches (Struktur) muss im NC-Teileprogramm und PLC-Anwenderprogramm identisch vereinbart werden.

Vom NC-Teileprogramm kann auf diese Daten mit den Befehlen \$A\_DBB[x], \$A\_DBW[x], \$A\_DBD[x], \$A\_DBR[x] zugegriffen werden (siehe /PGA1/ Listenhandbuch Systemvariablen).

Die konkrete Adresse im Datenfeld wird über einen Byte-Offset (0 bis 1023) im Parameter IVAR1 angegeben. Dabei muss die Ausrichtung entsprechend dem Datenformat gewählt werden, d.h. ein Dword beginnt auf einer 4-Byte-Grenze und ein Word auf einer 2-Byte-Grenze. Bytes können auf einem beliebigen Offset innerhalb des Datenfeldes liegen, Einzelbitzugriffe werden nicht unterstützt und vom FC 21 als Bytezugriff umgesetzt. Die Datentypinformation und Anzahl der Daten wird aus dem über S7Var übergebenen ANY-Parameter entnommen.

Die Datenkonsistenz wird nur für 1- und 2-Byte Zugriffe sowohl beim NCK als auch bei der PLC ohne zusätzliche programmiertechnische Maßnahmen sichergestellt. Bei der 2-Byte Konsistenz trifft dies nur beim Datentyp WORD bzw. INT zu, nicht aber beim Datentyp BYTE.

Bei Verwendung größerer Datentypen oder Übertragung von Feldern, die konsistent übertragen werden sollen, muss ein Semaphor-Byte im Parameter IVAR2 angegeben werden, über das der FC 21 die Gültigkeit bzw. Konsistenz eines Blockes erkennen kann. Dieses Handling muss auf der NC-Seite, d.h. im Teileprogramm, durch schreiben bzw. löschen des Semaphor-Bytes unterstützt werden. Das Semaphor-Byte liegt auch in dem gleichen Datenfeld, wie die eigentlichen Nutzdaten.

Ein Wert zwischen 0 und 1023 im IVAR2 bezeichnet das Semaphor-Byte.

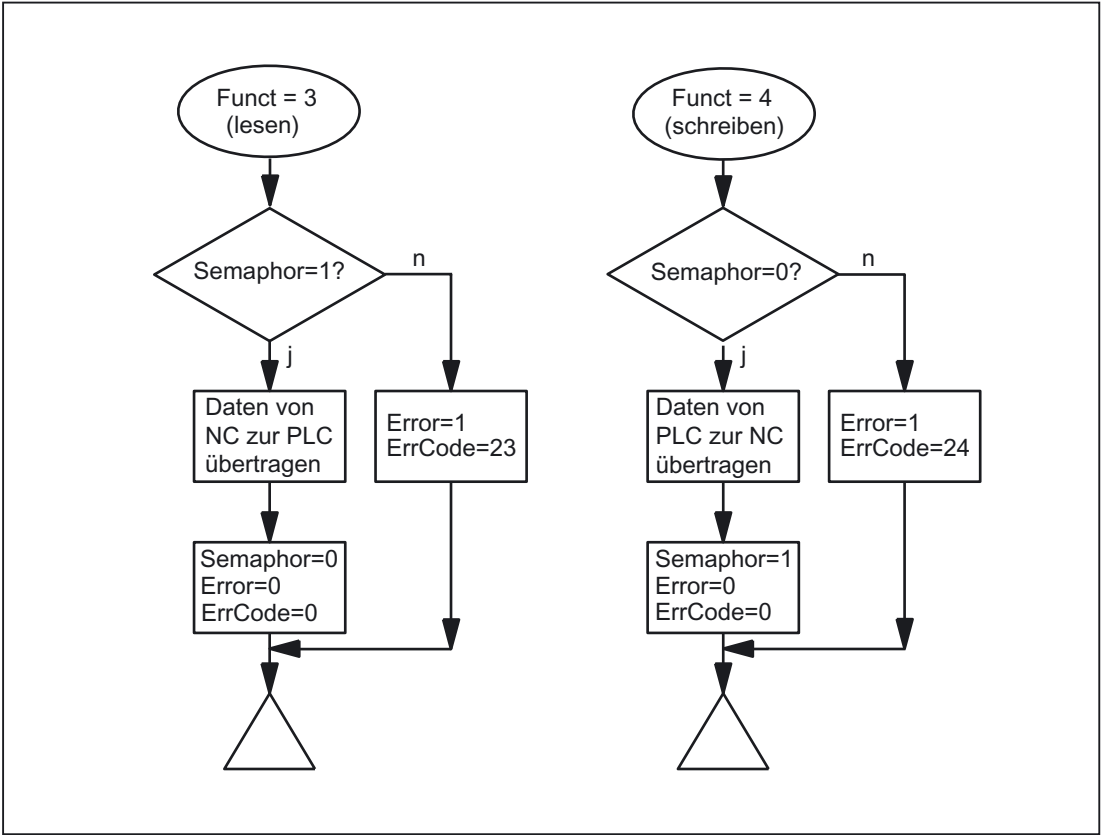
Bei der PLC wird über den FC 21 in dem gleichen Aufruf, in denen die Nutzdaten übertragen werden sollen, das Semaphorbyte gelesen und beschrieben. Der PLC Programmierer muss nur für die Bereitstellung einer Semaphor-Variablen sorgen. Bei dem Zugriff von der NC über das Teileprogramm muss der Semaphor-Mechanismus entsprechend dem unten abgebildeten Flussdiagramm über Einzelanweisungen programmiert werden. Die Sequenz muss für Lesen bzw. Schreiben von Variablen unterschiedlich ausgeführt sein.

Es werden nur Einzelvariable bzw. ARRAYS direkt durch die Semaphortechnik unterstützt. Eine Übertragung von Strukturen ist in Einzelaufträge zu zerteilen. Hierbei ist für die Datenkonsistenz dieser Struktur mit einem, vom Anwender zu programmierenden, Semaphormechanismus selber zu sorgen.

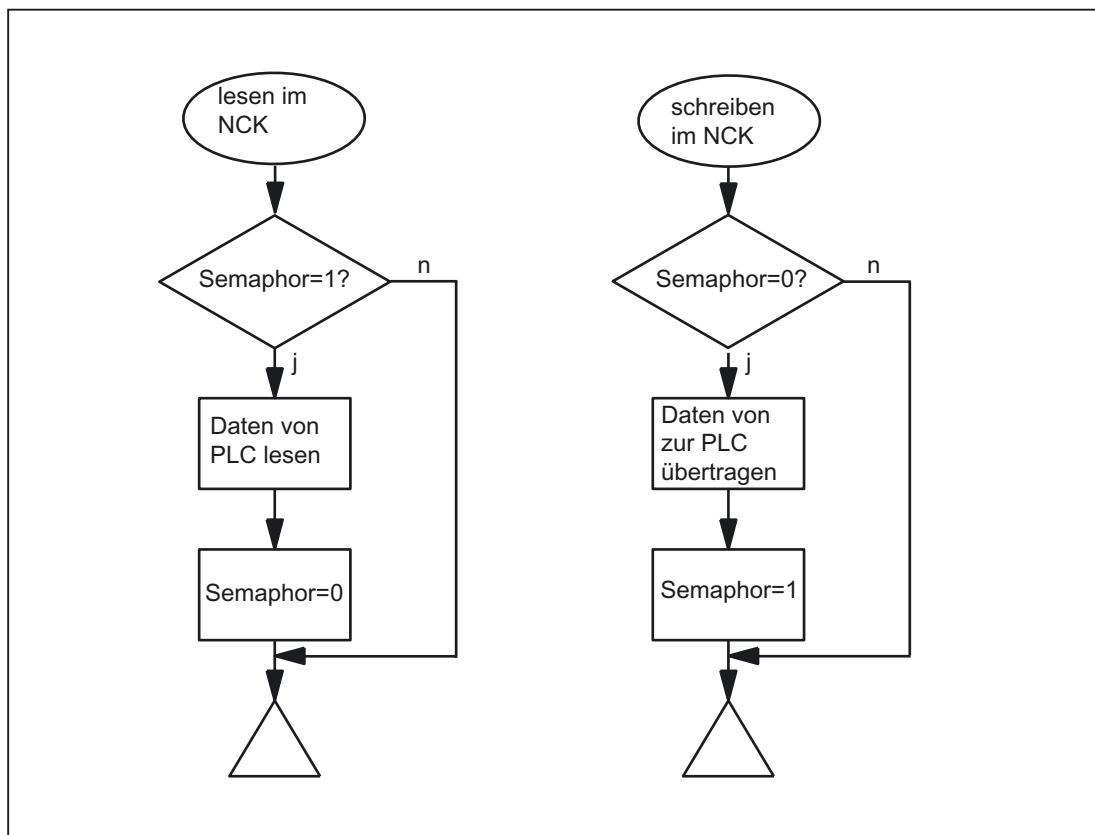
Wird IVAR2 = -1 gesetzt, erfolgt die Datenübertragung ohne Semaphor.

### Datenaustausch mit Semaphor in PLC (Prinzipschaltung des FC 21)





Prinzipieller Aufbau im NCK:



### Variablen-Wertebereiche

Folgende Signale sind relevant:

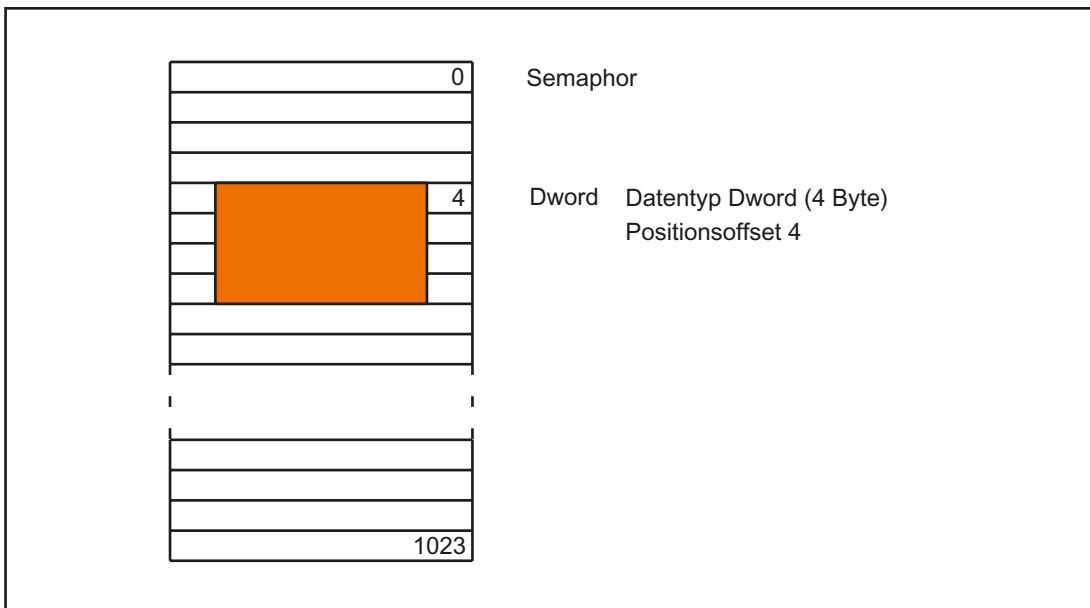
Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		= FC 21 aktiv
Funct	E	BYTE	3 ,4	3: Daten lesen 4: Daten schreiben
S7Var	E	ANY	S7-Datenbereich, außer Lokaldaten	Quell-/Ziel-Datenbereich
IVAR1	E	INT	0..1023	Positionsoffset
IVAR2	E	INT	-1 .. 1023	Semaphor-Byte Transfer ohne Semaphor: -1
Error	A	BOOL		
ErrCode	A	INT		20: Ausrichtungsfehler 21: unerlaubter Positionsoffset 22: unerlaubtes Semaphor-Byte 23: keine neue Daten zu lesen 24: kann keine Daten schreiben 25: Lokaldaten bei S7Var parametriert

### Aufrufbeispiel

**1. Doppelwort vom Positionsoffset 4 mit Semaphore im Byte 0 lesen und im MD100 speichern:**

Datentyp Dword (4 Byte)

Positionsoffset 4



```
CALL FC 21 (
    Enable := M 10.0,           // wenn TRUE, FC 21 aktiv
    Funct  := B#16#3,          // Daten lesen
    S7Var  := P#M 100.0 DWORD 1,
    IVAR1  := 4,
    IVAR2  := 0,
    Error  := M 10.1,
    ErrCode := MW12);
UN M10.1;           // Enable so lange 1, bis Wert gelesen
R M10.0;
```

Beispiele NCK Programmierung aus Synchronaktionen:

Datentransfer von NC zu PLC mit Schreiben der Daten aus Synchronaktionen heraus;  
Byte0 dient als Semaphore

ID=1 WHENEVER \$A\_DBB[0] == 0 DO \$A\_DBR[4] = \$AA\_IM[X] \$A\_DBB[0] = 1

Datentransfer von PLC zu NC mit Lesen der Daten aus Synchronaktionen heraus;  
Byte1 dient als Semaphore

ID=2 WHENEVER \$A\_DBB[1] == 1 DO \$R1 = \$A\_DBR[12] \$A\_DBB[1] = 0

**2. Wort vom Positionsoffset 8 ohne Semaphor lesen und im MW 104 speichern:**

```
CALL FC 21 (
    Enable :=      M 10.0,           //wenn TRUE, FC 21 aktiv
    Funct  :=      B#16#3,          //Daten lesen
    S7Var  :=      P#M 104.0 WORD 1,
    IVAR1  :=      8,
    IVAR2  :=      -1,
    Error  :=      M 10.1,
    ErrCode :=      MW12);
```

**Funktion**

**5: Steuersignale an Kanal aktualisieren:**

Die Funktion 5 dient einer schnellen Übertragung von wichtigen Steuersignalen, zwischen der zyklischen Datenübertragung. Die Datenbytes 6 und 7 der Anwender-Nahtstelle DB21 bis DB30 werden an die NC übertragen. Der Kanal wird im Parameter "IVAR1" angegeben. Hiermit kann z.B. die Vorschubsperr, Einlesesperre außerhalb des PLC Zyklus übertragen werden.

Folgende Signale sind relevant:

Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		1= FC 21 aktiv
Funct	E	BYTE	5	5: Steuersignale an Kanal
S7Var	E	ANY	S7-Datenbereich	nicht genutzt
IVAR1	E	INT	1. MaxKanal	Kanal - Nummer
Error	A	BOOL		
ErrCode	A	INT		1: "Funct" ungültig 10: Kanal - Nr. ungültig

**6: Steuersignale an Achsen aktualisieren:**

Die Funktion 6 dient einer schnellen Übertragung von wichtigen Steuersignalen, zwischen der zyklischen Datenübertragung. Das **Datenbyte 2** der Anwender-Nahtstelle DB31 bis DB61 wird an die NC übertragen. Die Übertragung wird für alle aktivierten Achsen vorgenommen. Hiermit kann z.B. die Reglerfreigabe außerhalb des PLC Zyklus übertragen werden.

Folgende Signale sind relevant:

Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		1= FC 21 aktiv
Funct	E	BYTE	6	6: Steuersignale an Achsen
S7Var	E	ANY	S7-Datenbereich	nicht genutzt
IVAR1	E	INT	0	
Error	A	BOOL		
ErrCode	A	INT		1: "Funct" ungültig

### 7: Steuersignale an Achsen aktualisieren:

Die Funktion 7 dient einer schnellen Übertragung von wichtigen Steuersignalen, zwischen der zyklischen Datenübertragung. Das **Datenbyte 4** der Anwender-Nahtstelle DB31 bis DB61 wird an die NC übertragen. Die Übertragung wird für alle aktivierten Achsen vorgenommen. Hiermit kann z.B. der Vorschubhalt außerhalb des PLC Zyklus übertragen werden.

Folgende Signale sind relevant:

Signal	Art	Typ	Wertebereich	Bemerkung
Enable	E	BOOL		1= FC 21 aktiv
Funct	E	BYTE	7	7: Steuersignale an Achsen
S7Var	E	ANY	S7-Datenbereich	nicht genutzt
IVAR1	E	INT	0	
Error	A	BOOL		
ErrCode	A	INT		1: "Funct" ungültig

## 2.12.25 FC 22: TM\_DIR Richtungsauswahl für Werkzeugverwaltung

### Funktionsbeschreibung

Der Baustein TM\_DIR liefert den kürzesten Weg für die Positionierung eines Magazins oder eines Revolvers anhand der Ist- und Sollposition.

Solange der Eingang **Start** "1-Signal" besitzt, werden alle Ausgangsparameter zyklisch aktualisiert. Hierbei können in darauffolgenden PLC-Zyklen Veränderungen der Eingangsparameter (z.B. Positionswerte) vorgenommen werden.  
Bei 0-Pegel des Startsignals sind die Ausgangssignale undefiniert.

Bei Richtungsauswahl mit Sonderpositionierung - Eingang "Offset" > 0 - wird eine neue Sollposition aus Soll-, Sonderposition, sowie der Anzahl Magazinplätze, nach der Formel berechnet:

Neue Sollpos. = (Sollpos. - (Sonderpos. - 1)) neg. modulo # Plätze

Die neue Sollposition entspricht der Platz-Nr., auf die das Magazin positioniert werden muss, damit die vom Anwender geforderte Sollposition auf der Platz-Nr. der Sonderposition steht. Die Richtungsoptimierung ist sowohl mit als auch ohne Sonderpositionierung aktiv.

Pro Magazin ist der Baustein einmal mit entsprechender Parametrierung aufzurufen.



---

### Warnung

Der Baustein darf nur in Verbindung mit der Werkzeugverwaltung aufgerufen werden bzw. nach dem Einrichten eines Datenbausteins DB 74 wie nachfolgend als Beispiel beschrieben. In diesem Beispiel gibt es zwei Magazine. Das erste Magazin hat 10 Plätze und das zweite hat 12 Plätze. Bei Anpassung an die reale Maschine ist das Datum AnzMag, MagNo[,] zu verändern.

---

```
DATA_BLOCK DB 74
STRUCT
    P: ARRAY [1..9] OF DINT;
    w1 : WORD;
    AnzMag : BYTE;
    res : BYTE;

    MagNo: array [1..16] of struct //Byte 40
        AnzPlatz : INT;
        res1 : BYTE;
        res2 : BYTE;
    end_struct;
end_struct
BEGIN
    P[4]:=L#320; //unbedingt erforderlich !!!
    AnzMag:=b#16#2; //Gesamt-Anzahl Magazine = 2
    MagNo[1].AnzPlatz:=10; //Anzahl Plätze für Magazin 1 = 10
```

```
MagNo[2].AnzPlatz:=12;           //Anzahl Plätze für Magazin 2 = 12
END_DATA_BLOCK
```

---

### Hinweis

Weitere Informationen zur Werkzeugverwaltung (auch in Bezug zur PLC) sind in der Funktionsbeschreibung Werkzeugverwaltung enthalten. Weiterhin stehen noch PI-Dienste für die Werkzeugverwaltung über den FB 4, FC 7 und FC 8 (siehe auch die entsprechende Kapitel in dieser Dokumentation) zur Verfügung.

---

## Deklaration der Funktion

### AWL-Darstellung

```
FUNCTION FC 22 : VOID
//NAME :           TM_DIR
VAR_INPUT
    MagNo :         INT;
    ReqPos :        INT;
    ActPos :        INT;
    Offset :        BYTE;
    Start :         BOOL;
END_VAR
VAR_OUTPUT
    Cw :            BOOL;
    Ccw :           BOOL;
    InPos :         BOOL;
    Diff :          INT;
    Error :         BOOL;
END_VAR
BEGIN
END_FUNCTION
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "TM\_DIR":

Signal	Art	Typ	Wertebereich	Bemerkung
MagNo	E	INT	1..	Magazinnummer
ReqPos	E	INT	1..	Sollplatz
ActPos	E	INT	1..	Istplatz
Offset	E	BYTE	0..	Offset für Sonderpositionierung
Start	E	BOOL		Start der Berechnung
Cw	A	BOOL		1 = Magazin im Uhrzeigersinn bewegen
Ccw	A	BOOL		1 = Magazin gegen den Uhrzeigersinn bewegen
InPos	A	BOOL		1 = Position erreicht
Diff	A	INT	0..	Differenzweg (kürzester Weg)
Error	A	BOOL		1 = Fehler

### Aufrufbeispiel

```
CALL FC 22(                                     //Werkzeugverwaltung Richtungsauswahl
           MagNo :=      2,                       //Magazin Nummer
           ReqPos :=    mw 20,                     //Zielposition
           ActPos :=    mw 22,                     //Aktuelle Position
           Offset :=    b#16#0,                   //Offset für
                                                    Sonderpositionierung
           Start :=     m 30.4,                   //Start-Anstoß
                                                    //Rückgabeparameter
           Cw :=        m 30.0,                     //Magazin im Uhrzeigersinn
                                                    //bewegen
           Ccw :=       m 30.1,                     //Magazin gegen
                                                    Uhrzeigersinn
                                                    //bewegen
           InPos :=     m 30.2,                     //Magazin in Position
           Diff :=      mw 32,                     //Differenzweg
           Error :=     m 30.3                     //Fehler aufgetreten
           );
```



## 2.12.26 FC 24: MCP\_IFM2 Übertragung der MSTT-Signale an die Nahtstelle

### Funktionsbeschreibung

Mit dem FC MCP\_IFM2 (M-Variante schmale Bedientafelfront) werden von der Maschinensteuertafel (MSTT) an die entsprechenden Signale der NCK-/PLC-Nahtstelle übertragen:

- Betriebsarten
- Achsanwahlen
- WKS/MKS Umschaltung
- Verfahrstasten
- Overrides bzw. Overridenachbildung

Im Grundprogramm (FC 2) werden weiterhin die Handradanwahlen, Betriebsarten und weitere Bediensignale von der Bedientafelfront (BT) bzw. vom MMC an die NCK-/PLC-Nahtstelle so übertragen, dass bei den Betriebsarten eine wahlweise Anwahl von der MSTT oder von der BT her möglich ist.

Die Übertragung der MMC-Signale an die Nahtstelle kann durch Setzen des Parameters im FB 1 (DB 7) "MMCToIF" auf den Wert FALSE abgeschaltet werden. Die Ein- und Ausschaltung von "MMCToIF" kann auch im zyklischen Programm durch Setzen oder Rücksetzen (z.B. R gp\_par.MMCToIF) erfolgen.

Für **Vorschub-Override**, **Achsfahr-** und **INC-Tasten** gelten abhängig von der aktiven Betriebsart bzw. vom angewählten Koordinatensystem folgende Festlegungen:

- **Vorschub-Override:**
  - Der Vorschub-Override wird auf die Nahtstelle des angewählten Kanals und auf die Nahtstelle der Achsen transferiert.
  - Die Vorschub-Override-Signale werden zusätzlich zum Nahtstellenbyte "Eilgangkorrektur" (DBB 5) an den NC-Kanal übergeben, wenn das MMC-Signal "Vorschubkorrektur für Eilgang wirksam" gesetzt ist (Ausnahme: Schalterstellung "Null"). Weiterhin wird mit diesem MMC-Signal auch "Eilgangkorrektur wirksam" gesetzt.
- **Maschinenfunktion INC- und Achsfahrtasten:**
  - Bei angewähltem MKS werden die Signale auf die Nahtstelle der angewählten Maschinenachsetransferiert.
  - Bei angewähltem WKS werden die Signale auf die Geo-Achs-Nahtstelle des parametrisierten Kanals transferiert.
  - Bei Umschaltung zwischen MKS und WKS erfolgt generell eine Abwahl der bis dahin angewählten Achse.

Die **Handrad-Anwahlsignale von MMC** werden ausdecodiert und in der zugehörigen Maschinen-Achs-Nahtstelle oder in der Geo-Achs-Nahtstelle des jeweiligen Handrads aktiviert (nur wenn FB 1 Parameter "HWheelMMC":= TRUE).

Die Ansteuerung der zugehörigen LED's der Maschinensteuertafel wird aus der Rückmeldung entsprechender Anwahlen abgeleitet.

Vorschub- und Spindel-Start/Stopp werden nicht an die Nahtstelle übertragen, sondern als Signal "FeedHold" bzw. "SpindleHold" selbsthaltend ausgegeben. Der Anwender kann diese Signale mit weiteren Signalen verknüpfen, die zu Vorschub- oder Spindel-Halt führen sollen (dies kann z.B. über die entsprechenden Eingangssignale des FC 10: AL\_MSG erfolgen). Zusätzlich werden die zugehörigen LEDs mit angesteuert.

Die **Spindelrichtung** (+, -) wird auch nicht direkt geschaltet, sondern als Ausgangsparameter "SpindleDir" zur Verfügung gestellt. Hierüber kann z.B. der FC 18 parametrierbar werden. Zusätzlich wird eine Freigabe der Spindel geschaltet über den Parameter "SpindleHold". Eine Möglichkeit, die Spindel direkt zu bewegen, besteht darin, die Spindel über entsprechende Achsanwahl vorzuwählen und über die (Achsen-) Richtungstasten zu verfahren.

Bei Ausfall der Maschinensteuertafel werden die Signale, die von dieser kommen, mit Null vorbesetzt; ebenso die Ausgangssignale "FeedHold" und "SpindleHold".

In einem PLC-Zyklus sind Mehrfachaufrufe des FC 24 bzw. auch FC 19, FC 25, FC 26 zulässig. Hierbei steuert der erste Aufruf im Zyklus die LED-Anzeigen an. Weiterhin werden im ersten Aufruf alle Aktionen des parametrierbaren Bausteins durchgeführt. Bei den weiteren Aufrufen findet nur noch eine reduzierte Bearbeitung von Kanal und BAG-Schnittstelle statt. Die Geometrieachsen werden nur im ersten Aufruf des Bausteins im Zyklus mit Richtungsvorgaben versorgt.

Die Einzelsatz-An-/Abwahl wird nur durch den ersten Aufruf im Zyklus beeinflusst.

Die zweite Maschinensteuertafel kann bearbeitet werden, wenn der Parameter "BAGNo" um B#16#10 erhöht ist. Bei der Parametrierung ist die BAG Nummer in dem unteren Nibble (untere 4 Bits) enthalten.

"BAGNo" = 0 oder B#16#10 bedeutet keine Bearbeitung der BAG Signale.

ChanNo = 0 bedeutet keine Bearbeitung der Kanal-Signale.

Die INC Anwahlen werden nur noch in die BAG-Schnittstelle übertragen. Hierdurch ergeben sich Laufzeit-Verbesserungen. Die Aktivierung für diese Vorgabe findet über den DB10, ... DBX 57.0 (INC-Eingänge im BAG-Bereich aktiv) durch diesen Baustein einmalig nach Hochlauf statt.

Weiterhin können zwei Maschinensteuertafeln parallel durch diesen Baustein bearbeitet werden. Hierbei ist der Aufruf des Bausteins für die 2. Maschinensteuertafel im OB 1 Zyklus zeitlich hinter den Aufruf für die 1. MSTT zu setzen. Eine Unterstützung von 2 MSTT ist in den Maschinensteuertafel-Bausteinen bis zu bestimmten Grenzen vorhanden (nicht unterstützt werden vom Standard die Achsnummern 10 bis 31, gegenseitige Verriegelungen der Achsanwahlen bei 2 MSTT).

## Flexible Achskonfiguration

Ab dem Softwarestand 6 ist eine Flexibilität bezüglich der Zuordnung Achsanwahlen bzw. Richtungstasten von Maschinen-Achsnummern verfügbar.

Der Einsatz von 2 Maschinensteuertafeln, die gleichzeitig betrieben werden sollen, wird durch die MSTT Bausteine insbesondere für den Anwendungsfall 2 Kanäle, 2 BAGs besser unterstützt. Bei den Achs-Tabellen der jeweiligen MSTT ist zu beachten, dass die Achsnummern auch in der parametrierbaren BAG Nummer des MSTT-Bausteins angegeben sind.

Für diese Flexibilität existieren Tabellen für Achsnummern im DB 10.

Für die 1. Maschinensteuertafel (MSTT) beginnt die Tabelle ab dem Byte 8

(symbolischer Name: MCP1AxisTbl[1..22]) und

für die 2. Maschinensteuertafel (MSTT) ab dem Byte 32

(symbolischer Name: MCP2AxisTbl[1..22]). Hier sind die Maschinen-Achsnummern byteweise einzutragen.

Der Eintrag 0 ist in der Achstabelle zulässig. Eine Prüfung auf eine unzulässige Achsnummer findet nicht statt und kann bei Falscheintrag zum PLC Stopp führen.

Für den **FC 24** kann auch eine Begrenzung der **maximal möglichen Achsanwahlen** erfolgen. Diese Obergrenze wird für die

1. Maschinensteuertafel im DB10.DBW30 (symbolischer Name: MCP1MaxAxis) bzw. für die
2. Maschinensteuertafel DB10.DBW54 (symbolischer Name: MCP2MaxAxis) für die jeweilige MSTT eingestellt werden.

Der voreingestellte Wert ist 0, damit wirkt die konfigurierte maximale Achsanzahl. Die Achsnummern und die Begrenzung können auch dynamisch angepasst werden. Danach muss eine erneute Achsanwahl bei FC 19 erfolgen. Während des Fahrens von Achsen über die jeweiligen Richtungstasten darf keine Umschaltung der Achsnummern erfolgen. Voreingestellt ist der Kompatibilitätsmodus mit den Achsnummern **1 bis 6** für beide MSTT und die Begrenzung auf die konfigurierte Achsanzahl.

## Deklaration der Funktion

```
FUNCTION FC 24: VOID
//NAME :                               MCP_IFM2

VAR_INPUT
    BAGNo :                               BYTE;
    ChanNo :                               BYTE;
    SpindleIFNo :                           BYTE;
END_VAR

VAR_OUTPUT
    FeedHold :                               BOOL;
    SpindleHold :                             BOOL;
    SpindleDir :                               BOOL;
END_VAR

BEGIN
END_FUNCTION
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "MCP\_IFM2":

Signal	Art	Typ	Wertebereich	Bemerkung
BAGNo	E	BYTE	0 - b#16#0A und b#16#10 - b#16#1A	BAG-Nr., in die die Betriebsartensignale übertragen werden. BAGNo >= b#16#10 bedeutet Zugriff auf die 2. Maschinensteuertafel.
ChanNo	E	BYTE	0 - B#16#0A	Kanal-Nr. für die Kanalsignale.
SpindleIFNo	E	BYTE	0 - 31 (B#16#1F)	Nummer der Achsschnittstelle, die als Spindel deklariert ist.
FeedHold	A	BOOL		Vorschub Halt von MSTT, selbsthaltend
SpindleHold	A	BOOL		Spindel Halt von MSTT, selbsthaltend
SpindleDir	A	BOOL		Spindel-Drehrichtung 0 entspricht + (links) 1 entspricht - (rechts)

### Aufrufbeispiel

```
CALL FC 24(                //schmale Maschinensteuertafel-M-Variante
           //Signale an Nahtstelle
           BAGNo :=        B#16#1,           //BAG Nr. 1
           ChanNo :=       B#16#1,           //Kanal Nr. 1
           SpindleIFNo :=  B#16#4,           //Spindel Schnittstellen- Nummer = 4
           FeedHold :=     m22.0,           //Vorschub Halt Signal selbsthaltend
           SpindleHold :=  db2.dbx151.0);    //Spindel Halt selbsthaltend in
                                           //Meldungs-DB
           SpindleDir :=   m22.1);          //Rückgabe Spindelrichtung
```

Mit dieser Parametrierung werden die Signale an die 1. BAG, den 1. Kanal und an alle Achsen übertragen. Zusätzlich wird der Spindeloverride in die 4. Achs/Spindel-Nahtstelle übertragen. Das Vorschub Halt Signal wird an Merker 22.0 und das Spindel Halt Signal an den Datenbaustein DB2, Datenbit 151.0 übergeben. Mit der Rückmeldung der Spindelrichtung über den Parameter SpindleDir kann eine Richtungsvorgabe für einen zusätzlich aufzurufenden FC 18 erfolgen.

## 2.12.27 FC 25: MCP\_IFT Übertragung der MSTT-/BT-Signale an die Nahtstelle

### Funktionsbeschreibung

Mit dem FC MCP\_IFT (T-Variante) werden von der Maschinensteuertafel (MSTT) an die entsprechenden Signale der NCK-/PLC-Nahtstelle übertragen:

- Betriebsarten
- Richtungstasten von vier Achsen
- WKS/MKS-Umschaltung
- Overrides
- Schlüsselschalter

Im Grundprogramm (FC 2) werden weiterhin die Handradanwahlen, Betriebsarten und weitere Bediensignale von der Bedientafelfront (BT) bzw. vom MMC an die NCK-/PLC-Nahtstelle so übertragen, dass bei den Betriebsarten eine wahlweise Anwahl von der MSTT oder von der BT her möglich ist.

Die Übertragung der MMC-Signale an die Nahtstelle kann durch Setzen des Parameters im FB 1 (DB 7) "MMCToIF" durch den Wert FALSE abgeschaltet werden.

Für **Vorschub-Override**, **Achsfahr**- und **INC-Tasten** gilt abhängig von der aktiven Betriebsart bzw. vom angewählten Koordinatensystem folgendes:

- **Vorschub-Override:**
  - Der Vorschub-Override wird auf die Nahtstelle des angewählten Kanals und auf die Nahtstelle der Achsen transferiert.
  - Die Vorschub-Override-Signale werden zusätzlich zum Nahtstellenbyte "Eilgangkorrektur" (DBB 5) an den NC-Kanal übergeben, wenn das MMC-Signal "Vorschubkorrektur für Eilgang wirksam" gesetzt ist (Ausnahme: Schalterstellung "Null"). Weiterhin wird mit diesem MMC-Signal auch "Eilgangkorrektur wirksam" gesetzt.
- **Maschinenfunktion INC- und Achsfahrtasten:**
  - Bei angewähltem MKS werden die Signale auf die Nahtstelle der angewählten Maschinenachsetransferiert.
  - Bei angewähltem WKS werden die Signale auf die Geo-Achs-Nahtstelle des parametrisierten Kanals transferiert.

Die **Handrad-Anwahlsignale von MMC** werden decodiert und in der zugehörigen Maschinen-Achs-Nahtstelle oder in der Geo-Achs-Nahtstelle des jeweiligen Handrads aktiviert (nur wenn im FB 1 der Parameter "HWheelMMC":= TRUE ist).

Die Ansteuerung der zugehörigen LED's der Maschinensteuertafel wird aus der Rückmeldung entsprechender Anwahlen abgeleitet.

Vorschub- und Spindel-Start/Stopp werden nicht an die Nahtstelle übertragen, sondern als Signal "FeedHold" bzw. "SpindleHold" selbsthaltend ausgehen. Der Anwender kann diese Signale mit weiteren Signalen verknüpfen, die zu Vorschub- oder Spindel-Halt führen sollen (dies kann z.B. über die entsprechenden Eingangssignale des FC 10: AL\_MSG erfolgen). Zusätzlich werden die zugehörigen LEDs mit angesteuert.

Bei Ausfall der Maschinensteuertafel werden die Signale, die von dieser kommen, mit Null vorbesetzt; ebenso die Ausgangssignale "FeedHold" und "SpindleHold".

In einem PLC-Zyklus sind Mehrfachaufrufe des FC 25 bzw. auch FC 19, FC 24, FC 26 zulässig. Hierbei steuert der erste Aufruf im Zyklus die LED-Anzeigen an. Weiterhin werden im ersten Aufruf alle Aktionen des parametrisierten Bausteins durchgeführt. Bei den weiteren Aufrufen findet nur noch eine reduzierte Bearbeitung von Kanal und BAG-Schnittstelle statt. Die Geometrieachsen werden nur im ersten Aufruf des Bausteins im Zyklus mit Richtungsvorgaben versorgt.

Die Einzelsatz-An-/Abwahl wird nur durch den ersten Zyklus beeinflusst.

Die zweite Maschinensteuertafel kann bearbeitet werden, wenn der Parameter

"BAGNo" um B#16#10 erhöht ist. Bei der Parametrierung ist die BAG Nummer in dem unteren Nibble (untere 4 Bits) enthalten.

"BAGNo" = 0 oder B#16#10 bedeutet keine Bearbeitung der BAG Signale.

ChanNo = 0 bedeutet keine Bearbeitung der Kanal-Signale.

## Flexible Achskonfiguration

Ab dem Softwarestand 6 ist eine Flexibilität bezüglich der Zuordnung Achsanwahlen bzw. Richtungstasten von Maschinen-Achsnummern verfügbar.

Der Einsatz von 2 Maschinensteuertafeln, die gleichzeitig betrieben werden sollen, wird durch die MSTT Bausteine insbesondere für den Anwendungsfall 2 Kanäle, 2 BAGs besser unterstützt. Hierbei ist der Aufruf des Bausteins für die 2. Maschinensteuertafel im OB1 Zyklus zeitlich hinter den Aufruf für die 1. MSTT zu setzen. Bei den Achs-Tabellen der jeweiligen MSTT ist zu beachten, dass die Achsnummern auch in der parametrisierten BAG Nummer des MSTT-Bausteins angegeben sind.

Für diese Flexibilität existieren Tabellen für Achsnummern im DB 10.

Für die 1. Maschinensteuertafel (MSTT) beginnt die Tabelle ab dem Byte 8

(symbolischer Name: MCP1AxisTbl[1..22]) und

für die 2. Maschinensteuertafel MSTT ab dem Byte 32

(symbolischer Name: MCP2AxisTbl[1..22]). Hier sind die Maschinen-Achsnummern byteweise einzutragen.

Der Eintrag 0 ist in der Achstabelle zulässig. Eine Prüfung auf eine unzulässige Achsnummer findet nicht statt und kann bei Falscheintrag zum PLC Stopp führen.

Die Begrenzung der **möglichen Achsanwahlen bei FC 25** erfolgt über die 0-Werte in der Achstabelle. Die Achsnummern können auch dynamisch angepasst werden. Während dem Fahren von Achsen über die jeweiligen Richtungstasten darf keine Umschaltung der Achsnummern erfolgen.

Voreingestellt ist der Kompatibilitätsmodus mit den Achsnummern **1 bis 4** für beide MSTT und die Begrenzung auf die konfigurierte Achszahl.

---

### Hinweis

Ergänzende Informationen siehe Funktionsbeschreibung des FC 19.

---

## Deklaration der Funktion

```

FUNCTION FC 25: VOID
//NAME :                               MCP_IFT

VAR_INPUT
    BAGNo :                               BYTE;
    ChanNo :                              BYTE;
    SpindleIFNo :                          BYTE;
END_VAR

VAR_OUTPUT
    FeedHold :                             BOOL;
    SpindleHold :                           BOOL;
END_VAR

BEGIN
END_FUNCTION
    
```

## Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "MCP\_IFT":

Signal	Art	Typ	Wertebereich	Bemerkung
BAGNo	E	BYTE	0 - b#16#0A und b#16#10 - b#16#1A	BAG-Nr., in die die Betriebsartensignale übertragen werden. BAGNo >= b#16#10 bedeutet Zugriff auf die 2. Maschinensteuertafel.
ChanNo	E	BYTE	0 - B#16#0A	Kanal-Nr. für die Kanalsignale.
SpindleIFNo	E	BYTE	0 - 31 (B#16#1F)	Nummer der Achsschnittstelle, die als Spindel deklariert ist.
FeedHold	A	BOOL		Vorschub Halt von MSTT, selbsthaltend
SpindleHold	A	BOOL		Spindel Halt von MSTT, selbsthaltend

### Aufrufbeispiel

```
CALL FC 25(           //Maschinensteuertafel-T-Variante
             //Signale an Nahtstelle

    BAGNo :=         B#16#1,           //BAG Nr. 1
    ChanNo :=        B#16#1,           //Kanal Nr. 1
    SpindleIFNo :=   B#16#4,           //Spindel Schnittstellen- Nummer = 4
    FeedHold :=      m22.0,           //Vorschub Halt Signal selbsthaltend
    SpindleHold :=   db2.dbx151.0);    //Spindel Halt selbsthaltend in
                                     //Meldungs-DB
```

Mit dieser Parametrierung werden die Signale an die 1. BAG, den 1. Kanal und an alle Achsen übertragen. Zusätzlich wird der Spindeloverride in die 4. Achs/Spindel-Nahtstelle übertragen. Das Vorschub Halt Signal wird an Merker 22.0 und das Spindel Halt Signal an den Datenbaustein DB2, Datenbit 151.0 übergeben.

## 2.12.28 FC 26: HPU\_MCP Übertragung der PHG / HT6-Signale an die Nahtstelle

### 2.12.28.1 FC 26: HPU\_MCP Übertragung der PHG / HT6-Signale an die Nahtstelle

#### Funktionsbeschreibung

Mit dem FC HPU\_MCP (Maschinensteuertafel-Signale des Handheld Terminals) werden von der Maschinensteuertafel (MSTT) an die entsprechenden Signale der NCK-/PLC-Nahtstelle übertragen:

- Betriebsarten
- WKS/MKS Umschaltung
- Verfahrtasten
- Override

Im Grundprogramm (FC 2) werden weiterhin die Handradanwahlen, Betriebsarten und weitere Bediensignale von der Bedientafelfront (BT) bzw. vom MMC an die NCK-/PLC-Nahtstelle so übertragen, dass bei den Betriebsarten eine wahlweise Anwahl von der MSTT oder von der BT her möglich ist.

Die Übertragung der MMC-Signale an die Nahtstelle kann durch Setzen des Parameters im FB 1 (DB 7) "MMCToIF" auf den Wert FALSE abgeschaltet werden.

Für **Vorschub-Override**, **Achsfahr**- und **INC-Tasten** gelten abhängig von der aktiven Betriebsart bzw. vom angewählten Koordinatensystem folgende Festlegungen:



- **Vorschub-Override:**

- Der Vorschub-Override wird auf die Nahtstelle des angewählten Kanals und auf die Nahtstelle der Achsen transferiert.
- Die Vorschub-Override-Signale werden zusätzlich zum Nahtstellenbyte "Eilgangkorrektur" (DBB 5) an den NC-Kanal übergeben, wenn das MMC-Signal "Vorschubkorrektur für Eilgang wirksam" gesetzt ist (Ausnahme: Schalterstellung "Null"). Weiterhin wird mit diesem MMC Signal auch "Eilgangkorrektur wirksam" gesetzt.

**Maschinenfunktion INC- und Achsfahrtasten:**

- Bei angewähltem MKS werden die Signale auf die Nahtstelle der angewählten Maschinenachse (für 6 Achsen) transferiert.
- Bei angewähltem WKS werden die Signale der ersten drei Achsen auf die Geo-Achs-Nahtstelle des parametrisierten Kanals transferiert. Die restlichen drei Achsen werden auf die Nahtstelle der angewählten Maschinenachsetransferiert.

Die **Handrad-Anwahlsignale von MMC** werden ausdecodiert und in der zugehörigen Maschinen-Achs-Nahtstelle oder in der Geo-Achs-Nahtstelle des jeweiligen Handrads aktiviert (nur wenn im FB 1 der Parameter "HWheelMMC":= TRUE ist).

Die Ansteuerung der zugehörigen LEDs der Maschinensteuertafel wird aus der Rückmeldung entsprechender Anwahlen abgeleitet.

Vorschub- und Spindel-Start/Stopp werden nicht an die Nahtstelle übertragen, sondern als Signal "FeedHold" bzw. "SpindleHold" selbsthaltend ausgegeben. Der Anwender kann diese Signale mit weiteren Signalen verknüpfen, die zu Vorschub- oder Spindel-Halt führen sollen (dies kann z.B. über die entsprechenden Eingangssignale des FC 10: AL\_MSG erfolgen). Zusätzlich werden die zugehörigen LEDs mit angesteuert.

Bei Ausfall der Maschinensteuertafel werden die Signale, die von dieser kommen, mit Null vorbesetzt.

In einem PLC-Zyklus sind Mehrfachaufrufe des FC 26 bzw. auch FC 19, FC 24, FC 25 zulässig. Hierbei steuert der erste Aufruf im Zyklus die LED-Anzeigen an. Weiterhin werden im ersten Aufruf alle Aktionen des parametrisierten Bausteins durchgeführt. Bei den weiteren Aufrufen findet nur noch eine reduzierte Bearbeitung von Kanal und BAG-Schnittstelle statt. Die Geometriechsen werden nur im ersten Aufruf des Bausteins im Zyklus mit Richtungsvorgaben versorgt.

Die Einzelsatz-An-/Abwahl wird nur durch den ersten Zyklus beeinflusst.

Die zweite Maschinensteuertafel kann bearbeitet werden, wenn der Parameter "BAGNo" um B#16#10 erhöht ist. Bei der Parametrierung ist die BAG Nummer in dem unteren Nibble (untere 4 Bits) enthalten.

"BAGNo" = 0 oder B#16#10 bedeutet keine Bearbeitung der BAG Signale.

ChanNo = 0 bedeutet keine Bearbeitung der Kanal-Signale.

Die INC-Anwahlen werden nur noch in die BAG-Schnittstelle übertragen. Hierdurch ergeben sich Laufzeit-Verbesserungen. Die Aktivierung für diese Vorgabe findet über den DB10, ... DBX 57.0 durch diesen Baustein einmalig nach Hochlauf statt. Weiterhin können 2 Maschinensteuertafeln parallel durch diesen Baustein bearbeitet werden. Hierbei ist der Aufruf des Bausteins für die 2. Maschinensteuertafel im OB 1 Zyklus zeitlich hinter den Aufruf für die 1. MTT zu setzen. Eine Unterstützung von 2 MTT ist in den Maschinensteuertafel-Bausteinen bis zu bestimmten Grenzen vorhanden (nicht unterstützt werden vom Standard die Achsnummern 10 bis 31, gegenseitige Verriegelungen der Achsanwahlen bei 2 MTT).

## Flexible Achskonfiguration

Es ist eine Flexibilität bezüglich der Zuordnung Achsanwahlen bzw. Richtungstasten von Maschinen–Achsnummern verfügbar.

Der Einsatz von 2 Maschinensteuertafeln, die gleichzeitig betrieben werden sollen, wird durch die MSTT Bausteine insbesondere für den Anwendungsfall 2 Kanäle, 2 BAGs besser unterstützt. Bei den Achs–Tabellen der jeweiligen MSTT ist zu beachten, dass die Achsnummern auch in der parametrisierten BAG Nummer des MSTT–Bausteins angegeben sind.

Für diese Flexibilität existieren Tabellen für Achsnummern im DB 10.

Für die 1. Maschinensteuertafel (MSTT) beginnt die Tabelle ab dem Byte 8

(symbolischer Name: MCP1AxisTbl[1..22]) und

für die 2. Maschinensteuertafel (MSTT) ab dem Byte 32

(symbolischer Name: MCP2AxisTbl[1..22]). Hier sind die Maschinen–Achsnummern byteweise einzutragen.

Der Eintrag 0 ist in der Achstabelle zulässig. Eine Prüfung auf eine unzulässige Achsnummer findet nicht statt und kann bei Falscheintrag zum PLC Stopp führen. Die Begrenzung der Achszahl bei **FC 26** erfolgt über die 0-Werte in der Achstabelle. Die Achsnummern können auch dynamisch angepasst werden. Während dem Fahren von Achsen über die jeweiligen Richtungstasten darf keine Umschaltung der Achsnummern erfolgen.

Voreingestellt ist der Kompatibilitätsmodus mit den Achsnummern **1 bis 6** für beide MCP und die Begrenzung auf die konfigurierte Achszahl.

---

### Hinweis

Beim FC 26 (PHG und HT6) gibt es die Abweichung gegenüber den anderen MSTT Bausteinen

FC 19, FC 24, FC 25 bei angewähltem Werkstück-Koordinaten System (WKS).

Hier wirken die ersten 3 Verfahrtasten auf die Kanal-Achsen (WKS-Achsen). Die nachfolgenden Verfahrtasten wirken auf die Maschinenachsen gemäß den aus der Achstabelle zugeordneten Achsnummern. Dies sind standardmäßig die Maschinen-Achsen 4 bis 6. Durch Beschalten des Ausgangsbyte 3, Bit 7 auf 1 von den MSTT-Ausgängen (Parameter "MCP1Out" bzw. "MCP2OUT") kann erreicht werden, dass bei eingeschalteten WKS nur die Kanalachsen verfahren werden können

---

## Deklaration der Funktion

```
FUNCTION FC 26: void
//NAME :                               HPU_MCP
VAR_INPUT
    BAGNo :                               BYTE;
    ChanNo :                               BYTE;
END_VAR
BEGIN
END_FUNCTION
```

### Erläuterung der Formalparameter

Die folgende Tabelle zeigt alle Formalparameter der Funktion "HPU\_MCP":

Signal	Art	Typ	Wertebereich	Bemerkung
BAGNo	E	BYTE	0 - b#16#0A und b#16#10 -b#16#1A	BAG-Nr., in die die Betriebsartensignale übertragen werden. BAGNo >= b#16#10 bedeutet Zugriff auf die 2. Maschinensteuertafel.
ChanNo	E	BYTE	0 - B#16#0A	Kanal-Nr. für die Kanalsignale

### 2.12.28.2 Anwahlsignale der MSTT auf die Anwender-Nahtstelle

#### Betriebsarten und Maschinenfunktionen

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (Parameter BAGNo) Darstellung für BAG 1
AUTOMATIC	DB11, ... DBX0.0
MDA	DB11, ... DBX0.1
JOG	DB11, ... DBX0.2
REPOS	DB11, ... DBX1.1
REF	DB11, ... DBX1.2
TEACH IN	DB11, ... DBX1.0
INC 1 ... 10 000, INC Var. (ab SW 5)	DB11, ... DBB2, Bit 0 bis 5

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (Parameter ChanNo)
INC 1 ... 10 000, INC Var. (bis SW 4)	DB21, ... DBB13, Bit 0 bis 5
INC 1 ... 10 000, INC Var. (bis SW 4)	DB21, ... DBB17, Bit 0 bis 5
INC 1 ... 10 000, INC Var. (bis SW 4)	DB21, ... DBB21, Bit 0 bis 5

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (6 Achs-DBs)
INC 1 ... 10 000, INC Var. (bis SW 4)	DB31, ... DBB5, Bit 0 bis 5

### Richtungstasten Eilgangüberlagerung

Die Übertragung ist abhängig von der angewählten Achse. Für nicht angewählte Achsen werden die zugehörigen Nahtstellen-Bits gelöscht.

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (Parameter ChanNo)
Richtungstaste +	DB21, ... DBX12.7
Richtungstaste -	DB21, ... DBX12.6
Eilgangüberlagerung	DB21, ... DBX12.5
Richtungstaste +	DB21, ... DBX16.7
Richtungstaste -	DB21, ... DBX16.6
Eilgangüberlagerung	DB21, ... DBX16.5
Richtungstaste +	DB21, ... DBX20.7
Richtungstaste -	DB21, ... DBX20.6
Eilgangüberlagerung	DB21, ... DBX20.5

Quelle: MSTT - Taster	Ziel: Nahtstellen-DB (6 zugeordnete Achs-DB)
Richtungstaste +	DB31, ... DBX4.7
Richtungstaste -	DB31, ... DBX4.6
Eilgangüberlagerung	DB31, ... DBX4.5

### Override

Quelle: MSTT - Einstellung	Ziel: Nahtstellen-DB (Parameter ChanNo)
Vorschuboverride	DB21, ... DBB4

Quelle: MSTT - Einstellung	Ziel: Nahtstellen-DB (6 zugeordnete Achs-DB)
Vorschuboverride	DB31, ... DBB0 (angewählte Achsnummer). Durch Vorschuboverride der 1. MSTT werden 6 Achsen beaufschlagt.

### Kanalsignale

Quelle: MSTT - Tasten	Ziel: Nahtstellen-DB (Parameter ChanNo)
NC-Start	DB21, ... DBX7.1
NC-Stop	DB21, ... DBX7.3
RESET	DB21, ... DBX7.7
Einzelstart	DB21, ... DBX0.4

### 2.12.28.3 Rückmeldungen der Anwender-Nahtstelle zur Ansteuerung von Anzeigen

#### Betriebsarten und Maschinenfunktionen

Ziel: MSTT - Ausgang	Quelle: Nahtstellen-DB (Parameter BAGNo) Darstellung für BAG 1
AUTOMATIC	DB11, ... DBX7.0
MDA	DB11, ... DBX6.1
JOG	DB11, ... DBX6.2
REPOS	DB11, ... DBX7.1
REF	DB11, ... DBX7.2
TEACH IN	DB11, ... DBX7.0

WKS/MKS - Ausgang wird durch Betätigen der Taste angesteuert.

#### Aufrufbeispiel

```
CALL FC 26(                                     //Maschinensteuertafel des PHG / HT6
                                                //Signale an der Nahtstelle
        BAGNo :=      B#16#1,                 //BAG Nr. 1
        ChanNo :=    B#16#1);                 //Kanal Nr. 1
```

Mit dieser Parametrierung werden die Signale von der ersten parametrisierten Maschinensteuertafel an die 1. BAG, den 1. Kanal und an alle Achsen übertragen.

## 2.12.29 FC 19, FC 24, FC 25, FC 26 Beschreibung Quellcode

### Aufgabe

Maschinensteuertafel an Anwendernahtstelle  
(FC19 M-Variante, FC24 schmale Variante, FC25 T-Variante, FC26 PHG/HT6-Variante).

### Zugehörige Bausteine

DB 7, Anzahl BAG, Kanäle, Achsen  
DB 7, Pointer der Maschinensteuertafeln,  
DB 8, Ablage für den nächsten Zyklus  
FC 20, Ausgabe von Fehlermeldungen

### Verwendete Ressourcen

Keine.

### Allgemeines

Die Bausteine FC 19 (M-Version), FC 24 (schmale Version), FC 25 (T-Version) und FC 26 (PHG/HT6-Version) übertragen die Signale der Maschinensteuertafel von und zur Anwendernahtstelle. In der Eingangsparametrierung wird durch "BAGNo" die BAG ausgewählt, die von dem Baustein bearbeitet wird. Über den Parameter "BAGNo" wird auch die Nummer der Maschinensteuertafel selektiert (Bit4). Bei dem Parameter "ChanNo" wird der Kanal ausgewählt, der bearbeitet wird. Der Parameter "SpindleFNo" definiert die Achsnahtstelle der Spindel. Der Spindeloverride wird auf diese Spindelnahtstelle transferiert. Die Eingangsparameter werden bezüglich Falschparametrierung überprüft.

Die Ausgangsparameter "FeedHold" und "SpindleHold" werden aus den 4 Tasten Vorschub/Spindel-Sperre, Vorschub/Spindel-Freigabe gebildet und mit "logisch 1" bei Sperre zurückgeliefert. Informationen für den nächsten Zyklus werden abhängig von der Maschinensteuertafel-Nummer im DB 8, Byte 0 bis 3 bzw. Byte 62 bis 65 abgespeichert. Diese Informationen sind Flankenmerker, Vorschubwert und angewählte Achsnummer. Die Bausteine werden mit Nutzdaten durch die Pointer-Parameter im DB 7 "MCP1In" und "MCP1Out" ("MCP2In" und "MCP2Out") versorgt. Die Pointer werden indirekt über einen weiteren Zeiger aus dem VAR Teil des DB 7 adressiert um die Absolutadressierung zu vermeiden. Dieser zusätzliche Zeiger ist im FB 1 symbolisch ermittelt worden.

### Bausteinbeschreibung

Alle 4 Bausteine sind ähnlich strukturiert und für die einzelnen Teilaufgaben gegliedert:

Im Netzwerk Eingang werden verschiedene Parameter in lokale Variablen kopiert. Hierbei werden auch die Maschinensteuersignale (Nutzdaten Ein-/Ausgangsbereich) über die verschiedenen Zeiger im DB 7 (gp\_par) umkopiert. Mit diesen lokalen Variablen wird aus Effizienzgründen im Baustein gearbeitet. Für den Anlauf werden einige Werte initialisiert.

Im Netzwerk Global\_ein wird die MKS/WKS Umschaltung mit Flankenbewertung, Achsanwahlen, Richtungstasten und Eilgangüberlagerung für die weitere Bearbeitung im Baustein ermittelt. In diesem Programmteil sollten anwenderspezifische Veränderungen stattfinden, die sich hauptsächlich an den Achsanwahlen orientieren.

Im Netzwerk NC werden nur die Schlüsselschalter-Informationen kopiert.

Das Netzwerk BAG

transferiert die Betriebsarten der Taster als dynamische Signale zum NCK.

Bei der BAG-Nummer 0 wird dieses Netzwerk nicht bearbeitet. Eine zu große BAG-Nummer hat eine Meldung 401901 bzw. 402501 zur Folge und es wird nach Stopp geschaltet.

Im Netzwerk Kanal

werden die Funktionen NC-Start, Stopp, Reset und Einzelsatz werden mit entsprechenden Rückmeldungen aktiviert. Die Richtungstasten der GEO-Achsen werden bei entsprechender Vorwahl versorgt, anderenfalls entsorgt. Bei Kanal-Nummer 0 wird dieses Netzwerk nicht bearbeitet. Eine zu große Nummer hat eine Meldung 401902 bzw. 402502 zur Folge und es wird nach Stopp geschaltet.

Das Netzwerk Spindel

überträgt den Spindeloverride in die über "SpindleIFNo" parametrisierte Nahtstelle.

Das Netzwerk Achsen

überträgt den Vorschuboverride auf die angewählte Achsnahtstelle. Die Richtungstasten werden der angewählten Achse/Spindel zugeordnet. Bei einer vorher angewählten Achse wird die Richtungsinformation auf 0 gesetzt. Die INC Rückmeldung der angewählten Achse wird zur Anzeige gebracht.

Im Netzwerk Global\_aus

werden die Ausgangsparameter aufbereitet und die LED-Signale der INC-Maschinenfunktion gebildet.

Das Netzwerk Ausgang

überträgt die Ausgangssignale der Maschinensteuertafel vom Abbild der VAR\_TEMP auf die logische Adresse. Weiterhin werden die Daten für den nächsten Zyklus gerettet.

## **Erweiterung Achsanwahl**

Im Netzwerk Global\_Ein

ist für mehr als 9 Achsen einzugreifen. Falls hier weitere Tasten und LED der Maschinensteuertafel verwendet werden, sind folgende Maßnahmen zu treffen:

1. Der Befehl UD DW#16#Wert löscht alle definierten LEDs für Achsanwahlen. Derzeit werden mit der Bitmaske genau die 9 Achsanwahl-LED bearbeitet.
2. Der Befehl UW W#16# mit dem Kommentar "Maskieren aller Achsanwahl Tasten" prüft ob eine neue Richtungsanwahl erfolgte. Hier muss die Bitleiste angepasst werden.
3. Die Sprungleiste (SPL) ist mit neuen Sprungmarken zu erweitern. Die neuen Sprungmarken sind vor der Marke m009 einzufügen in abwärtszählender Reihenfolge. Bei den neuen Sprungmarken sind Anwahlinformationen, wie bei Marke m009, m008 dargestellt, zu erweitern.

## 2.13 Signal-/Datenbeschreibungen

### 2.13.1 Nahtstellensignale NCK/PLC, MMC/PLC, MSTT/PLC

#### Allgemeines

Die Nahtstellensignale NCK/PLC, HMI/PLC und MSTT/PLC sind im Listenhandbuch für SINUMERIK 840D aufgelistet, mit Verweisen auf die jeweilige Funktionsbeschreibung, in denen die Signale beschrieben sind.

**Literatur:**

/LIS2/ Listen (Buch 2)

Die NCK-Signale, die vom Grundprogramm ausgewertet und aufbereitet zur Anwendernahtstelle übergeben werden, sind in den folgenden Kapiteln aufgeführt.

### 2.13.2 Dekodierte M-Signale

#### Allgemeines

Die im Teileprogramm, ASUP oder Synchronaktionen programmierten M-Funktionen werden von der NC Kanal-spezifisch an die PLC übertragen:

- M-Funktionen aus Kanal 1: DB21
- M-Funktionen aus Kanal 2: DB22
- usw.

Die Signaldauer beträgt 1 PLC-Zykluszeit.

---

**Hinweis**

Folgende Spindel-spezifischen M-Funktionen werden nicht ausdekodiert: M3, M4, M5 und M70.

---



Adresse in DB21, ...	Variable	Typ	Kommentar
DBX 194.0 ... 7	M_Fkt_M0 ... M7	Bool	M-Signale M0 ... M7
DBX 195.0 ... 7	M_Fkt_M8 ... M15	Bool	M-Signale M8 ... M15
DBX 196.0 ... 7	M_Fkt_M16 ... M23	Bool	M-Signale M16 ... M23
DBX 197.0 ... 7	M_Fkt_M24 ... M31	Bool	M-Signale M24 ... M31
DBX 198.0 ... 7	M_Fkt_M32 ... M39	Bool	M-Signale M32 ... M39
DBX 199.0 ... 7	M_Fkt_M40 ... M47	Bool	M-Signale M40 ... M47
DBX 200.0 ... 7	M_Fkt_M48 ... M55	Bool	M-Signale M48 ... M55
DBX 201.0 ... 7	M_Fkt_M56 ... M63	Bool	M-Signale M56 ... M63
DBX 202.0 ... 7	M_Fkt_M64 ... M71	Bool	M-Signale M64 ... M71
DBX 203.0 ... 7	M_Fkt_M72 ... M79	Bool	M-Signale M72 ... M79
DBX 204.0 ... 7	M_Fkt_M80 ... M87	Bool	M-Signale M80 ... M87
DBX 205.0 ... 7	M_Fkt_M88 ... M95	Bool	M-Signale M88 ... M95
DBX 206.0 ... 3	M_Fkt_M96 ... M99	Bool	M-Signale M96 ... M99

---

#### **Hinweis**

M02/M30 als Hilfsfunktionsausgabe an die PLC sagt nicht aus, dass das Teileprogramm beendet wurde. Um das Ende eines Teileprogramms im Kanal sicher zu ermitteln, muss DB21, ... DBX33.5 (M02/M30 aktiv) ausgewertet werden. Der Kanalzustand muss RESET sein. Die Hilfsfunktionsausgabe könnte aus einem asynchronen Unterprogramm (ASUP) oder einer Synchronaktion stammen und hat in diesen Fällen nichts mit dem tatsächlichen Teileprogrammende zu tun.

---

### 2.13.3 G-Funktionen

#### Allgemeines

Die im Teileprogramm, ASUP oder Synchronaktionen programmierten G-Funktionen werden von der NC Kanal-spezifisch an die PLC übertragen:

- G-Funktionen aus Kanal 1: DB21
- G-Funktionen aus Kanal 2: DB22
- usw.

Die Signaldauer beträgt 1 PLC-Zyklus.

#### POWER ON

Nach POWER ON wird in der NC/PLC-Nahtstelle für alle G-Gruppen der Wert Null, d.h. aktive G-Gruppe undefiniert, vorgegeben.

#### Teileprogramm-Ende bzw. -Abbruch

Nach Teileprogramm-Ende bzw. -Abbruch bleibt der letzte Zustand der G-Gruppe erhalten.

#### NC-START

Nach NC-START werden die Werte der im Maschinendatum:

MD22510 \$NC\_GCODE\_GROUPS\_TO\_PLC

angegebenen 8 G-Gruppen entsprechend der über Maschinendaten festgelegten Grundstellung, sowie der im Teileprogramm programmierten Werte, überschrieben.

Adresse in DB21, ...	Variable	Typ	Grundstellung	Kommentar
DBB 208	G_FKT_GR_1	BYTE	0	Aktive G-Funktion der Gruppe 1
DBB 209	G_FKT_GR_2	BYTE	0	Aktive G-Funktion der Gruppe 2
DBB 210	G_FKT_GR_3	BYTE	0	Aktive G-Funktion der Gruppe 3
DBB 211	G_FKT_GR_4	BYTE	0	Aktive G-Funktion der Gruppe 4
DBB 212	G_FKT_GR_5	BYTE	0	Aktive G-Funktion der Gruppe 5
DBB 213	G_FKT_GR_6	BYTE	0	Aktive G-Funktion der Gruppe 6
DBB 214	G_FKT_GR_7	BYTE	0	Aktive G-Funktion der Gruppe 7
DBB 215	G_FKT_GR_8	BYTE	0	Aktive G-Funktion der Gruppe 8
DBB 216	G_FKT_GR_9	BYTE	0	Aktive G-Funktion der Gruppe 9
DBB 217	G_FKT_GR_10	BYTE	0	Aktive G-Funktion der Gruppe 10
DBB 218	G_FKT_GR_11	BYTE	0	Aktive G-Funktion der Gruppe 11
DBB 219	G_FKT_GR_12	BYTE	0	Aktive G-Funktion der Gruppe 12
DBB 220	G_FKT_GR_13	BYTE	0	Aktive G-Funktion der Gruppe 13
DBB 221	G_FKT_GR_14	BYTE	0	Aktive G-Funktion der Gruppe 14
DBB 222	G_FKT_GR_15	BYTE	0	Aktive G-Funktion der Gruppe 15
DBB 223	G_FKT_GR_16	BYTE	0	Aktive G-Funktion der Gruppe 16
DBB 224	G_FKT_GR_17	BYTE	0	Aktive G-Funktion der Gruppe 17
DBB 225	G_FKT_GR_18	BYTE	0	Aktive G-Funktion der Gruppe 18
DBB 226	G_FKT_GR_19	BYTE	0	Aktive G-Funktion der Gruppe 19
DBB 227	G_FKT_GR_20	BYTE	0	Aktive G-Funktion der Gruppe 20
DBB 228	G_FKT_GR_21	BYTE	0	Aktive G-Funktion der Gruppe 21

Adresse in DB21, ...	Variable	Typ	Grundstellung	Kommentar
DBB 229	G_FKT_GR_22	BYTE	0	Aktive G-Funktion der Gruppe 22
DBB 230	G_FKT_GR_23	BYTE	0	Aktive G-Funktion der Gruppe 23
DBB 231	G_FKT_GR_24	BYTE	0	Aktive G-Funktion der Gruppe 24
DBB 232	G_FKT_GR_25	BYTE	0	Aktive G-Funktion der Gruppe 25
DBB 233	G_FKT_GR_26	BYTE	0	Aktive G-Funktion der Gruppe 26
DBB 234	G_FKT_GR_27	BYTE	0	Aktive G-Funktion der Gruppe 27
DBB 235	G_FKT_GR_28	BYTE	0	Aktive G-Funktion der Gruppe 28
DBB 236	G_FKT_GR_29	BYTE	0	Aktive G-Funktion der Gruppe 29
DBB 237	G_FKT_GR_30	BYTE	0	Aktive G-Funktion der Gruppe 30
DBB 238	G_FKT_GR_31	BYTE	0	Aktive G-Funktion der Gruppe 31
DBB 239	G_FKT_GR_32	BYTE	0	Aktive G-Funktion der Gruppe 32
DBB 240	G_FKT_GR_33	BYTE	0	Aktive G-Funktion der Gruppe 33
DBB 241	G_FKT_GR_34	BYTE	0	Aktive G-Funktion der Gruppe 34
DBB 242	G_FKT_GR_35	BYTE	0	Aktive G-Funktion der Gruppe 35
DBB 243	G_FKT_GR_36	BYTE	0	Aktive G-Funktion der Gruppe 36
DBB 244	G_FKT_GR_37	BYTE	0	Aktive G-Funktion der Gruppe 37
DBB 245	G_FKT_GR_38	BYTE	0	Aktive G-Funktion der Gruppe 38
DBB 246	G_FKT_GR_39	BYTE	0	Aktive G-Funktion der Gruppe 39
...	...	...	...	...
DBB 271	G_FKT_GR_64	BYTE	0	Aktive G-Funktion der Gruppe 64

### G-Funktionen (Werte)

Eine vollständige Auflistung aller G-Funktionen findet sich in:

**Literatur:**

/PG/ Programmierhandbuch Grundlagen, "Liste der G-Funktionen/Wegbedingungen".

## 2.13.4 Meldesignale im DB2

### Allgemeines

Über den DB2 wird dem Anwender die Möglichkeit gegeben, für einzelne Signale Meldungen auf der Bedientafelfront anzuzeigen. Die Signale sind - wie in den Listen der Nahtstellensignale dargestellt - in vordefinierte Gruppen unterteilt. Bei Auftreten oder Verschwinden bzw. Quittieren einer Meldung wird die in der Spalte Meldenummer aufgeführte Nummer zur PCU übertragen. Zu jeder Meldenummer kann ein Text hinterlegt werden.

Literatur:

/IAD/ Inbetriebnahmenleitung; Meldenummern

---

### Hinweis

Die Anzahl der Anwenderbereiche kann über den FB1 parametrierbar werden.

Nach Änderung der Konfiguration (FB1: MsgUser) müssen DB2/DB3 gelöscht werden.

---

### Kanalbereiche im DB2

Bereich	Adresse	Meldenummer
Kanal 1	DBX0.0 - DBX11.7	510.000 -- 510.231
Kanal 1, Geo-Achsen	DBX12.0 - DBX17.7	511.100 - 511.315
Kanal 2	DBX18.0 - DBX29.7	520.000 - 520.231
Kanal 2, Geo-Achsen	DBX30.0 - DBX35.7	521.100 - 521.315
Kanal 3	DBX36.0 - DBX47.7	530.000 - 530.231
Kanal 3, Geo-Achsen	DBX48.0 - DBX53.7	531.000 - 531.315
Kanal 4	DBX54.0 - DBX65.7	540.000 - 540.231
Kanal 4, Geo-Achsen	DBX66.0 - DBX71.7	541.100 - 541.315
Kanal 5	DBX72.0 - DBX83.7	550.000 - 550.231
Kanal 5, Geo-Achsen	DBX84.0 - DBX89.7	551.100 - 551.315
Kanal 6	DBX90.0 - DBX101.7	560.000 - 560.231
Kanal 6, Geo-Achsen	DBX102.0 - DBX107.7	561.100 - 561.315
Kanal 7	DBX108.0 - DBX119.7	570.000 - 570.231
Kanal 7, Geo-Achsen	DBX120.0 - DBX125.7	570.100 - 571.315
Kanal 8	DBX126.0 - DBX137.7	580.000 - 580.231
Kanal 8, Geo-Achsen	DBX138.0 - DBX143.7	581.100 - 581.315
Kanal 9, Kanal 10 derzeit nicht realisiert		

## Anwenderbereiche im DB2

<b>Bereich</b>	<b>Adresse</b>	<b>Meldenummer</b>
Achse/Spindel 1	DBX144.0 - DBX145.7	600.100 - 600.115
Achse/Spindel 2	DBX146.0 - DBX147.7	600.200 - 600.215
Achse/Spindel 3	DBX148.0 - DBX149.7	600.300 - 600.315
Achse/Spindel 4	DBX150.0 - DBX151.7	600.400 - 600.415
Achse/Spindel 5	DBX152.0 - DBX153.7	600.500 - 600.515
Achse/Spindel 6	DBX154.0 - DBX155.7	600.600 - 600.615
Achse/Spindel 7	DBX156.0 - DBX157.7	600.700 - 600.715
Achse/Spindel 8	DBX158.0 - DBX159.7	600.800 - 600.815
Achse/Spindel 9	DBX160.0 - DBX161.7	600.900 - 600.915
Achse/Spindel 10	DBX162.0 - DBX163.7	601.000 - 601.015
Achse/Spindel 11	DBX164.0 - DBX165.7	601.100 - 601.115
Achse/Spindel 12	DBX166.0 - DBX167.7	601.200 - 601.215
Achse/Spindel 13	DBX168.0 - DBX169.7	601.300 - 601.315
Achse/Spindel 14	DBX170.0 - DBX171.7	601.400 - 601.415
Achse/Spindel 15	DBX172.0 - DBX173.7	601.500 - 601.515
Achse/Spindel 16	DBX174.0 - DBX175.7	601.600 - 601.615
Achse/Spindel 17	DBX176.0 - DBX177.7	601.700 - 601.715
Achse/Spindel 18	DBX178.0 - DBX179.7	601.800 - 601.815
Achse 19 bis 31 zur Zeit nicht realisiert		

### Anwenderbereiche im DB2

Bereich	Adresse	Meldenummer
Anwenderbereich 0	DBX180.0 - DBX187.7	700.000 - 700.063
Anwenderbereich 1	DBX188.0 - DBX195.7	700.100 - 700.163
Anwenderbereich 2	DBX196.0 - DBX203.7	700.200 - 700.263
Anwenderbereich 3	DBX204.0 - DBX211.7	700.300 - 700.363
Anwenderbereich 4	DBX212.0 - DBX219.7	700.400 - 700.463
Anwenderbereich 5	DBX220.0 - DBX227.7	700.500 - 700.563
Anwenderbereich 6	DBX228.0 - DBX235.7	700.600 - 700.663
Anwenderbereich 7	DBX236.0 - DBX243.7	700.700 - 700.763
Anwenderbereich 8	DBX244.0 - DBX251.7	700.800 - 700.863
Anwenderbereich 9	DBX252.0 - DBX259.7	700.900 - 700.963
Anwenderbereich 10	DBX260.0 - DBX267.7	710.000 - 701.063
Anwenderbereich 11	DBX268.0 - DBX275.7	710.100 - 701.163
Anwenderbereich 12	DBX276.0 - DBX283.7	710.200 - 701.263
Anwenderbereich 13	DBX284.0 - DBX291.7	710.300 - 701.363
Anwenderbereich 14	DBX292.0 - DBX299.7	710.400 - 701.463
Anwenderbereich 15	DBX300.0 - DBX307.7	710.500 - 701.563
Anwenderbereich 16	DBX308.0 - DBX315.7	710.600 - 701.663
Anwenderbereich 17	DBX316.0 - DBX323.7	710.700 - 701.763
Anwenderbereich 18	DBX324.0 - DBX331.7	710.800 - 701.863
Anwenderbereich 19	DBX332.0 - DBX339.7	710.900 - 701.963
Anwenderbereich 20	DBX340.0 - DBX347.7	702.000 - 702.063
Anwenderbereich 21	DBX348.0 - DBX355.7	702.100 - 702.163
Anwenderbereich 22	DBX356.0 - DBX363.7	702.200 - 702.263
Anwenderbereich 23	DBX364.0 - DBX371.7	702.300 - 702.363
Anwenderbereich 24	DBX372.0 - DBX379.7	702.400 - 702.463

## 2.14 Programmiertipps mit STEP 7

### 2.14.1 Allgemeines

#### Allgemeines

Im Folgenden sollen einige Hinweise gegeben werden, um komplexere Abläufe in STEP 7 zu programmieren. Es handelt sich hierbei im Wesentlichen um den Umgang mit dem Datentyp POINTER bzw. ANY. Grundlegende Hinweise zum Aufbau vom Datentyp POINTER und ANY sind im STEP 7 Handbuch "Entwerfen von Anwenderprogrammen" im Kapitel "Register der CPU und Speichern von Daten" enthalten.

### 2.14.2 Daten kopieren

Für das schnelle Kopieren von Daten von einem DB in einen anderen empfiehlt sich bei

- größeren Datenmengen der Einsatz der Systemfunktionen SFC BLKMOV bzw. SFC FILL, da hier ein wesentlich schnelleres Kopieren stattfinden kann.
- kleineren Datenmengen die untenstehende Routine, da die Versorgung der ANY-Parameter an den SFCs zusätzliche Zeit kostet.

#### Beispiel für das schnelle Kopieren von Daten von einem DB in einen anderen

Code	Kommentar
	// DB xx.[AR1] ist Quelle
	// DI yy.[AR2] ist Ziel
AUF           DB 100;	//Quell DB
LAR1           P#20.0;	//Anfangsadresse Quelle auf Datenbyte 20
AUF           DI 101;	//Ziel DB
LAR2           P#50.0;	//Anfangsadresse Ziel auf Datenbyte 50
	//AR1, AR2, DB, DI vorher geladen
L               4;	//8 Byte transferieren
M001:	
L               DBW [AR1,P#0.0];	//Wortweise kopieren
T               DIW [AR2,P#0.0];	
+AR1           P#2.0;	
+AR2           P#2.0;	
TAK;	
LOOP           M001;	

## 2.14.3 ANY und POINTER

### 2.14.3.1 Variable POINTER bzw. ANY für Übergabe an FC oder FB

In STEP 7 ab der Version 1 besteht die Möglichkeit einen Pointer bzw. einen ANY in der VAR\_TEMP zu definieren. Eine Versorgung eines ANY soll im Folgenden an 2 Beispielen dargestellt werden.

#### Beispiel 1 ANY-Parameter über eine Auswahlliste an einen anderen FB (FC) übergeben

An einem FB (FC) sind **mehrere** ANY-Parameter definiert. Es soll nun über eine Auswahlliste ein bestimmter ANY-Parameter an einen anderen FB (FC) übergeben werden. Dieses funktioniert nur über einen ANY in der VAR\_TEMP. Im Parameter "WelcherAny" kann 1 bis 4 parametrisiert sein, um Addr1 bis Addr4 auszuwählen.

---

#### Hinweis

Im Baustein wird das Adressregister AR2 verwendet. Dieses Adressregister AR2 wird bei auch bei Multi-Instanz DB verwendet. Deshalb darf dieser FB **nicht** als Multi-Instanz DB erklärt sein.

---

FUNCTIONBLOCK FB 100	Kommentar
CODE_VERSION1	//ab STEP 7 Version 2 um Multi-Instanz DB zu deaktivieren
VAR_INPUT	
WelcherAny : INT;	
Addr1 : ANY;	//feste Reihenfolge beachten
Addr2 : ANY;	
Addr3 : ANY;	
Addr4 : ANY;	
END_VAR	
VAR_TEMP	
dbchr : WORD ;	
Anzahl : WORD ;	
typ : BYTE;	
Temp_addr : ANY;	
END_VAR	
BEGIN	
NETWORK	
TITLE =	
L        WelcherAny;	
DEC 1;	
L        P#10.0;	//10 Byte pro ANY



FUNCTIONBLOCK FB 100	Kommentar
*I;	
LAR2;	
L           P##Addr1;	
+AR2;	//Anfangsadresse der ANYs addieren
L           P##Temp_addr;	
LAR1 ;	//Pointer holen aus VAR_TEMP
L           DID [AR2,P#0.0];	//Zeigerwerte übertragen in VAR_TEM
T           LD [AR1,P#0.0];	
L           DID [AR2,P#4.0];	
T           LD [AR1,P#4.0];	
L           DIW [AR2,P#8.0];	
T           LW [AR1,P#8.0];	
CALL FB 101, DB 100	
(ANYPAR := #Temp_addr);	//ANYPAR ist der Datentyp ANY

### Beispiel 2 ein vorher zusammengebauten ANY-Parameter an einen anderen FB (FC) übergeben

Es soll **ein** ANY-Parameter, der vorher zusammengebaut wurde, an einen anderen FB (FC) übergeben werden. Dieses funktioniert nur über einen ANY in der VAR\_TEMP

FUNCTIONBLOCK FB 100	Kommentar
VAR_INPUT	
DBNummer: INT;	
DOffset : INT;	
Datentyp: INT;	
Anzahl: INT;	
END_VAR	
VAR_TEMP	
dbchr : WORD ;	
Temp_addr : ANY;	
END_VAR	
BEGIN	
NETWORK	
TITLE =	
L           P##Temp_addr;	
LAR1 ;	//Pointer holen aus VAR_TEMP
L           B#16#10;	//Kennung ANY
T           LB [AR1,P#0.0];	
L           Datentyp;	
T           LB [AR1,P#1.0];	
L           Anzahl;	
T           LW [AR1,P#2.0];	
L           DBNummer;	

FUNCTIONBLOCK FB 100	Kommentar
T LW [AR1,P#4.0];	
L DBOffset;	
SLD 3;	//Offset ist ein Bitoffset
T LD [AR1,P#6.0];	
CALL FB 101, DB 100	
(ANYPAR := #Temp_addr);	//ANYPAR ist der Datentyp ANY

### 2.14.3.2 Allgemeines

#### Allgemeines

In folgenden Programmbeispielen sollen Programmier-Mechanismen dargestellt werden. Hierbei soll der Zugriff auf Ein-/Aus- und Durchgangvariablen (VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT) vom Datentyp "POINTER" bzw. "ANY" innerhalb eines FC bzw. FB aufgezeigt werden. Die Zugriffe sind so dargestellt, das eine teilsymbolische Programmierform verwendet werden kann.

### 2.14.3.3 Verwendung von POINTER und ANY im FC, wenn POINTER bzw. ANY als Parameter vorliegt

#### Funktionsbeschreibung

Der FC 99 hat Eingangsparameter, die als POINTER bzw. ANY definiert sind.

In dem Beispiel wird ein Rumpfprogramm gezeigt, um auf die Teilkomponenten des POINTER bzw. ANY zuzugreifen. Hierbei wird der über den POINTER, ANY parametrisierte DB aufgeschlagen und der Adressoffset als bereichsübergreifender Zeiger im Adressregister AR1 hinterlegt. Somit kann auf Datenelemente der Variable (in der Regel Strukturen, Arrays) zugegriffen werden, die über den POINTER, ANY adressiert sind.

Dieser Zugriff ist im Beispiel am Ende der jeweiligen Programmsequenz dargestellt. Bei dem Datentyp ANY besteht zusätzlich die Möglichkeit, anhand des Datentyps und der Anzahl von Elementen eine Überprüfung bzw. Verzweigung beim Zugriff der Variablen durchzuführen.

### Beispiel im FC, wenn POINTER bzw. ANY als Parameter vorliegt

FUNCTION FC 99: VOID	Kommentar
VAR_INPUT	
Row : BYTE ;	
Convert : BOOL ;	//Zahlenwandlung aktivieren
Addr : POINTER;	//Zeigt auf Variable
Addr1 : ANY;	
END_VAR	
VAR_TEMP	
dbchr : WORD ;	
Anzahl : WORD ;	
typ : BYTE;	
END_VAR	
BEGIN	
NETWORK	
TITLE =	
	//POINTER
L                P##Addr;	
LAR1 ;	//Pointer holen
L                W [AR1,P#0.0];	//DB-Nummer holen
T                #dbchr;	
L                D [AR1,P#2.0];	//Offsetteil des Pointers
LAR1 ;	
AUF DB [#dbchr];	//Db der Variablen aufschlagen
L B [AR1,P#40.0];	//hole Byte-Wert über Pointer mit dem //Adressoffset 40 //ANY
L                P##Addr1;	
LAR1 ;	//ANY holen
L                B [AR1,P#1.0];	//Typ hole
T                #typ;	
L                W [AR1,P#2.0];	//Anzahl holen
T                #Anzahl;	
L                W [AR1,P#4.0];	//DB-Nummer holen
T                #dbchr;	
L                D [AR1,P#6.0];	//Offsetteil des Pointers
LAR1 ;	
AUF              DB [#dbchr];	//Db der Variablen aufschlagen
L                B [AR1,P#0.0];	//hole Byte-Wert über ANY

### 2.14.3.4 Verwendung von POINTER und ANY im FB, wenn POINTER bzw. ANY als Parameter vorliegt

#### Funktionsbeschreibung

Der FB 99 hat Eingangsparameter, die als POINTER bzw. ANY definiert sind.

In dem Beispiel wird ein Rumpfprogramm gezeigt um auf die Teilkomponenten des POINTER bzw. ANY zuzugreifen. Hierbei wird der über den POINTER, ANY parametrisierte DB aufgeschlagen und der Adressoffset als bereichsübergreifender Zeiger im Adressregister AR1 hinterlegt. Somit kann auf Datenelemente der Variable (in der Regel Strukturen, Arrays) zugegriffen werden, die über den POINTER, ANY adressiert sind.

Dieser Zugriff ist im Beispiel am Ende der jeweiligen Programmsequenz dargestellt. Bei dem Datentyp ANY besteht zusätzlich die Möglichkeit anhand des Datentyps und der Anzahl von Elementen eine Überprüfung bzw. Verzweigung beim Zugriff der Variablen durchzuführen.

#### Beispiel im FB, wenn POINTER bzw. ANY als Parameter vorliegt

FUNCTIONBLOCK FB 99	Kommentar
VAR_INPUT	
Row : BYTE ;	
Convert : BOOL ;	//Zahlenwandlung aktivieren
Addr : POINTER ;	//Zeigt auf Variable
Addr1 : ANY;	
END_VAR	
VAR_TEMP	
dbchr : WORD ;	
Anzahl : WORD ;	
typ : BYTE;	
END_VAR	
BEGIN	
NETWORK	
TITLE =	
	//POINTER
L                    P##Addr;	
LAR1;	//Pointer holen aus Instanz-DB
L                    DIW [AR1,P#0.0];	//DB-Nummer holen
T                    #dbchr;	
L                    DID [AR1,P#2.0];	//Offsetteil des Pointers
LAR1 ;	
AUF                  DB [#dbchr];	//Db der Variablen aufschlagen
L                    B [AR1,P#40.0];	//hole Byte-Wert über Pointer mit dem
//Adressoffset 40	
	//ANY
L                    P##Addr1;	
LAR1 ;	//ANY holen aus Instanz-DB

FUNCTIONBLOCK FB 99	Kommentar
L           DIB [AR1,P#1.0];	//Typ holen
T           #typ;	
L           DIW [AR1,P#2.0];	//Anzahl holen
T           #Anzahl;	
L           DIW [AR1,P#4.0];	//DB-Nummer holen
T           #dbchr;	
L           DID [AR1,P#6.0];	//Offsetteil des Pointers
LAR1 ;	
AUF        DB [#dbchr];	//Db der Variablen aufschlagen
L           B [AR1,P#0.0];	//hole Byte-Wert über ANY

### 2.14.4 Multi-Instanz DB

In STEP 7 ab der Version 2 können FBs Multi-Instanzfähig, d.h. mit Multi-Instanz DBs versehen sein. Multi-Instanz DBs zeichnen sich dadurch aus, dass ein Datenbaustein für verschiedene Instanzen von FBs verwendet werden kann, siehe Dokumentation STEP 7. Somit kann das Mengengerüst der DBs optimiert werden.

Das Aktivieren von Multi-Instanz DBs sollte nur dann erfolgen, wenn diese auch genutzt werden, da sich Laufzeit und Codegröße der FBs erhöhen.

---

#### Hinweis

Bei komplexeren Programmen in FBs, die Zeiger und Adressregister verwenden, sind bei FBs, die Multi-Instanzfähig sein sollen, vom Programmierer bestimmte Regeln einzuhalten. Bei Multi-Instanzen wird die Anfangsadresse der Variablen (VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT, VAR) mit dem DI-Datenbaustein-Register und dem Adressregister AR2 übergeben. Bei Zugriffen innerhalb des Multi-Instanzfähigen FB regelt der Compiler die Zugriffe auf diese Variablen über das Adressregister AR2 selbstständig. Wenn allerdings komplexere Programmteile in diesem FB auch mit Adressregistern arbeiten müssen (z.B. Daten kopieren), dann ist vor dem Ändern des AR2 der bisherige Inhalt zu retten. Das AR2-Register muss vor einem Zugriff auf eine Instanz-Variable (VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT, VAR) wieder mit dem ursprünglichen Inhalt restauriert sein. Das AR2 Register der Instanz ist am sinnvollsten in eine Lokal-Variable (VAR\_TEMP) zu retten. Der Befehl 'Zeiger auf eine Instanzvariable laden' liefert einen Zeigerwert ab dem Beginn der Instanzdaten. Um auf diese Variable über Zeiger zugreifen zu können, ist der Offset, der im AR2 steht, hinzuzurechnen.

---

## Beispiel

FUNCTION_BLOCK FB 99	Kommentar
VAR_INPUT	
varin: INT;	
END_VAR	
VAR	
variable1: ARRAY[0..9] OF INT;	
variable2: INT;	
END_VAR	
BEGIN	
L            P##variable1;	//Pointer auf Anfang des ARRAYS //Im Accu steht jetzt der Wert 8500 0010 //im AR2 steht auch ein Bereichs-übergreifender Zeiger. Wenn Bereichs-übergreifend gearbeitet werden soll, dann ist bei der Addition dieser beiden Zeiger ein Bereich auszublenden.
UD            DW#16#00FF_FFFF,	//Ausblenden des Bereichs
LAR1	//ins AR1 laden
TAR2;	
+AR1 AR2;	//AR2 Instanz Offset hinzuaddieren //Jetzt kann über AR1 indirekt auf das ARRAY der variable1 zugegriffen werden.
L            DIW [AR1, P#0.0];	//z.B. Zugriff auf erstes Element
END_FUNCTION_BLOCK	

## 2.14.5 Strings

### Allgemeines

Der Datentyp STRING wird durch bestimmte Dienste des Grundprogramms benötigt. Aus diesem Grund sollen ein paar zusätzliche Informationen zum Aufbau der Strings und dem generellen Umgang bei Parameterzuweisungen beschrieben werden.

### Aufbau des STRING

Ein Datum mit dem Datentyp STRING ist generell in einem Datenbaustein abgelegt (definiert). Bei der Definition gibt es 2 Varianten:

1. Einer Variablen wird nur der Datentyp STRING zugewiesen. Hierbei generiert der STEP 7-Compiler eine Länge von 254 Zeichen.
2. Einer Variablen wird der Datentyp STRING mit einer Längenangabe in eckigen Klammern zugewiesen (z.B. [32]). Hierbei generiert der STEP 7-Compiler eine String-Länge entsprechend der Vorgabe.

Für eine Variable vom Datentyp STRING werden immer 2 Byte mehr abgelegt, als die Definition ausweist. Hierbei ist in dem 1. Byte die Anzahl der maximal möglichen Zeichen durch den STEP 7 -Compiler abgelegt. Im 2. Byte ist die Anzahl der genutzten Zeichen enthalten. Normalerweise wird hier die Nutzlänge des zugewiesenen STRINGs durch den STEP 7-Compiler abgelegt. Ab dem 3. Byte sind die Zeichen (1 Byte pro Zeichen) abgelegt.

Eine Zuweisung von String-Parametern an Bausteine des Grundprogramms erfolgt generell über den Datentyp POINTER bzw. ANY. Die Zuweisung soll generell über symbolische Programmierung erfolgen. Hierbei ist der Datenbaustein, in dem sich der zu parametrierende String befindet, in die Symbolliste aufzunehmen. Anschließend ist die Zuweisung an den Grundprogramm-Baustein mit symbolischen Namen des Datenbausteins, anschließenden Punkt und symbolischen Namen der String-Variable durchzuführen.

## 2.14.6 Ermittlung von Offsetadressen auf Datenbaustein-Strukturen

### Allgemeines

Eine weitere häufig vorkommende Aufgabe ist die symbolische Ermittlung einer Offsetadresse innerhalb eines strukturierten DBs. In diesem DB liegt z.B. an beliebiger Stelle ein ARRAY oder eine STRUKTUR. Man möchte, nachdem das Adressregister symbolisch mit der Anfangsadresse geladen wurde, über ein Adressregister auf einzelne Elemente des ARRAY oder der STRUKTUR zugreifen. Ein Weg wie man das Adressregister symbolisch laden kann, führt über einen FC, der als Eingangsparameter einen Pointer hat. Diesem Eingangsparameter des FC wird nun im Programm symbolisch die Adresse des ARRAY oder der STRUKTUR zugewiesen. Der Programmcode im FC ermittelt nun die Offsetadresse aus dem Eingangsparameter und übergibt die Offsetadresse im Adressregister (AR1) an die rufende Funktion. Somit ist auch bei indirekten Zugriffen eine Symbolische Adressierung möglich.

FUNCTION FC 99: VOID	Kommentar
VAR_INPUT	
Addr : POINTER ;	//Zeigt auf Variable
END_VAR	
BEGIN	
NETWORK	
TITLE =	
L        P##Addr;	
LAR1 ;	//Pointer holen von Addr
L        D [AR1,P#2.0];	//Offsetteil des Pointers der Variable
LAR1 ;	
END_FUNCTION	



## Randbedingungen

Es sind keine Randbedingungen zu beachten.



## Beispiele

Es sind keine Beispiele vorhanden.



## Datenlisten

### 5.1 Maschinendaten

#### 5.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10100	PLC_CYCLIC_TIMEOUT	Zyklische PLC-Überwachungszeit
14504	MAXNUM_USER_DATA_INT	Anzahl der Anwenderdaten (INT)
14506	MAXNUM_USER_DATA_HEX	Anzahl der Anwenderdaten (HEX)
14508	MAXNUM_USER_DATA_FLOAT	Anzahl der Anwenderdaten (FLOAT)
14510	USER_DATA_INT	Anwenderdatum (INT)
14512	USER_DATA_HEX	Anwenderdatum (HEX)
14514	USER_DATA_FLOAT[n]	Anwenderdatum (FLOAT)

Ein Maschinendatum im Integer-/Hexformat wird in der NC als DWORD gehandhabt.

Ein Maschinendatum im Fließkomma-Format wird in der NC als FLOAT (8 Byte IEEE) gehandhabt.

Sie werden jeweils in der NC/PLC-Nahtstelle abgelegt und können vom PLC-Anwenderprogramm bereits im Hochlauf der PLC aus dem DB 20 gelesen werden.

#### 5.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
28150	MM_NUM_VDIVAR_ELEMENTS	Anzahl Elemente für das Schreiben von PLC-Variablen



# Index

## A

Anlauf und Synchronisation NCK-PLC, 2-43  
ASUPs, 2-49

## B

BAG, 2-44  
Belegung der DBs, 2-71  
Busadressen bei SINUMERIK 840D, 2-40

## D

Diagnosepuffer der PLC, 2-26

## E

Ereignisgesteuerter Signalaustausch, 1-1, 1-2

## F

FB 10  
Sicherheits-Relais, 2-157  
FB 11  
Bremsentest, 2-161  
FB 2  
GET NC-Variable lesen, 2-100  
FB 29  
Diagnose Signalrekorder und Datentrigger, 2-167  
FB 3  
PUT NC-Variable schreiben, 2-108  
FB 4  
Interrupt zuordnen, 2-120  
PI\_SERV Allgemeine PI-Dienste, 2-116  
Werkzeugverwaltungsdienste, 2-125  
FB 5  
GETGUD GUD-Variable lesen, 2-141  
FB 7  
PI\_SERV2 Allgemeine PI Dienste, 2-147  
FB 9  
Bedieneinheitumschaltung, 2-91  
FB 9 MzuN Bedieneinheitumschaltung, 2-151

FC 10  
AL\_MSG, 2-191  
FC 13  
BHGDisp, 2-194  
FC 15  
POS\_AX, 2-198  
FC 16  
PART\_AX, 2-202  
FC 17  
YDelta, 2-206  
FC 18  
SpinCtrl, 2-210  
FC 19  
MCP\_IFM, 2-221  
FC 2  
GP\_HP, 2-171  
FC 22  
TM\_DIR, 2-238  
FC 24  
MCP\_IFM2, 2-241  
FC 25  
MCP\_IFT, 2-245  
FC 3  
GP\_PRAL, 2-173  
FC 7  
TM\_REV, 2-176  
FC 8  
TM\_TRANS, 2-180  
FC 9  
ASUP, 2-188

## K

Konkurrierende Achsen, 2-49

## M

MAXNUM\_USER\_DATA\_FLOAT, 5-1  
MAXNUM\_USER\_DATA\_HEX, 5-1  
MAXNUM\_USER\_DATA\_INT, 5-1  
MD14504, 2-57  
MD14506, 2-57  
MD14508, 2-57  
MD35400, 2-212

M-Dekodierung nach Liste, 2-52  
Meldesignale in DB2, 2-260  
MM\_NUM\_VDIVAR\_ELEMENTS, 5-1

## N

Nahtstelle  
  840D, 2-24  
  PLC/HMI, 2-34  
  PLC/MSTT, 2-37  
Nahtstelle PLC/NCK, 2-28  
NC-Ausfall, 2-46  
NC-Variable lesen/schreiben, 2-50  
NC-Variablen, 2-84  
NC-VAR-Selector, 2-78  
  Inbetriebnahme, Installation, 2-90

## P

PI-Dienste  
  Übersicht, 2-119  
PLC CPUs, Eigenschaften, 2-23  
PLC\_CYCLIC\_TIMEOUT, 5-1  
PLC-Grundprogramm (P3)|Ankopplung der PLC-CPUs, 2-23  
PLC-Grundprogramm (P3)|Physikalische Nahtstellen 840D, 2-24  
PLC-Grundprogramm (P3)|PLC-Nahtstelle bei SINUMERIK 840D, 2-24  
PLC-Meldungen, 2-35  
Programmier- und Parametrierwerkzeuge, 2-76  
Programmiergeräte, 2-76  
Programmiertipps mit STEP 7, 2-263  
  ANY und POINTER, 2-266

Multi-Instanz DB, 2-269  
STRINGs, 2-271  
Variable POINTER bzw. ANY, 2-264  
Verwendung ANY und POINTER im FB, 2-268  
Verwendung ANY und POINTER im FC, 2-266  
Projektierbarkeit von Maschinensteuertafel, Bedienhandgerät, 2-61  
Prozessalarm Bearbeitung, 2-46

## S

Signale  
  PLC/Achsen, Spindeln, 2-33  
  PLC/BAG, 2-31  
  PLC/NC, 2-30  
  PLC/NCK-Kanäle, 2-32  
Speicherbedarf des PLC-Grundprogramms, 2-73  
  maximal, 2-75  
  minimal, 2-75  
Symbolische Programmierung des Anwenderprogramms mit Nahtstellen-DB, 2-51

## U

USER\_DATA\_FLOAT[n], 5-1  
USER\_DATA\_HEX[n], 5-1  
USER\_DATA\_INT[n], 5-1

## Z

Zyklischer Betrieb, 2-43  
Zyklischer Signalaustausch, 1-1