

# SIEMENS

## SIMATIC TDC

### System- und Kommunikations- projektierung D7-SYS

Handbuch

Ausgabe 07/2009  
A5E01115022-04

Vorwort,  
Inhaltsverzeichnis

---

**In wenigen Schritten zum  
ersten Projekt** **1**

---

**Systemsoftware** **2**

---

**Kommunikations-  
projektierung** **3**

---

Index

## Sicherheitstechnische Hinweise

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 Gefährdungstufe 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 Gefährdungstufen 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 können.

### Copyright Siemens AG 2009 All rights reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

Siemens AG  
Automation and Drives  
Geschäftsgebiet Industrial Automation Systems  
Postfach 4848, D- 90327 Nürnberg

### 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.

Siemens AG 2009  
Technische Änderungen bleiben vorbehalten

# Vorwort

## Zweck des Handbuchs

Dieses Handbuch erläutert Ihnen die prinzipielle Nutzung und die Funktionen der Automatisierungssoftware D7-SYS mit dem Schwerpunkt für die entsprechende Technologie- und Antriebsregelungskomponente SIMATIC TDC, FM 458-1 DP, T400 oder SIMADYN D.

TDC: Technology and Drives Control

## Erforderliche Grundkenntnisse

Dieses Handbuch richtet sich an Programmierer und Inbetriebsetzer. Zum Verständnis des Handbuchs sind allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik erforderlich.

## Gültigkeitsbereich des Handbuchs

Dieses Handbuch ist gültig für SIMATIC D7-SYS ab Version 7.1 SP1.

## Einordnung in die Informationslandschaft

Dieses Handbuch ist Bestandteil des Dokumentationspakets für die Technologie- und Antriebsregelungskomponenten T400, FM 458, SIMADYN D, SIMATIC TDC und SIMATIC D7-SYS.

Titel	Inhalt
<b>System- und Kommunikationsprojektierung D7-SYS</b>	<p><b>In wenigen Schritten zum ersten Projekt</b></p> <p>Dieses Kapitel bietet einen sehr einfachen Einstieg in die Methodik des Aufbaus und der Programmierung des Regelsystems SIMATIC TDC/ SIMADYN D. Es ist insbesondere für den Erstanwender interessant.</p> <p><b>Systemsoftware</b></p> <p>Dieses Kapitel vermittelt das grundlegende Wissen über den Aufbau des Betriebssystems und eines Anwendungsprogramms einer CPU. Es sollte genutzt werden, um sich einen Überblick über die Programmiermethodik zu verschaffen und darauf das Design seines Anwenderprogramms aufzubauen.</p> <p><b>Kommunikationsprojektierung</b></p> <p>Dieses Kapitel erläutert Ihnen das grundlegende Wissen über die Kommunikationsmöglichkeiten und wie Sie Kopplungen zu Kommunikations-Partnern projektieren.</p> <p><b>Umstieg von STRUC V4.x auf D7-SYS</b></p> <p>In diesem Kapitel sind wesentliche Merkmale enthalten, die sich mit Einführung von SIMATIC D7-SYS gegenüber STRUC V4.x geändert haben.</p>
<b>D7-SYS – STEP 7, CFC und SFC projektieren</b>	<p><b>Basissoftware</b></p> <p>Dieses Kapitel erläutert Ihnen die prinzipielle Nutzung und die Funktionen der Automatisierungssoftware STEP 7. Als Erstanwender verschafft es Ihnen einen Überblick über die Vorgehensweise beim Konfigurieren, Programmieren und bei der Inbetriebnahme einer Station.</p> <p>Beim Arbeiten mit der Basissoftware können Sie gezielt auf die Online-Hilfe zurückgreifen, die Ihnen Unterstützung zu den Detailfragen der Software-Nutzung bietet.</p> <p><b>CFC</b></p> <p>Die Sprache CFC (Continuous Function Chart) bietet Ihnen die Möglichkeit, graphische Verschaltungen von Bausteinen zu realisieren.</p> <p>Beim Arbeiten mit der jeweiligen Software können Sie zudem die Online-Hilfe nutzen, die Ihnen die Detailfragen zu der Nutzung der Editoren/Compiler beantwortet.</p> <p><b>SFC</b></p> <p>Projektierung von Ablaufsteuerungen mit Hilfe des SFC (Sequential function chart) der SIMATIC S7.</p> <p>Im SFC-Editor erstellen Sie mit grafischen Mitteln den Ablaufplan. Dabei werden die SFC-Elemente des Plans nach festgelegten Regeln platziert.</p>
<b>Hardware</b>	<p>Das gesamte Hardwarespektrum wird als Referenz in diesen Handbüchern beschrieben.</p>

Titel	Inhalt
<b>Funktionsbausteine auswählen</b>	<p>Das Referenzhandbuch gibt Ihnen einen Überblick über alle Funktionsbausteine für die entsprechenden Technologie- und Antriebsregelungskomponenten SIMATIC TDC, FM 458-1 DP, T400 und SIMADYN D.</p> <p><b>Kapitel 1</b> In diesem Kapitel werden die Funktionsbausteine beschrieben die in <b>allen Zielsystemen</b> von SIMATIC D7-SYS projektiert werden können.</p> <p><b>Kapitel 2</b> In diesem Kapitel werden die Funktionsbausteine beschrieben, die <b>nur</b> für SIMATIC TDC projektiert werden können.</p> <p><b>Kapitel 3</b> In diesem Kapitel werden die Funktionsbausteine beschrieben, die <b>nur</b> für die Applikationsbaugruppe FM 458-1 DP projektiert werden können</p> <p><b>Kapitel 4</b> In diesem Kapitel werden die Funktionsbausteine beschrieben, die <b>nur</b> für SIMADYN D und T400 projektiert werden können.</p>

## Wegweiser

Sie sollten das Handbuch als Erstanwender in folgender Weise nutzen:

- Lesen Sie die ersten Kapitel vor Nutzung der Software, um sich mit der Begriffswelt und der prinzipiellen Vorgehensweise vertraut zu machen.
- Nutzen Sie die jeweiligen Kapitel des Handbuchs dann, wenn Sie einen bestimmten Bearbeitungsschritt (z.B. Laden von Programmen) durchführen wollen.

Wenn Sie bereits ein kleines Projekt durchgeführt und dadurch einige Erfahrung gesammelt haben, so können Sie einzelne Kapitel des Handbuchs unabhängig voneinander lesen, um sich über ein Thema zu informieren.

## Besondere Hinweise

Der Benutzerteil dieses Handbuchs enthält keine ausführlichen Anleitungen mit einzelnen Schrittfolgen, sondern soll grundsätzliche Vorgehensweisen verdeutlichen. Genauere Informationen zu den Dialogen der Software und deren Bearbeitung finden Sie jeweils in der Online-Hilfe.

## Trainingscenter

Um Ihnen den Einstieg in das Automatisierungssystem zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in D 90327 Nürnberg.

Internet: <http://www.sitrain.com>

## Service & Support im Internet

Zusätzlich zu unserem Dokumentationsangebot bieten wir Ihnen im Internet (<http://www.siemens.com/automation/service&support>) unser komplettes Wissen online an.

Dort finden Sie:

- den Newsletter, der Sie ständig mit den aktuellsten Informationen zu Ihren Produkten versorgt.
- die für Sie richtigen Dokumente über unsere Suche in Service & Support.
- ein Forum, in welchem Anwender und Spezialisten weltweit Erfahrungen austauschen.
- Ihren Ansprechpartner für Automation & Drives vor Ort über unsere Ansprechpartner-Datenbank.
- Informationen über Vor-Ort-Service, Reparaturen, Ersatzteile und vieles mehr steht für Sie unter dem Begriff "Leistungen" bereit.

# Inhaltsverzeichnis

<b>Vorwort</b> .....	<b>iii</b>
<b>1 In wenigen Schritten zum ersten Projekt</b> .....	<b>1-1</b>
1.1    Voraussetzungen.....	1-2
1.1.1    Software und Hardware.....	1-2
1.1.2    Was Sie erwartet.....	1-4
1.2    Neues Projekt anlegen.....	1-5
1.3    Hardware festlegen.....	1-5
1.4    Erstellen eines CFC-Plans.....	1-6
1.4.1    Einen neuen Plan erzeugen.....	1-6
1.4.2    Einfügen, Parametrieren und Verschalten der Funktionsbausteine.....	1-7
1.5    Prüfen, Übersetzen und Laden des Projekts.....	1-11
1.5.1    Konsistenz des Projekts prüfen und übersetzen.....	1-11
1.5.2    Laden des Anwenderprojekts auf die SIMATIC TDC-CPU-Baugruppe.....	1-11
1.6    Anwenderprojekt testen.....	1-13
1.6.1    Verbindung online trennen.....	1-14
1.6.2    Verbindung online erstellen.....	1-14
1.6.3    Parametrierung online ändern.....	1-14
1.6.4    Baustein online einfügen.....	1-14
1.6.5    Baustein online löschen.....	1-14
1.7    Ergebnisse.....	1-15
1.8    Projekt archivieren.....	1-15
<b>2 Systemsoftware</b> .....	<b>2-1</b>
2.1    Projektierung.....	2-2
2.1.1    Allgemeine Beschreibung.....	2-2
2.1.1.1    Projektierungswerkzeuge.....	2-2
2.1.1.2    Projektierungsschritte.....	2-3
2.1.1.3    Nomenklatur und Bibliotheken.....	2-3
2.1.2    Konfigurieren der Hardware.....	2-4
2.1.2.1    Der erste Schritt: Auswählen der Hardwarebaugruppen.....	2-5
2.1.2.2    Der zweite Schritt: Parametrieren der Hardwarebaugruppen.....	2-6
2.1.2.3    Der dritte Schritt: Überprüfung der Projektierung.....	2-7
2.1.3    Erstellen von CFC-Plänen.....	2-7

2.1.3.1	Der erste Schritt: Auswählen der Funktionsbausteine .....	2-8
2.1.3.2	Der zweite Schritt: Parametrieren und Verschalten der Funktionsbausteine .....	2-8
2.1.3.3	Der dritte Schritt: Übersetzen und Laden des Anwenderprogramms in die CPU.....	2-14
2.1.4	Betriebszustände einer CPU Baugruppe.....	2-15
2.1.5	Beschreibung und Verwendung von Signaltransporten .....	2-16
2.1.5.1	Datenkonsistenz .....	2-17
2.1.5.2	Datenaustausch innerhalb der gleichen Task einer CPU.....	2-17
2.1.5.3	Datenaustausch zwischen verschiedenen Tasks einer CPU .....	2-18
2.1.5.4	Datenaustausch zwischen zyklischen Tasks von mehreren CPUs.....	2-19
2.1.5.5	Datenaustausch zwischen Alarmtasks mehrerer CPU´s.....	2-20
2.1.5.6	Minimierung von Totzeiten.....	2-20
2.1.5.7	Bearbeitungsreihenfolge innerhalb eines CPU-Grundtaktes.....	2-20
2.1.5.8	Verschaltungsänderungen und Grenzanzahl von Verschaltungen .....	2-21
2.1.6	Bedeutung und Einsatzmöglichkeiten des Prozeßabbildes .....	2-22
2.1.6.1	Realisierung des Prozeßabbildes .....	2-23
2.1.6.2	Prozeßabbild bei zyklischen Tasks.....	2-24
2.1.6.3	Prozeßabbild bei Alarmtasks .....	2-26
2.1.7	Bedeutung und Einsatz der CPU-Synchronisation.....	2-26
2.1.7.1	Uhrzeitsynchronisation .....	2-27
2.1.7.2	Synchronisation des eigenen Grundtaktes auf den Grundtakt einer Master-CPU.....	2-27
2.1.7.3	Synchronisation des eigenen Grundtaktes auf Alarmtasks einer Master-CPU... ..	2-27
2.1.7.4	Synchronisation von eigenen Alarmtasks auf Alarmtasks einer Master-CPU.....	2-27
2.1.7.5	Synchronisation von mehreren SIMATIC TDC/SIMADYN D-Stationen .....	2-27
2.1.7.6	Reaktion bei Ausfall der Synchronisation .....	2-28
2.1.7.7	Projektierung der CPU-Grundtakt-Synchronisation.....	2-28
2.1.7.8	Projektierung der Alarmtask-Synchronisation .....	2-29
2.1.7.9	Beispiel einer Synchronisationskonstellation.....	2-30
2.1.8	Bedeutung der Prozessorauslastung .....	2-31
2.1.8.1	Ermittlung der ungefähren Prozessorauslastung .....	2-31
2.1.8.2	Ermittlung der genauen Prozessorauslastung.....	2-31
2.1.8.3	Veranschaulichung der Arbeitsweise des Aufgabenverwalters .....	2-32
2.1.8.4	Beseitigung von Zyklusfehlern.....	2-33
2.1.9	Technische Daten des Betriebssystems .....	2-34
2.1.9.1	Leistungsmerkmale.....	2-34
2.1.9.2	Die Grundfunktionen des Betriebssystems .....	2-36
2.1.9.3	Der Dienst Service.....	2-39
2.2	Funktionsbeschreibung und Benutzerhinweise .....	2-42
2.2.1	Fatale Systemfehler "H" .....	2-42
2.2.2	Hintergrund-Verarbeitung .....	2-45
2.2.2.1	On-Line-Testmodus .....	2-45
2.3	Systemplan @SIMD.....	2-46
<b>3</b>	<b>Kommunikationsprojektierung .....</b>	<b>3-1</b>
3.1	Einführung.....	3-2
3.1.1	Grundlagen der Kommunikation .....	3-2
3.1.1.1	Übersicht Kopplungen.....	3-2
3.1.2	Übersicht Kommunikations-Dienste.....	3-7
3.1.3	Anschlüsse der Kommunikationsbausteine .....	3-7



3.1.3.1	Initialisierungsanschluss CTS.....	3-7
3.1.3.2	Adressanschlüsse AT, AR und US.....	3-8
3.1.3.3	Übertragungsmodus, Anschluss MOD.....	3-10
3.1.3.4	Firmwarestatus, Anschluss ECL, ECO.....	3-14
3.1.3.5	Zustandsanzeige, Ausgang YTS.....	3-14
3.1.4	Funktionsweise der Kopplungen.....	3-15
3.1.4.1	Kopplungs-Zentralbausteine.....	3-16
3.1.4.2	Sender und Empfänger.....	3-17
3.1.4.3	Kompatible Nutzdatenstrukturen.....	3-18
3.1.4.4	Anzahl der Kopplungsbaugruppen in einem Baugruppenträger.....	3-19
3.1.4.5	Reorganisation einer Datenschnittstelle.....	3-20
3.2	Baugruppenträgerlokale Kopplungen.....	3-21
3.2.1	CPU-lokale Kopplung.....	3-21
3.2.2	Direkte CPU-CPU-Kopplung.....	3-22
3.2.3	Koppelspeicher-Kopplung.....	3-23
3.3	Baugruppenträger-Kopplung CP52M0.....	3-24
3.3.1	Anwendungsgebiete.....	3-24
3.3.2	Ein- und Ausschaltverhalten.....	3-25
3.3.3	Synchronisations- und Triggermöglichkeiten.....	3-26
3.3.4	Projektierung.....	3-26
3.3.4.1	Projektierung in HWKonfig.....	3-26
3.3.4.2	Projektierung in CFC.....	3-28
3.3.5	Leistungsdaten.....	3-29
3.3.5.1	Übertragungsraten.....	3-29
3.3.5.2	Leitungslängen.....	3-29
3.3.5.3	Schnittstellenbelegung.....	3-29
3.4	Baugruppenträger-Kopplung CP53M0.....	3-30
3.4.1	Hardware-Aufbau.....	3-33
3.4.2	Leistungsumfang.....	3-33
3.4.3	Reaktion bei "Abschaltung" eines Koppelpartners.....	3-34
3.4.4	Reaktion bei "Zuschaltung" des Master-Baugruppenträgers.....	3-34
3.4.4.1	Quittierung.....	3-34
3.4.5	Wiederanlauffähigkeit.....	3-36
3.4.6	Projektierung.....	3-36
3.4.7	Einschränkungen.....	3-37
3.5	Kopplung TCP/IP (CP51M1).....	3-38
3.5.1	TCP/IP und UDP im Vergleich.....	3-39
3.5.2	Musterkonfiguration.....	3-40
3.5.3	Projektierungsschritte.....	3-40
3.5.3.1	Projektierung im HWKonfig.....	3-40
3.5.3.2	Projektierung mit CFC.....	3-41
3.5.3.2.1	Zentralbaustein @TCPIP.....	3-41
3.5.3.2.2	Empfangsbaustein CRV.....	3-42
3.5.3.2.3	Sendebaustein CTV.....	3-42
3.5.4	Anwendungs-Hinweise.....	3-43
3.5.4.1	Kanalanzahl.....	3-43
3.5.4.2	Telegrammlänge.....	3-44
3.5.4.3	„Ping“ auf CP51M1.....	3-44

3.5.4.4	Performance .....	3-44
3.5.5	Kommunikation mit WinCC .....	3-44
3.5.5.1	Standardkopplung .....	3-45
3.5.5.2	Kopplung über TDC PMC Kanal-DLL .....	3-45
3.5.6	Zentraler Service .....	3-45
3.5.7	Uhrzeitsynchronisation .....	3-45
3.5.8	Umstieg von CP5100 auf CP51M1 .....	3-45
3.6	Kopplung TCP/IP (CP5100) .....	3-47
3.6.1	TCP/IP und UDP im Vergleich .....	3-48
3.6.2	Musterkonfiguration .....	3-49
3.6.3	Projektierungsschritte .....	3-50
3.6.3.1	Projektierung im HWKonfig .....	3-50
3.6.3.2	Projektierung mit CFC .....	3-51
3.6.3.2.1	Zentralbaustein @TCPIP .....	3-51
3.6.3.2.2	Empfangsbaustein CRV .....	3-51
3.6.3.2.3	Sendebaustein CTV .....	3-52
3.6.4	Anwendungs-Hinweise .....	3-53
3.6.4.1	Kanalanzahl .....	3-53
3.6.4.2	Telegrammlänge .....	3-54
3.6.4.3	„Ping“ auf CP5100 .....	3-54
3.6.4.4	Performance .....	3-54
3.7	Kopplung PROFIBUS DP (CP50M1) .....	3-55
3.7.1	Allgemeine Grundlagen .....	3-55
3.7.2	Projektierung .....	3-56
3.7.2.1	Konfigurieren des DP-Systems auf CP50M1 .....	3-56
3.7.2.2	Projektierung der Kommunikation in CFC .....	3-56
3.7.2.3	Projektierung als Slave .....	3-58
3.7.2.4	Shared Input .....	3-58
3.7.3	Äqidistanz .....	3-58
3.7.4	Kommandos SYNC/FREEZE .....	3-58
3.7.4.1	SYNC/FREEZE-Projektierungsvarianten .....	3-59
3.7.5	Inbetriebnahme/Diagnose .....	3-64
3.7.5.1	Funktionsbausteine Diagnose .....	3-64
3.7.5.2	Error-Class (ECL) und Error-Code (ECO) .....	3-67
3.8	Kopplung PROFIBUS DP (CP50M0) .....	3-68
3.8.1	Projektierung mit D7-SYS .....	3-69
3.8.1.1	Kopplungs-Zentralbaustein .....	3-69
3.8.1.2	Adressanschlüsse AT, AR .....	3-70
3.8.1.3	Kommandos SYNC/FREEZE .....	3-71
3.8.1.4	SYNC/FREEZE-Projektierungsvarianten .....	3-72
3.8.1.5	Funktionsbaustein Diagnose .....	3-76
3.8.2	Projektierung mit COM PROFIBUS .....	3-80
3.8.2.1	Abgleich mit CFC-Projektierung .....	3-80
3.8.2.2	CP50M0 als PROFIBUS-Slave .....	3-81
3.8.2.3	Laden der Datenbasis .....	3-82
3.8.3	Inbetriebnahme/Diagnose .....	3-83
3.8.3.1	Leuchtdioden .....	3-83
3.8.3.2	Error-Class (ECL) und Error-Code (ECO) .....	3-84
3.8.3.3	Applikationsbeispiel Kopplung PROFIBUS DP .....	3-86

3.8.3.4	Musterkonfiguration und Systemvoraussetzungen .....	3-86
3.8.3.5	Checkliste der benötigten Hard- und Softwarekomponenten für SIMATIC TDC .....	3-88
3.8.3.6	Projektierung unter STEP 7 CFC .....	3-89
3.8.3.7	Verwendung von Sende- und Empfangsbausteinen .....	3-91
3.8.3.8	Projektierung der Musterkonfiguration in CFC .....	3-92
3.8.3.9	Konfiguration des Kommunikationsmoduls SS52 mit COM PROFIBUS.....	3-95
3.8.3.10	Generierung der COM-Datenbasis mit COM PROFIBUS .....	3-95
3.8.3.11	Download der COM-Datenbasis auf CP50M0.....	3-103
3.8.3.12	Arbeiten mit dem Downloadtool „SS52load“.....	3-104
3.8.3.13	Verhalten der CP50M0 während und nach dem Download .....	3-104
3.9	Kopplung MPI .....	3-105
3.9.1	Eigenschaften und Hardware .....	3-105
3.9.2	Projektierung.....	3-105
3.10	Tabellenfunktion .....	3-106
3.10.1	Einleitung .....	3-106
3.10.1.1	Übersicht „Handbetrieb“.....	3-107
3.10.1.2	Übersicht „Automatikbetrieb: Kommunikation“ .....	3-107
3.10.1.3	Übersicht „Automatikbetrieb: Speicherkarte .....	3-109
3.10.1.4	Funktionsbaustein WR_TAB .....	3-109
3.10.2	Handbetrieb .....	3-111
3.10.2.1	Anwendung .....	3-111
3.10.2.2	Projektierung.....	3-112
3.10.3	Automatikbetrieb: Kommunikation.....	3-113
3.10.3.1	Anwendung mit S7-Steuerung und SIMATIC FM 458 Applikationsbaugruppe.....	3-113
3.10.3.2	Projektierung für S7-Steuerung und Applikationsbaugruppe FM 458.....	3-115
3.10.3.3	Tabellenwerte in Datenbaustein einfügen .....	3-116
3.10.3.3.1	Tabellenwerte manuell eingeben.....	3-117
3.10.3.3.2	Tabellenwerte importieren .....	3-121
3.10.3.3.3	Nachladen von Tabellenwerten in einen DB .....	3-132
3.10.3.4	Aufbau des Datentelegramms bei TCP/IP- oder DUST1-Verbindung .....	3-134
3.10.4	Automatikbetrieb: Speicherkarte .....	3-135
3.10.4.1	Erstellung einer Tabellendatei im csv-Format .....	3-135
3.10.4.2	Arbeiten mit dem D7-SYS additionalComponentBuilder .....	3-137
3.10.4.3	Laden .....	3-141
3.10.4.4	Projektierung der Funktionsbausteine .....	3-142
3.11	Kommunikations-Dienst Meldesystem .....	3-144
3.11.1	Eintragslogik der Meldeeintragsbausteine .....	3-144
3.11.1.1	Meldeeintragsbausteine für eine kommende Meldung.....	3-144
3.11.1.2	Meldeeintragsbausteine für eine kommende und eine gehende Meldung .....	3-145
3.11.2	Projektierungsbeispiel für Meldesystem .....	3-145
3.11.3	Ausgabeformate des Meldeauswertebausteins MSI .....	3-149
3.11.3.1	Aufbau einer Fehler- oder Warnmeldung .....	3-149
3.11.3.2	Übersicht der Meldungsformate .....	3-149
3.11.3.3	Aufbau einer Überlaufmeldung.....	3-151
3.11.3.4	Aufbau einer Kommunikationsfehlermeldung.....	3-151
3.11.3.5	Aufbau einer Systemfehlermeldung .....	3-152
3.11.3.6	Detaillierte Beschreibung der Meldungsformate des Funktionsbausteins MSI.....	3-152
3.11.3.7	Ausgabeformat des Meldeauswertebausteins MSIPRI .....	3-156
3.12	Kommunikations-Dienst Prozessdaten.....	3-159
3.12.1	Empfangs- und Sendebausteine .....	3-159

3.12.1.1	Virtuelle Verbindungen.....	3-159
3.12.1.2	Anschlüsse der Bausteine CRV, CTV .....	3-163
3.12.2	Kanal-Rangierbausteine CCC4 und CDC4 .....	3-163
3.12.2.1	Sammelbaustein CCC4 .....	3-164
3.12.2.2	Verteilerbaustein CDC4 .....	3-165
3.12.2.3	Kompatible Nutzdatenstruktur .....	3-166
3.12.3	Diagnoseausgänge .....	3-166
3.12.3.1	Störungsursache .....	3-166
3.12.3.2	Kanalzuordnung .....	3-167
3.12.3.3	Kanalzustände .....	3-168
3.12.4	Einführung „Zeiger-basierte Kommunikationsbausteine .....	3-168
3.12.4.1	Prinzipielle Funktionsweise.....	3-169
3.12.4.2	Anwendungen .....	3-169
3.12.4.3	Merkmale der zeiger-basierten Kommunikation .....	3-169
3.12.4.4	Zugehörige Funktionsbausteine .....	3-170
3.12.4.5	Zeiger-Schnittstelle .....	3-171
3.12.4.6	Projektierungshinweise .....	3-171
3.12.4.7	Beispiele als CFC-Screenshots .....	3-173
3.13	Kommunikations-Dienst Service .....	3-178
3.13.1	Funktionsbaustein SER .....	3-179
3.13.2	Systembelastung, Antwortzeiten .....	3-180
3.14	Kommunikations-Dienst Uhrzeitsynchronisation .....	3-181
3.15	Kommunikation mit SIMATIC Operator Panels .....	3-183
3.15.1	Musterkonfiguration .....	3-184
3.15.2	Projektieren von SIMATIC TDC.....	3-184
3.15.2.1	Auswahl der Komponenten im HWKonfig .....	3-184
3.15.2.2	Projektieren mit CFC.....	3-185
3.15.2.2.1	Initialisierung des OP7 .....	3-185
3.15.2.2.2	Lesen von Funktionsbaustein-Anschlüssen .....	3-185
3.15.2.2.3	Schreiben von Funktionsbaustein-Anschlüssen.....	3-186
3.15.2.2.4	Projektierung von Betriebsmeldungen.....	3-187
3.15.2.2.5	Projektierung von Störmeldungen .....	3-187
3.15.2.2.6	Projektierung der Funktionstastatur.....	3-188
3.15.2.2.7	Projektierung des Schnittstellenbereichs.....	3-188
3.15.2.3	Importieren der Symboltabelle.....	3-189
3.15.3	Projektierung des OP7 mit ProTool/Lite .....	3-189
3.15.4	Anwendungs-Hinweise .....	3-190
3.15.4.1	Rechenzeiten .....	3-190
3.16	WinCC- Anbindung an SIMATIC TDC über Standard-Kanal (SIMATIC S7 Protocol Suite.CHN).....	3-191
3.16.1	Kopplung über TCP/IP mit „BuB“- Funktionen.....	3-192
3.16.1.1	Projektierung der kopplungsrelevanten TDC- Hardware.....	3-192
3.16.1.2	CFC-Projektierung .....	3-193
3.16.1.2.1	Projektierung der kopplungsrelevanten CFC-Funktionsbausteine .....	3-193
3.16.1.2.2	Markierung der Funktionsbausteinanschlüsse in den CFC-Plänen und Erzeugung des Adressbuches.....	3-195
3.16.1.3	WinCC- Projektierung .....	3-201
3.16.2	Projektierungsvariante „S7DB“ .....	3-207
3.16.3	Kopplungsvarianten MPI und PROFIBUS DP .....	3-209
3.16.3.1	Hardwarekonfiguration.....	3-209

3.16.3.2	CFC-Projektierung .....	3-214
3.16.3.3	WinCC-Projektierung .....	3-215
3.16.4	Projektierung mit dem „D7-SYS-OS-Engineering-Tool“ .....	3-217
3.17	Kommunikation mit WinCC (TCP/IP).....	3-225
3.17.1	Voraussetzungen .....	3-225
3.17.2	Prozessvariablen .....	3-226
3.17.2.1	SIMATIC TDC-Projektierung .....	3-226
3.17.2.2	WinCC-Projektierung .....	3-228
3.17.3	Bitmeldeverfahren.....	3-229
3.17.4	SIMATIC TDC-Meldungen.....	3-229
3.17.4.1	SIMATIC TDC-Projektierung .....	3-229
3.17.4.2	WinCC-Projektierung .....	3-231
3.17.5	Adressbucherzeugung mit CFC-Editor.....	3-231
3.17.6	Adresslistenimportwerkzeug ADRIMP.....	3-232
3.17.6.1	Voraussetzungen .....	3-232
3.17.6.1.1	Erzeugen der Variablendefinitionsdatei.....	3-232
3.17.6.1.2	Erzeugen und Importieren einer neuen Signalliste .....	3-232
3.17.6.1.3	Importieren einer vorhandenen Signalliste.....	3-233
3.17.6.2	Überprüfen des erzeugten Datenhaushaltes in WinCC .....	3-233
3.17.7	Verbindungsaufbau SIMATIC TDC-WinCC.....	3-234
3.17.7.1	WinCC aktivieren .....	3-234
3.17.7.2	SIMATIC TDC aktivieren .....	3-234
3.18	Kommunikations-Dienst Trace .....	3-235
3.18.1	Einfach-Trace .....	3-235
3.18.1.1	Arbeitsweise des @TCP.....	3-235
3.18.1.2	Arbeitsweise der Erfassungsbausteine .....	3-237
3.18.1.3	Arbeitsweise des Header-Bausteins TRHI.....	3-238
3.18.1.4	Einfach-Trace Projektierung .....	3-239
3.18.1.5	Antworttelegramme.....	3-242
 <b>Index .....</b>		 <b>I-1</b>



# 1 In wenigen Schritten zum ersten Projekt

<b>Kapitelübersicht</b>	1.1	Voraussetzungen	1-2
	1.2	Neues Projekt anlegen	1-5
	1.3	Hardware festlegen	1-5
	1.4	Erstellen eines CFC-Plans	1-6
	1.5	Prüfen, Übersetzen und Laden des Projekts	1-11
	1.6	Anwenderprojekt testen	1-13
	1.7	Ergebnisse	1-15
	1.8	Projekt archivieren	1-15

## 1.1 Voraussetzungen

**Einführung** Diese Kurzanleitung ist für Einsteiger gedacht und zeigt die grundsätzliche Vorgehensweise bei der Projekterstellung.

Genauere Informationen zu den Dialogen der Entwicklungs-Software und deren Bearbeitung finden Sie in der entsprechenden Online-Hilfe.

### 1.1.1 Software und Hardware

**Software** Auf Ihrem PG/PC mit Windows 95/98/ME/NT 4.0/2000 müssen die drei Softwarepakete

- STEP 7
- CFC
- D7-SYS




in genau dieser Reihenfolge installiert sein. Für STEP7 und CFC ist eine Autorisierung erforderlich.

#### HINWEIS

Die aktuellen Installations- und Benutzerhinweise finden Sie in den jeweiligen „Liesmich“-Dateien. Versionsabhängigkeiten beachten!

Während der STEP7 Installation werden Sie nach der Online-Schnittstelle gefragt, müssen aber für SIMATIC TDC nichts auswählen und installieren. (Fenster „Schließen“ und nachfolgendes Fenster mit „OK“ beenden.)

**Hardware** Für das Beispielprojekt „My First Project“ benötigen Sie folgende Hardware-Komponenten:

Komponente	Funktion	Abbildung/Bestellnummer
<b>Baugruppenträger UR5213</b> mit Stromversorgung 21 Steckplätze, 64 Bit Bus, Lüfter, AC 115/230 V	... ist der Baugruppenträger für eine SIMATIC TDC-Station. ... dient der mechanischen Aufnahme der Baugruppen und versorgt sie mit Strom.	 <b>6DD1682-0CH0</b>
<b>CPU-Baugruppe CPU551</b> (auf Steckplatz 1) 64 Bit, 266 MHz, 32 MByte SD-RAM, 8 Binäreingänge davon 4 alarmfähig	... führt das Anwenderprogramm aus. ... tauscht über die Rückwandleiterplatte des Baugruppenträgers mit anderen Baugruppen Daten aus. ... kommuniziert über die serielle Schnittstelle mit einem PG/PC.	 <b>6DD1600-0BA1</b>
<b>Programmspeichermodul MC521</b> 2 MByte Anwenderprogramm-speicher, 8 kByte Änderungsspeicher	... speichert das Betriebssystem, das Anwenderprogramm und die Online-Änderungen.	 <b>6DD1610-0AH3</b>



<p><b>Servicekabel SC 67</b></p>	<p>... verbindet die CPU-Baugruppe mit dem PG/PC.</p>	 <p><b>6DD1684-0GH0</b></p>
<p><b>Signalbaugruppe SM500</b> (auf Steckplatz 2) 16 BA, 16 BE, 8AE, 4AE integrierend, 8AA, 4 Impulsgeber-Eingänge, 4 Absolutwertgeber-Eingänge</p>	<p>... stellt analoge und digitale Ein-/Ausgänge sowie Inkremental- und Absolutwertgeberanschlüsse zur Verfügung.</p>	 <p><b>6DD1640-0AH0</b></p>
<p><b>Interfacekabel SC 62</b> Länge: 2 m</p>	<p>... verbindet die Ein-/Ausgänge der SM500-Baugruppe mit bis zu 5 Interfacemodulen SBxx oder SU12.</p>	 <p><b>6DD1684-0GC0</b></p>
<p><b>Interfacemodul SB10</b> 2 x 8 Schraubklemmen, LED-Anzeigen</p>	<p>... bietet Ihnen die Möglichkeit, das Anwenderprogramm während der Inbetriebnahme und im Betrieb zu testen, da die Zustände der Digitalausgänge über Leuchtdioden (LED) angezeigt werden.</p>	 <p><b>6DD1681-0AE2</b></p>

Bild 1-1 Baugruppenliste zum Beispielprojekt „My First Project“

**HINWEIS**

Technische Daten stehen im „SIMATIC TDC-Hardware-Handbuch“ und weitere Bestell-Informationen finden Sie in „Ergänzungen zum Katalog DA99.1999“.

## 1.1.2 Was Sie erwartet

**Von der Aufgabe zum ersten Projekt** Das Beispiel „My First Project“ führt Sie Schritt für Schritt zum lauffähigen Projekt.

### 1. Aufgabenstellung analysieren

Dadurch wissen Sie, welche Funktionsbausteine, welche Ein- und Ausgänge Sie benötigen und welche Hardware dafür in Frage kommt:

### 2. Hardware festlegen

Diese Hardware-Informationen werden Sie in STEP7 verwenden, um die Baugruppen einzutragen und Ihre Eigenschaften festzulegen.

### 3. Projektieren und Übersetzen

Im CFC erstellen Sie unter Verwendung der Funktionsbausteine die Projektierung und übersetzen diese. Nach allen Überprüfungen bauen Sie die Hardware auf.

### 4. Projektierung testen

Jetzt können Sie das Programm auf den SIMATIC TDC-Baugruppen ablaufen lassen, Online testen und ändern.

### 5. Projekt archivieren

Diese Vorgehensweise können Sie später auf Ihre eigenen Anwendungen übertragen.

**Die Aufgabe** Die Aufgabe enthält zwei Teile:

1. Ein **Sägezahngenerator** mit einer festen Frequenz gibt seinen Wert über einen DA-Wandler aus.
2. Ein **Lauflicht** mit acht Kanälen.

Zunächst definieren Sie für die entsprechenden Teilaufgaben die Einzelfunktionen und legen die notwendige Hardware fest:

### 1. Sägezahngenerator

Ein Sägezahn wird durch einen Integrator, der sich nach dem Überschreiten eines oberen Grenzwertes zurücksetzt, gebildet. Der Wert des Integrators wird über einen analogen Ausgang ausgegeben.


### 2. Lauflicht

Acht Komparatoren vergleichen den Sägezahnwert mit konstanten Werten. Die Ergebnisse werden über Digitalausgänge ausgegeben und steuern die LED-Anzeigen auf dem Interfacemodul an.

Das Lauflicht hat folgende Phasen:

- Alle Leuchtdioden sind aus.
- Die Leuchtdioden werden nacheinander ein- und wieder ausgeschaltet, sodass immer nur eine leuchtet.

## 1.2 Neues Projekt anlegen

Schritt	Vorgehensweise	Ergebnis
1	Doppelklick auf das Symbol  . (Wenn der STEP 7-Assistent startet, diesen abbrechen.)	Der SIMATIC Manager wird geöffnet.
2	Wählen Sie <b>Datei &gt; Neu</b> . Geben Sie in das Dialogfeld Projekt „ <b>My First Project</b> “ ein. Wählen Sie im Dialogfeld Ablageort (Pfad) „ <b>LW:\Siemens\Step7\S7proj</b> “. Klicken Sie auf <b>OK</b> .	Ihr neues Projekt wird angezeigt.
3	Wählen Sie <b>Einfügen &gt; Station &gt; SIMATIC TDC-Station</b> .	Das Hardware-Objekt „ <b>SIMATIC TDC-Station</b> “ wird eingefügt.

## 1.3 Hardware festlegen

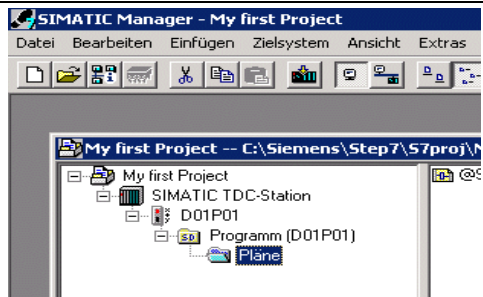
Der Aufbau des SIMATIC TDC-Baugruppenträgers wird in STEP 7 (HW Konfig) eingetragen.

Schritt	Vorgehensweise	Ergebnis
4	Markieren Sie das Hardware-Objekt „ <b>SIMATIC TDC-Station</b> “ und wählen Sie <b>Bearbeiten &gt; Objekt öffnen</b> .	<b>HW Konfig</b> wird aufgerufen.
5	Öffnen Sie ggf. mit <b>Ansicht &gt; Katalog</b> den Hardware-Katalog.	Der Hardware-Katalog wird mit allen verfügbaren Baugruppenfamilien geöffnet.
6	Wählen Sie aus der Baugruppenfamilie <b>SIMATIC TDC</b> und dem Katalog <b>Baugruppenträger</b> den <b>UR5213</b> aus und ziehen Sie ihn per Drag & Drop in das (obere) Fenster.	Der Baugruppenträger wird mit 21 Steckplätzen angezeigt.
7	Platzieren Sie nun nacheinander <b>&gt;CPU-Baugruppen &gt; CPU551</b> auf Steckplatz 1 <b>&gt;Signalbaugruppen &gt; SM500</b> auf Steckplatz 2 <b>&gt;Baugruppenträger &gt; SR51</b> auf die Steckplätze 3 bis 21	Der Baugruppenträger ist fertig bestückt.
8	Öffnen Sie mit <b>Bearbeiten &gt; Objekteigenschaften</b> den Eigenschaftendialog der CPU-Baugruppe CPU551.	Das CPU551-Dialogfeld mit allgemeinen Baugruppeninformationen und den Einstellregistern für Adressen, Grundtakt, zyklische Tasks und Alarmtasks erscheint.

9	Im Register <b>Grundtakt</b> stellen Sie die Grundabtastzeit T0 (hier: 1 ms) ein. Klicken Sie auf das Register <b>zyklische Tasks</b> und stellen Sie die Abtastzeit T1 auf 2 ms und T2 auf 4 ms. Klicken Sie auf <b>OK</b> .	Die erforderlichen Abtastzeiten sind eingetragen.  Der Eigenschaftendialog wird geschlossen.
10	Öffnen Sie mit <b>Bearbeiten &gt; Objekteigenschaften</b> den Eigenschaftendialog der Signalbaugruppe SM500.	Das SM500-Dialogfeld mit allgemeinen Baugruppeninformationen und dem Einstellregister für Adressen erscheint.
11	Im Register <b>Adressen</b> klicken Sie auf den <b>Vorbelegen</b> -Button. Klicken Sie auf <b>OK</b> .	Allen Adressen werden symbolische Namen für die spätere Verwendung in CFC-Plänen zugewiesen.
12	Prüfen Sie mit <b>Station &gt; Konsistenz prüfen</b> die Zusammenstellung Ihrer Hardware.	Wenn fehlerfrei, mit Schritt 13 fortfahren, sonst Hardwarezusammenstellung überprüfen.
13	Übersetzen Sie mit <b>Station &gt; Speichern und übersetzen</b> Ihre Hardwarekonfiguration.	Die Hardwarekonfiguration ist abgeschlossen.

## 1.4 Erstellen eines CFC-Plans

### 1.4.1 Einen neuen Plan erzeugen

Schritt	Vorgehensweise	Ergebnis
14	Wechseln Sie in den SIMATIC Manager und klappen Sie den Projektbaum bis zum Objekt Pläne aus.  Markieren Sie Pläne durch anklicken.	
15	Erzeugen Sie zweimal mit <b>Einfügen &gt; S7-Software &gt; CFC</b> einen neuen CFC-Plan.	Auf der rechten Seite des Projektfensters werden als neue Objekte die Pläne CFC1 und CFC2 dargestellt.
16	Wählen Sie den Plan CFC1 im Projektfenster aus und öffnen Sie mit <b>Bearbeiten &gt; Objekteigenschaften</b> den Eigenschaftendialog.  Tragen Sie den Namen „sawtooth generator“ ein. Klicken Sie auf <b>OK</b> .	Sie erhalten den Eigenschaftendialog des CFC-Plans.  Der Eigenschaftendialog wird geschlossen.
17	Wiederholen Sie Schritt 16 mit dem Plan CFC2 und benennen Sie ihn in „running lights“ um.	Die Pläne erscheinen im Projektfenster unter ihrem neuen Namen.

## 1.4.2 Einfügen, Parametrieren und Verschalten der Funktionsbausteine

Schritt	Vorgehensweise	Ergebnis
18	Wählen Sie den Plan „sawtooth generator“ aus und öffnen Sie den CFC-Editor mit <b>Bearbeiten &gt; Objekt öffnen</b> .	Der CFC-Editor mit dem Arbeitsbereich (1 Blatt) und dem Bausteinkatalog wird geöffnet. (Katalog fehlt? <b>Ansicht &gt; Katalog</b> wählen) (>1 Blatt? <b>Ansicht &gt; Blattsicht</b> wählen)
19	Öffnen Sie die Bausteinfamilie <b>Regelung</b> und ziehen Sie den Funktionsbaustein <b>INT</b> (Integrator) mit Drag & Drop in den Arbeitsbereich.	Der Baustein ist auf dem Blatt platziert und mit der Kennung für Ablauf in der zyklischen Task T1 versehen.
20	Mit <b>Bearbeiten &gt; Objekteigenschaften</b> öffnen Sie den Eigenschaftendialog des Funktionsbausteins INT.	Das INT-Dialogfeld mit allgemeinen Bausteininformationen und dem Einstellregister Anschlüsse erscheint.
21	Im Register <b>Allgemein</b> ändern Sie den Namen auf „sawtooth“.	
22	Im Register <b>Anschlüsse</b> tragen Sie Werte für die Bausteineingänge ein, z.B: <ul style="list-style-type: none"> <li>• X = 1</li> <li>• LU = 11250</li> <li>• TI = 5 ms</li> </ul> Klicken Sie auf <b>OK</b> .	Der Eigenschaftendialog wird geschlossen und die Funktionsbausteineingänge sind jetzt mit Werten versehen.
23	<b>Klicken</b> Sie zuerst auf den Ausgang QU und danach auf den Eingang S.	Der Ausgang QU (Obere Grenze) ist auf den Eingang S (Setzen) zurückgekoppelt.
24	Wählen Sie aus der Bausteinfamilie <b>EIN/AUS</b> den <b>DAC</b> (Analogausgabe) aus und platzieren Sie ihn neben dem Funktionsbaustein INT. Über <b>Bearbeiten &gt; Objekteigenschaften</b> öffnen Sie das Dialogfeld und ändern den Namen auf „analog output“. Im Register <b>Anschlüsse</b> tragen Sie z.B. ein: <ul style="list-style-type: none"> <li>• DM = 0</li> <li>• OFF= 0</li> <li>• SF = 1E6</li> </ul> Klicken Sie auf <b>OK</b> . Markieren Sie den Anschluss AD (Hardware-Adresse), rufen Sie mit <b>Einfügen &gt; Verschaltung zu Operand</b> das Dialogfeld für die Objektverschaltungen auf und öffnen Sie das Auswahlfenster. Sie markieren den ersten Eintrag und klicken auf <b>OK</b>	Die Eingänge des Bausteins sind parametrierbar.  Die Hardwareadresse des ersten analogen Ausgangskanals ist zugewiesen.
25	<b>Klicken</b> Sie im Baustein „sawtooth“ auf den Ausgang Y und danach im Baustein „analog output“ auf den Eingang X.	Der Sägezahngenerator ist mit dem Analogausgang verschaltet.

Alle Änderungen im CFC-Plan werden sofort abgespeichert.

Für die zweite Teilaufgabe (Lauflicht) gehen Sie nach dem gleichen Schema vor (ab Schritt 18).

Wechseln Sie in den SIMATIC Manager, öffnen Sie den CFC-Plan „running lights“, fügen Sie die Funktionsbausteine in den CFC-Plan ein, parametrieren und verschalten Sie.

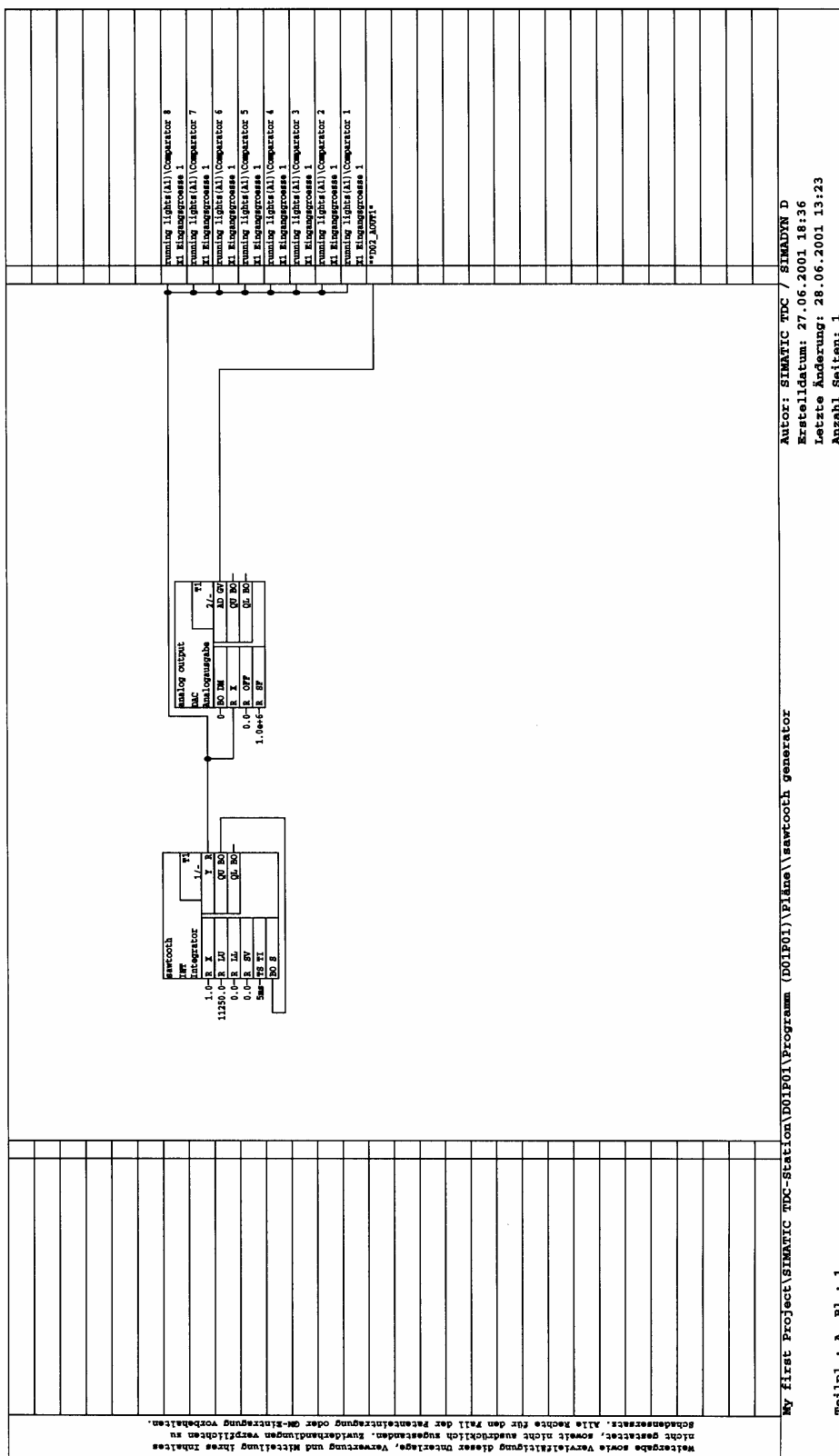
Alle notwendigen Informationen (Bausteinanzahl, -typ und -parameter) können Sie aus den nachfolgenden Abbildungen entnehmen. Ordnen Sie den ersten Funktionsbaustein (und alle weiteren) über

**Bearbeiten > Ablaufreihenfolge** in die zyklische Task T2 ein. Die Verschaltung vom Baustein „sawtooth“ zu den Komparatoren erfolgt durch CFC-Fensterumschaltung (**Fenster > ...**).

11.07.2001 16:46:53

My first Project\SIMATIC\_TDC-Station\D01P01\...\sawtooth generator

SIMATIC



Seite 1

Bild 1-2 Plan „sawtooth generator“

SIMATIC My first Project\SIMATIC TDC-Station\D01P01\...\running lights 11.07.2001 16:47:27

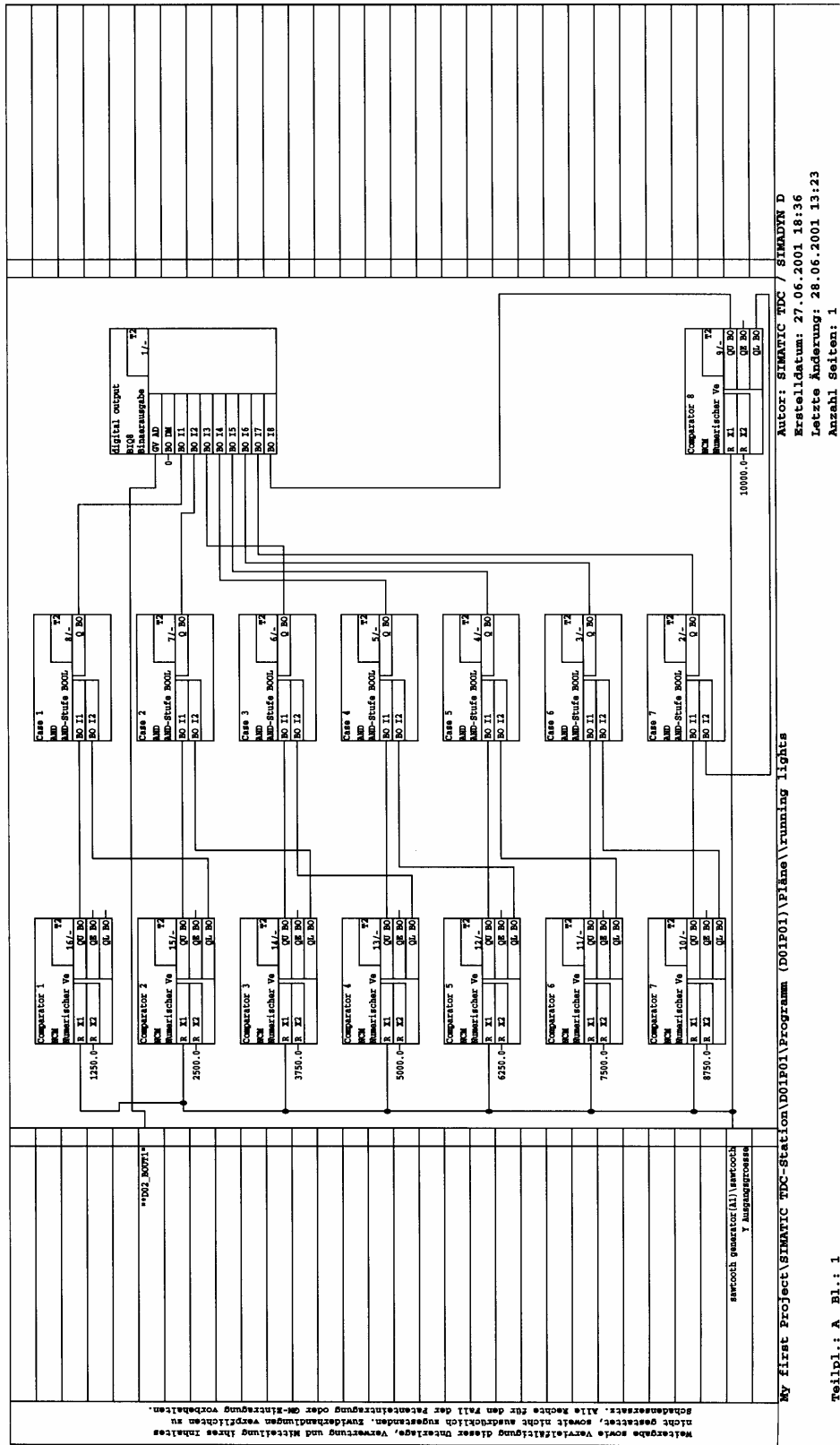


Bild 1-3 Plan „running lights“



## 1.5 Prüfen, Übersetzen und Laden des Projekts

### 1.5.1 Konsistenz des Projekts prüfen und übersetzen

Schritt	Vorgehensweise	Ergebnis
26	Starten Sie mit <b>Plan &gt; Konsistenz prüfen &gt; Pläne als Programm</b> , dann <b>OK</b> die Konsistenzprüfung Ihres Projekts. <b>Quittieren</b> Sie das Dialogfenster bzw. sehen Sie sich die Fehlermeldungen über <b>Details</b> genauer an.	Ein Dialogfenster zeigt das Ergebnis.
27	Starten Sie nach erfolgreicher Konsistenzprüfung mit <b>Plan &gt; Übersetzen &gt; Pläne als Programm</b> , dann <b>OK</b> die Übersetzung des Projekts. <b>Quittieren</b> Sie das Dialogfenster bzw. sehen Sie sich die Fehlermeldungen über <b>Details</b> genauer an.	Ein Dialogfenster zeigt das Ergebnis.  Ihr erstes Anwenderprojekt ist erstellt.

### 1.5.2 Laden des Anwenderprojekts auf die SIMATIC TDC-CPU-Baugruppe

**Einleitung** SIMATIC TDC bietet Ihnen die Möglichkeit

- online oder
- offline zu laden.

**Offline laden** Sie haben u.U. keine Verbindung von Ihrem PC/PG zur SIMATIC TDC-Station, darum nutzen Sie die Möglichkeit, auf ein Speichermodul zu laden.

Schritt	Vorgehensweise	Ergebnis
28	Wählen Sie <b>Zielsystem &gt; Laden</b> .	Sie erhalten ein Dialogfenster mit Optionen.
29	Wählen Sie „Anwenderprogramm“ und „offline“ aus. Schieben Sie das Speichermodul in den PCMCIA-Steckplatz des PG/PC. Starten Sie den Ladevorgang mit <b>OK</b> .	Eine Fortschrittsanzeige zeigt, wie das Speichermodul mit dem System und Ihrem Anwenderprogramm beschrieben wird.
30	Schieben Sie das Speichermodul in die SIMATIC TDC-Station und starten Sie die Station neu.	Ihr Anwenderprogramm wird gestartet.

**Online laden**

Sie haben eine Verbindung von Ihrem PC/PG zur SIMATIC TDC-Station und können das Programmspeichermodul in der CPU-Baugruppe direkt beschreiben.

Schritt	Vorgehensweise	Ergebnis
28	Überprüfen Sie, ob Ihre SIMATIC TDC-Station (Hardware) richtig aufgebaut und angeschlossen ist.	Beachten Sie dazu die Aufbauhinweise und Anschlussmöglichkeiten für die einzelnen Hardware-Komponenten in der entsprechenden Hardware-Dokumentation!
29	Stecken Sie das Speichermodul in die CPU-Baugruppe und starten Sie die SIMATIC TDC-Station.	Auf der Anzeige der CPU-Baugruppe erscheint eine blinkende „0“.
30	Installieren Sie im SIMATIC-Manager mit dem Menübefehl: <b>Extras &gt; PG/PC-Schnittstelle einstellen...</b> die Schnittstelle zwischen der SIMATIC TDC-Station und dem PC.	Sie erhalten ein Dialogfenster "Schnittstellen installieren/deinstallieren", in dem verschiedene Schnittstellen angeboten werden.
31	Wählen Sie in dem Dialogfenster „DUST1 Protokoll“ aus und installieren Sie dieses Protokoll mit <b>Installieren</b> → Quittieren Sie mit "Ja" und <b>Schließen</b> Sie das Dialogfenster. Wählen Sie die benutzte Schnittstelle aus und quittieren Sie mit <b>OK</b> .	Sie erhalten ein Dialogfenster, in dem Sie mit "Ja" oder "Nein", die Entscheidung treffen, ob Sie sofort online gehen möchten. Das Dialogfenster "PG-Schnittstelle einstellen" wird eingeblendet, in dem Sie den Zugriffsweg "DUST1 (COM1)" bzw. "DUST1 (COM2)" auswählen können.
32	Wählen Sie <b>Zielsystem &gt; Laden</b> .	Sie erhalten ein Dialogfenster mit Optionen.
33	Wählen Sie „System und Anwenderprogramm“, „Online (COM1)“ und Umladen beim erstmaligen Laden des Anwenderprogramms aus. <b>Hinweis:</b> Beim wiederholten Laden eines Anwenderprogramms können Sie dann auch als Umfang nur „Anwenderprogramm“ ohne "Umladen" angeben. Starten Sie mit "Laden".	Eine Fortschrittsanzeige zeigt, wie das Speichermodul mit dem System und Ihrem Anwenderprogramm beschrieben wird. Wenn der Ladevorgang abgeschlossen ist wird das Dialogfenster "Betriebszustand" mit dem Zustand "STOP" angezeigt.
34	Starten Sie die SIMATIC TDC-Station mit "Neustart" und wählen Sie danach "Schließen".	Ihr Anwenderprogramm wird gestartet und das Dialogfenster "Betriebszustand" wird mit dem Zustand "RUN" angezeigt.

## 1.6 Anwenderprojekt testen

### Einleitung

Im Testbetrieb können Sie

- Werte der Bausteinanschlüsse beobachten und Werte der Bausteineingänge verändern,
- Verbindungen erzeugen und löschen und
- Bausteine einfügen und löschen.

Die zum Test angemeldeten Werte sind gelb hinterlegt. Durch das Verändern von Parametern an Bausteineingängen können Sie gut das Verhalten beobachten.

Bevor Sie mit dem Test beginnen, überprüfen Sie, ob folgende Voraussetzungen erfüllt sind:

- Sie haben eine Verbindung zwischen dem PG/PC und Ihrer SIMATIC TDC-Station hergestellt.
- Sie haben das aktuelle Projekt auf das Speichermodul geladen, das sich in der CPU-Baugruppe befindet.
- Der dazugehörige CFC-Plan (z.B. „running lights“) ist geöffnet.

Schritt	Vorgehensweise	Ergebnis
35	Wählen Sie den Menübefehl: <b>Zielsystem &gt; Vergleichen</b> , um das Dialogfeld „Vergleichen“ anzuzeigen.	Die CPU-Namen mit Datum und Uhrzeit der letzten Übersetzung zwischen der aktuellen Projektierung und dem aktuellen CPU-Programm werden angezeigt. Bei Übereinstimmung ist das Ergebnis "Die Projektierung und das CPU-Programm stimmen überein".  Sie haben damit überprüft, dass die Kommunikation zwischen PG/PC und SIMATIC TDC-Station möglich ist.
36	Wählen Sie den Menübefehl: <b>Test &gt; Testeinstellungen</b> Geben Sie die Auffrischperiode für die Bildschirmdarstellung in Zehntelsekunden ein.  Bestätigen Sie die Änderung mit "OK".	Im Testbetrieb wird der Wert der Anschlüsse mit der eingestellten Auffrischperiode zyklisch auf dem Bildschirm aktualisiert.  Wenn die Rechenzeit nicht ausreicht, die Auffrischperiode zu erfüllen, werden Sie gewarnt. Die Regelung hat in jedem Fall die höhere Priorität.
37	Bevor Sie in den Testmodus schalten, stellen Sie die Test-Betriebsart von „Prozessbetrieb“ auf „Laborbetrieb“ mit <b>Test &gt; Laborbetrieb</b> um.  <b>Hinweis:</b> Im „Prozessbetrieb“ ist als Voreinstellung <b>kein</b> Anschluss zum Beobachten angemeldet. In dieser Test-Betriebsart müssten Sie die betreffenden Bausteine markieren und explizit zum Beobachten anmelden.	Damit sind alle Bausteinanschlüsse automatisch für das „Beobachten“ eingeschaltet (Werte sind gelb hinterlegt).
38	Wählen Sie den Menübefehl: <b>Test &gt; Testmodus</b>	In der Statusleiste erscheint grün hinterlegt der Text „Test: RUN (Labor)“.  Im Testmodus konnten Sie das dynamische Verhalten beobachten und verändern (Online).

### 1.6.1 Verbindung online trennen

**Vorgehensweise**      Selektieren Sie mit dem Mauszeiger im CFC-Plan den Bausteinanschluss dessen Verbindung sie trennen wollen. Danach entfernen Sie diese mit **Bearbeiten > Löschen**.

**Ergebnis**              Die Verbindungslinie zwischen den Anschlüssen verschwindet und am Anschluss erscheint als Parameterwert der letzte Wert, der auf der Verbindung übertragen wurde.

---

**HINWEIS**              Verbindungen zu globalen Operanden können online weder neu erzeugt noch gelöscht werden.

---

### 1.6.2 Verbindung online erstellen

**Vorgehensweise**      Selektieren Sie mit dem Mauszeiger im CFC-Plan den Bausteinanschluss von dem Sie eine Verbindung herstellen wollen.  
Mit gedrückter Umschalttaste selektieren Sie nun den Bausteinanschluss zu dem die Verbindung führen soll.

**Ergebnis**              Die Verbindungslinie zwischen den selektierten Anschlüssen wird erstellt und am Ausgang erscheint der aktuelle Parameterwert, der gerade übertragen wird.

### 1.6.3 Parametrierung online ändern

**Vorgehensweise**      Selektieren Sie mit einem Doppelklick den Bausteineingang, dessen Parameterwert geändert werden soll. Das Dialogfeld „Eigenschaften-Anschluss“ wird angezeigt, in dem Sie den Wert ändern können.

**Ergebnis**              Sie können die Auswirkung der Änderung sofort im CFC-Plan erkennen.

### 1.6.4 Baustein online einfügen

**Vorgehensweise**      Rufen Sie mit dem Befehl **Ansicht > Katalog** den Bausteinkatalog auf. Öffnen Sie die Bausteinfamilie und ziehen den gewählten Funktionsbaustein mit Drag & Drop in den Arbeitsbereich.

---

**HINWEIS**              Nicht alle Funktionsbausteine sind online einfügbar. Siehe unter „Projektierungsdaten“ in der Online-Hilfe zum Baustein.

---

### 1.6.5 Baustein online löschen

**Vorgehensweise**      Selektieren Sie den Funktionsbaustein und entfernen Sie ihn mit dem Befehl **Bearbeiten > Löschen**.

## 1.7 Ergebnisse

Sie haben nun die elementarsten Hantierungen in der CFC-Projektierung kennengelernt. Sie wissen wie man mit dem SIMATIC Manager ein Projekt angelegt, einen CFC-Plan erstellt und Funktionsbausteine aus einer Bibliothek eingefügt. Die Funktionsbausteine haben Sie verschaltet und parametrisiert. Sie haben ein ablauffähiges Programm erzeugt und in die CPU geladen. Im Testmodus konnten Sie das dynamische Verhalten beobachten und verändern.

Nun können Sie die Ergebnisse für das Beispielprojekt „My First Project“ im **Prozessbetrieb** anschauen, wenn Sie die dafür notwendige Hardware der SIMATIC TDC-Station (siehe Tabelle 1-1, Abschnitt 1.1.2) aufgebaut und angeschlossen haben.

### Sägezahngenerator

Um sich den Sägezahn anzusehen, müssen Sie ein Oszilloskop an die SIMATIC TDC-Station anschließen. Die nachfolgende Tabelle zeigt Ihnen die Belegung der Pins am Ausgangsstecker X1 der Signalbaugruppe SM500.

Der Bereich der Ausgangsspannung geht von -10 V bis +10 V.

Pin	Funktion	Ausgang
1	Analogausgang 1 +	Sägezahn
2	Analogausgang 1 -	

Tabelle 1-1 Auszug aus der Pinbelegung SM500, Stecker X1

### Lauflicht

Die Funktion des Lauflichts können Sie an der LED-Anzeige vom Interfacemodul SB10 beobachten.

## 1.8 Projekt archivieren

Schritt	Vorgehensweise	Ergebnis
44	Wählen Sie Im SIMATIC Manager <b>Datei &gt; Archivieren</b> .	Das Dialogfeld „Archivieren“ wird eingeblendet.
45	Wählen Sie im Dialogfeld "Archivieren" das Anwenderprojekt „My First Project“ aus. Klicken Sie auf <b>OK</b> .	Das Dialogfeld „Archivieren-Archiv auswählen“ wird eingeblendet. Der voreingestellte Dateiname „My_first.zip“ ist bereits mit Ablagepfad eingetragen.
46	Ändern Sie im Dialogfeld „Archivieren-Archiv auswählen“ bei Bedarf den Dateinamen und /oder den Ablagepfad und klicken Sie dann auf „Speichern“.	Das Projekt wird jetzt in dem ausgewählten Ablagepfad und Dateinamen als Zip-Datei gespeichert.

### HINWEIS

Wenn Sie in der Menüleiste **Datei > Dearchivieren** auswählen kann das archivierte Projekt jederzeit mit diesem Stand wiederhergestellt werden.



## 2 Systemsoftware

<b>Kapitelübersicht</b>	2.1	Projektierung	2-2
	2.2	Funktionsbeschreibung und Benutzerhinweise	2-42
	2.3	Systemplan @SIMD	2-46

## 2.1 Projektierung

### 2.1.1 Allgemeine Beschreibung

Das vorliegende Kapitel ist als Anleitung und Unterstützung zur Projektierung gedacht. Es erläutert allgemeine Projektierungsbedingungen bezüglich der eingesetzten Hard- und Software von SIMATIC TDC/SIMADYN D.

Es wird vorausgesetzt, daß der Leser mit Windows 95/98/NT, der Bedienung des SIMATIC Managers, HWKonfig sowie des CFC-Editors vertraut ist; sie werden hier nicht erläutert. Die Projektierungsanleitungen werden anhand von Bildern und Grafiken verdeutlicht. Diese Bilder sollen bestimmte Merkmale herausheben und haben nicht den Anspruch, die CFC-Fenster identisch darzustellen. Auf die Hardware (z.B. CPU's, Speichermodule, Kabel etc.) wird in diesem Handbuch nicht eingegangen, auch wenn in Beispiel-Projektierungen Hardware-Bezeichnungen verwendet werden; dafür ist das Handbuch "Hardware" heranzuziehen.

Dieses Handbuch gliedert sich in folgende Kapitel:

- Allgemeine Beschreibung
- Konfigurieren der Hardware
- Erstellen von CFC-Plänen
- Betriebszustände einer CPU-Baugruppe
- Beispielprojektierung einer CPU-Baugruppe
- Verwendung von Signaltransporten
- Bedeutung und Einsatzmöglichkeiten des Prozeßabbildes
- Bedeutung und Einsatzmöglichkeiten der CPU-Synchronisation
- Bedeutung der Prozessorauslastung

Zur Realisierung der meisten Anwendungen genügt die Lektüre der Kapitel "Allgemeine Beschreibung" bis Kapitel "Erstellen von CFC-Plänen". In den darauffolgenden Kapiteln sind noch detailliertere Informationen zu besonderen Systemeigenschaften von SIMATIC TDC/SIMADYN D beschrieben.

#### 2.1.1.1 Projektierungswerkzeuge

In der Praxis ist ein Projektteur vertraut mit der Auswahl der benötigten Hardwarebaugruppen aus einem Baugruppenspektrum und dem Erstellen von Funktionsplänen bzw. Blockschaltbildern zur Erzielung der gewünschten Technologiefunktionen. Diese Tätigkeiten werden von SIMATIC TDC/SIMADYN D durch **HWKonfig** (Projektierungswerkzeug



zur Beschreibung der Hardwarekonfiguration von SIMATIC TDC/SIMADYN D-Stationen) und den **CFC** (Bausteintechnik mit zahlreichen Standardfunktionsbausteinen) abgedeckt.

### 2.1.1.2 Projektierungsschritte

Eine Projektierung von SIMATIC TDC/SIMADYN D wird durch die Abfolge der Projektierungsschritte realisiert:

1. Erstellen der Hardwarekonfiguration und
2. Erstellen der CFC-Pläne.

### 2.1.1.3 Nomenklatur und Bibliotheken

#### Namensvergabe

Generell gilt für die bei der Projektierung zu vergebenden Namen bei SIMATIC TDC/SIMADYN D:

- Stationsnamen
  - max. 24 Zeichen
- Baugruppen
  - als maximale Länge sind 6 Zeichen erlaubt.

Zeichenfolge	erlaubte Zeichen	Beispiel
erstes Zeichen	Alpha- und Sonderzeichen	A-Z, @
zweites Zeichen	Alphanumerische Zeichen und Sonderzeichen	A-Z, 0-9, _, oder @ wenn erstes Zeichen @ ist
weitere Zeichen	Alphanumerische Zeichen und Sonderzeichen	A-Z, 0-9, _

Tabelle 2-1 Nomenklatur bei Namensvergabe von Baugruppen

- Plan- bzw. Funktionsbausteinennamen
  - eine Verbindung beider Namen darf eine maximale Länge von 24 Zeichen nicht überschreiten.

Name	max. Länge	erlaubte Zeichen	nicht erlaubte Zeichen
Plan	22		*, -, ?, <, >,
Funktionsbaustein	16		“

Tabelle 2-2 Nomenklatur bei Namensvergabe von Plan- und Funktionsbausteinennamen

- Kommentare dürfen
  - für Baugruppen bis zu 255 Zeichen,
  - für Pläne bis zu 255 Zeichen,
  - für Funktionsbausteine und Parameter bis zu 80 Zeichen lang sein.
- Anschlüsse mit speziellen Funktionen haben vorangestellte Zusätze:
  - das Dollarsymbol "\$" (Verschaltung von Signalen zwischen CPUs),
  - das Sternsymbol "\*" (symbolische Hardwareadressen),
  - oder das Ausrufezeichen "!" (virtuelle Adressierung).

Diese Zusätze werden von HWKonfig bzw. CFC automatisch vorangestellt. Funktionsbausteinennamen dürfen auf einer CPU nicht mehrfach auftreten. Die Einhaltung der Namensregeln wird bei der Eingabe geprüft.

## Bibliotheken

Hardwarebaugruppen und Funktionsbausteintypen sind in Bibliotheken zusammengefaßt. Die gewünschten Funktionsbausteine können daraus mit HWKonfig bzw. dem CFC-Editor aufgerufen werden.

Pro CPU können mehrere Funktionsbausteinbibliotheken benutzt werden. Als Vorbelegung steht bereits die Standard-Funktionsbausteinbibliothek "FBSLIB" mit über 200 Funktionsbausteinen zur Verfügung, deren Funktionalität für die meisten Anwendungen ausreicht. Weitere Zusatzbibliotheken können bei Bedarf für die jeweilige CPU "importiert" werden. Die Bibliotheken sind im Verzeichnis „step7\s7cfc\sdblocks\std (SIMADYN D) bzw. ...\.tdc (SIMATIC TDC)“ zu finden.

## 2.1.2 Konfigurieren der Hardware

### **SIMATIC TDC/SIMADYN D- Station konfigurieren**

Zur Projektierung der Hardwarekonfiguration von SIMATIC TDC/SIMADYN D-Stationen dient HWKonfig. Eine SIMATIC TDC/SIMADYN D-Station besteht dabei aus einem Baugruppenträger mit bis zu 20/8 CPUs und sonstigen Hardwarebaugruppen. Bei Bedarf können auch mehrere Stationen miteinander gekoppelt werden. Die jeweils zu projektierenden Baugruppen können aus einem Baugruppenspektrum, dem Hardwarekatalog des HWKonfig, ausgewählt werden. Hier stehen Baugruppenträger, CPUs, Ein/Ausgabebaugruppen, Kopplungsbaugruppen etc. zur Auswahl zur Verfügung.

HWKonfig definiert die Hardwarekonfiguration des Systems durch:

- den eingesetzten Baugruppenträger mit Bestimmung der Busstruktur (Busabschluß, Daisy Chain),
- seine Bestückung mit den projektierten Hardwarebaugruppen sowie

- die Festlegung hardwarerelevanter Angaben wie Tasks, Synchronisation etc.

### 2.1.2.1 Der erste Schritt: Auswählen der Hardwarebaugruppen

Als Baugruppen aus dem Hardware-Katalog von HWKonfig stehen zur Verfügung:

#### Kurzübersicht der Hardware

Hardware	Beschreibung
Baugruppenträger	Versch. Typen je nach Steckplatzanzahl, Busausstattung, Belüftung etc.
Ein/Ausgabebaugruppen	Peripheriebaugruppen zur Eingabe/Ausgabe von Prozeßsignalen (Analog-/ Binär-/ Ein-/Ausgaben, Drehzahlerfassungen etc.)
Erweiterungsbaugruppen	Peripheriebaugruppen zur Eingabe/Ausgabe von Prozeßsignalen. Sie sind zur Erzielung hoher Datenraten unter Umgehung des Rückwandbusses direkt mit einer CPU-Baugruppe verbunden.
Kommunikationsbaugruppen	Baugruppen zum Bereitstellen der Kommunikationsdienste.
Koppelspeicher	Baugruppen zum Datenaustausch zwischen mehreren CPU's.
CPU-Baugruppen	Baugruppen, auf denen das projektierte Steuer- bzw. Regelprogramm abgearbeitet wird. Neben einer CPU können bis zu zwei Erweiterungsbaugruppen gesteckt werden.
Sonderbaugruppen	Baugruppen mit Sonderfunktionen.
Steckplatzabdeckungen	Abdeckung eines unbelegten Steckplatzes gegen Verschmutzung und als EMV-Maßnahme.
Submodule	Ein Modul das in bzw. auf eine Baugruppe gesteckt wird, z.B. ein Speichermodul für eine CPU oder ein Schnittstellenmodul für eine Kommunikationsbaugruppe.
Technologiekomponenten	Baugruppenträger sowie Baugruppen für Stromrichtergeräte.

Tabelle 2-3 Hardwarekomponenten

#### Weitere Informationen

zu den einzelnen Baugruppen und Modulen siehe Handbuch "SIMATIC TDC/SIMADYN D Hardware".

Für jeden Steckplatz des Baugruppenträgers wird mit HWKonfig eine Baugruppe eventuell mit Submodul projektiert. Dadurch ergibt sich bei der Projektierung ein genaues Abbild des realen Baugruppenträgers. Jedes Modul ist beim Aufruf mit einem Namensvorschlag versehen, der gemäß den Namensregeln geändert werden kann. Freigelassene Steckplätze sollten mit Steckplatzabdeckungen versehen werden.

## 2.1.2.2 Der zweite Schritt: Parametrieren der Hardwarebaugruppen

Nach der Auswahl sind die Baugruppen mit HWKonfig zu parametrieren. Dabei sind

- die Abtastzeiten der zyklischen Tasks,
- die Synchronisation von zyklischen oder alarmgesteuerten Tasks mehrerer CPUs einer Station,
- die Prozeßalarme und Kommentare

einzustellen. Dazu gibt es verschiedene Parametrierdialogfenster in HWKonfig.

### Parametrierdialoge in HWKonfig

In den Dialogfenstern der einzelnen Baugruppen können die Voreinstellungen noch verändert werden. So enthält z.B. der Parametrierdialog für CPU-Baugruppen u.a. die Angabe "zyklische Tasks". Damit sind die Abtastzeiten der 5 zyklischen Tasks veränderbar.

### Bezeichnungsschema

In HWKonfig müssen mindestens ein Baugruppenträger und alle dort vorhandenen Baugruppen und Subbaugruppen konfiguriert werden. Bei der Erzeugung einer Baugruppe wird der Baugruppenname mit einem Vorschlag besetzt. Der Vorschlag ist im Rahmen der Namenslänge (max.6 Zeichen) und des Zeichenvorrats (siehe Kapitel "Allgemeine Beschreibung"), mit (A-Z,0-9,\_,@) frei überschreibbar. Es wird empfohlen, die Namen gemäß dem Schema der folgenden Tabelle für Anlagenkomponenten zu wählen:

Hardware	Logischer Name	Bezeichner	Bedeutung
Baugruppenträger	An00	n	Baugruppenträgernummer, beginnend ab 1
CPU	Dxy_Pn	xy n	Steckplatznummer Nummer der CPU
Submodul	Dxyj	xy j	Steckplatznummer Nummer des Submoduls
Koppelspeicher	Dxy__A	xy	xy = Steckplatznummer
Rahmenkopplung	Dxy__B	xy	xy = Steckplatznummer
serielle Kopplungen	Dxy__C	xy	xy = Steckplatznummer
sonstige Baugruppen	Dxy	xy	xy = Steckplatznummer

Tabelle 2-4 Bezeichnungsschema für die Hardwarekonfiguration in HWKonfig

### Definition Steckplatznummer

Die Steckplatznummer einer Baugruppe bezeichnet die Nummer des Steckplatzes im Baugruppenträger, auf dem die betreffende Baugruppe projiziert ist. Bei einem SR24 mit 24 Steckplätzen sind das die Steckplätze 1 bis 24.

Alle Submodule einer Baugruppe werden, mit 1 beginnend, durchnummeriert. Das in der Tabelle am weitesten oben stehende Submodul erhält die Nummer 1.

Der empfohlene CPU-Baugruppenname ist 6 Zeichen lang. Die logische Prozessornummer (im Baugruppenträger von links nach rechts) erscheint im Betrieb unabhängig von der Namensgebung auf der Siebensegmentanzeige der CPU-Baugruppe.

---

**HINWEIS** Die projektierten Baugruppennamen müssen innerhalb einer Station eindeutig sein.

---

**Die verschiedenen Tasks einer CPU** Die Bearbeitung der projektierten Funktionsbausteine erfolgt über

- 5 zyklische Tasks oder/und
- 8 Alarmtasks.

Dabei kann der Start einer Alarmtask gegenüber dem Zeitpunkt des auslösenden Prozeßalarms um eine frei projektierbare Verzögerungszeit versetzt werden.

**Der Systemplan** Der Systemplan, in dem das Verhalten der Siebensegmentanzeige, des Quittiertasters etc. projektiert ist, wird in einem neu angelegten SIMATIC TDC-Programm/SIMADYN D verwaltet und darf nicht gelöscht werden. Die Abtastzeit des Systemplans ist mit ca. 128 ms werksseitig vorbesetzt.

### 2.1.2.3 Der dritte Schritt: Überprüfung der Projektierung

Am Ende der Hardwarekonfiguration sind die projektierten Angaben durch eine stationsweite Konsistenzprüfung zu verifizieren. Dabei wird die gesamte Hardware-Projektierung von HWKonfig auf Fehlerfreiheit geprüft. Etwaige Fehler oder Unvollständigkeiten der Projektierung werden angezeigt und können so behoben werden (siehe Kapitel "Beispielprojektierung einer CPU-Baugruppe").

### 2.1.3 Erstellen von CFC-Plänen

**Beschreibung des CFC-Editors** Ein CFC-Plan (Continuous Function Chart) wird mit dem CFC-Editor erstellt. Dies ist ein Projektierungswerkzeug zur Beschreibung kontinuierlicher Vorgänge durch grafische Verschaltung komplexer Funktionen in Form von einzelnen Funktionsbausteinen. Der CFC enthält somit die grafische Realisierung eines Technologieproblems durch Verschalten und Parametrieren von Funktionsbausteinen. Dadurch wird eine für den Projektteur leichte blockschaltbildnahe Programmierung ermöglicht.

## Aufbau der CFC-Pläne

Ein CFC besteht aus mehreren CFC-Plänen zu je 6 Blättern. Jedes Blatt kann eine unterschiedliche Anzahl von verschiedenen Funktionsbausteinen enthalten. Die Anzahl ist nur durch die mögliche grafische Anordnung begrenzt. In der Übersichtsdarstellung des CFC-Editors werden alle 6 Blätter eines Plans dargestellt, in der Blattdarstellung kann ein einzelnes Blatt detailliert dargestellt werden. Die im CFC-Editor aufrufbaren Funktionsbausteine sind in Funktionsbausteinklassen untergliedert, die jeweils zusammengehörige Funktionalität enthalten. z.B. Logikbausteine, Arithmetikbausteine etc. Jede Funktionsbausteinklasse enthält wiederum eine Anzahl verschiedener Funktionsbausteintypen.

Der CFC-Editor definiert die technologische Projektierung durch:

- die Auswahl, Verschaltung und Parametrierung der projektierten Funktionsbausteine,
- die Festlegung der Ablaufeigenschaften der Funktionsbausteine,
- die Erzeugung ablauffähiger Programme zum Programmieren der CPU-Speichermodule.

### 2.1.3.1 Der erste Schritt: Auswählen der Funktionsbausteine

In der Standardbibliothek FBSLIB stehen verschiedene Funktionsbausteinklassen zur Verfügung. Die einzelnen Funktionsbausteine können mit dem CFC-Editor aufgerufen und auf den Planblättern platziert werden. Ein nachträgliches Löschen, Verschieben und Kopieren von Einzelbausteinen oder Bausteingruppen ist jederzeit möglich.

#### Weitere Informationen

zu Funktionsbausteinen siehe Referenzhandbuch "SIMATIC TDC/SIMADYN D Funktionsbausteinebibliothek".

### 2.1.3.2 Der zweite Schritt: Parametrieren und Verschalten der Funktionsbausteine

Nach der Auswahl der Funktionsbausteine sind diese mit dem CFC-Editor zu verschalten und zu parametrieren. Dabei ist auch die Task, in der die einzelnen Funktionsbausteine gerechnet werden, festzulegen.

#### Parametriedialoge im CFC-Editor

Durch einen Doppelklick auf den jeweiligen Funktionsbausteinkopf bzw. unter der Menüauswahl **Bearbeiten > Objekteigenschaften** können abweichend von den Voreinstellungen folgende Angaben projektiert werden:

Angaben	Beschreibung
Allgemein	Projektierbar ist der Name und ein Kommentar, der im Funktionsbausteinkopf angezeigt wird. Unter "Spezielle Objekteigenschaften" können Sie die Schritte ausführen, die nötig sind, um einen Baustein für das Bedienen und Beobachten durch WinCC vorzubereiten.

Angaben	Beschreibung
Ablaufeigenschaften	Hier läßt sich die unter "Funktionsbaustein einfügen" festgelegte Ablaufreihenfolge eines Funktionsbausteines innerhalb einer Task verändern. Der selektierte Funktionsbaustein kann in der Ablaufreihenfolge "gesucht", "ausgebaut" und an anderer Stelle wieder "eingebaut" werden.
Anschlüsse	Hier können für alle Parameter folgende Anschlußdaten eingegeben werden: <ul style="list-style-type: none"> <li>• Wert und Kommentar von Ein- und Ausgangsparametern</li> <li>• Sichtbarkeit im CFC-Plan für nicht verschaltete Parameter</li> <li>• Kennung Parameter für Test angemeldet setzen oder sperren</li> <li>• Skalierungswert für Parameter mit Datentyp REAL</li> <li>• Einheitentexte</li> </ul>

Tabelle 2-5 Funktionsbausteine projektieren

**Weitere Informationen**

zur Thematik "Funktionsbausteine verschalten" siehe Handbuch "D7-SYS – STEP 7, CFC und SFC projektieren, Kapitel CFC".

**Festlegen der Ablaufeigenschaften**

Mehrere innerhalb einer Task aufeinanderfolgend berechnete Funktionsbausteine können zu einer Ablaufgruppe zusammengefaßt werden. Diese bietet neben der möglichen Strukturierung der Tasks die Möglichkeit, die Bearbeitung separat ein- und auszuschalten.

**HINWEIS**

Wird eine Ablaufgruppe über einen mit ihr verschalteten Funktionsbausteinausgang ausgeschaltet, so werden alle in ihr enthaltenen Funktionsbausteine nicht mehr gerechnet.

Durch Zuordnung der Funktionsbausteine zu einer zyklischen oder alarm-gesteuerten Task oder Ablaufgruppe und Festlegung der Position innerhalb der Task oder Ablaufgruppe bestimmt der Projektteur die Ablaufeigenschaften der Funktionsbausteine. Diese Eigenschaften sind entscheidend für das Verhalten des Zielsystems bezüglich

- der Totzeiten,
- der Reaktionszeiten,
- der Stabilität von zeitabhängigen Strukturen.

**Zuordnung der Funktionsbausteine zu zyklischen Tasks**

Die Zuordnung der Funktionsbausteine zu einer der fünf zyklischen Tasks geschieht beim Bausteinaufruf mit Hilfe des CFC-Editors oder im Programmteil Ablaufreihenfolge des CFC-Editors. Jeder Funktionsbaustein kann so einer zyklischen Task und einer Abarbeitungsreihenfolge innerhalb der Abtastzeit der Task zugeordnet werden.

**Zuordnung der Funktionsbausteine zu Alarm-Tasks**

Um Funktionsbausteine alarmgesteuert abzuarbeiten, werden diese beim Aufruf oder im Programmteil "Ablaufreihenfolge des CFC-Editors" in der gewünschten Reihenfolge unter einer der 8 Alarmtasks eingetragen. Die Berechnung einzelner Funktionsbausteine kann so durch einen bestimmten Prozeßalarm angestoßen werden.

---

**HINWEIS**

Im Gegensatz zu zyklischen Tasks werden Alarmtasks nicht in äquidistanten Zeitabständen gestartet, sondern jeweils beim Auftreten eines Prozessalarms.

---

**Ersatzabtastzeit projektieren**

Einige Funktionsbausteine, z.B. manche Regelungsbausteine, benötigen aus programmtechnischen Gründen eine Bearbeitung in regelmäßigen zeitlichen Abständen. Sollen diese in einer Alarmtask projiziert werden, so muß für diese Alarmtask im Programmteil HWKonfig eine Ersatzabtastzeit projiziert werden. Diese sollte etwa der mittleren Zeitspanne zwischen zwei Prozeßalarmen entsprechen.

Sie können mit einem Doppelklick auf die Baugruppe die Ersatzabtastzeit unter dem Menüpunkt **Grundtakt > Synchronisieren** projektieren.



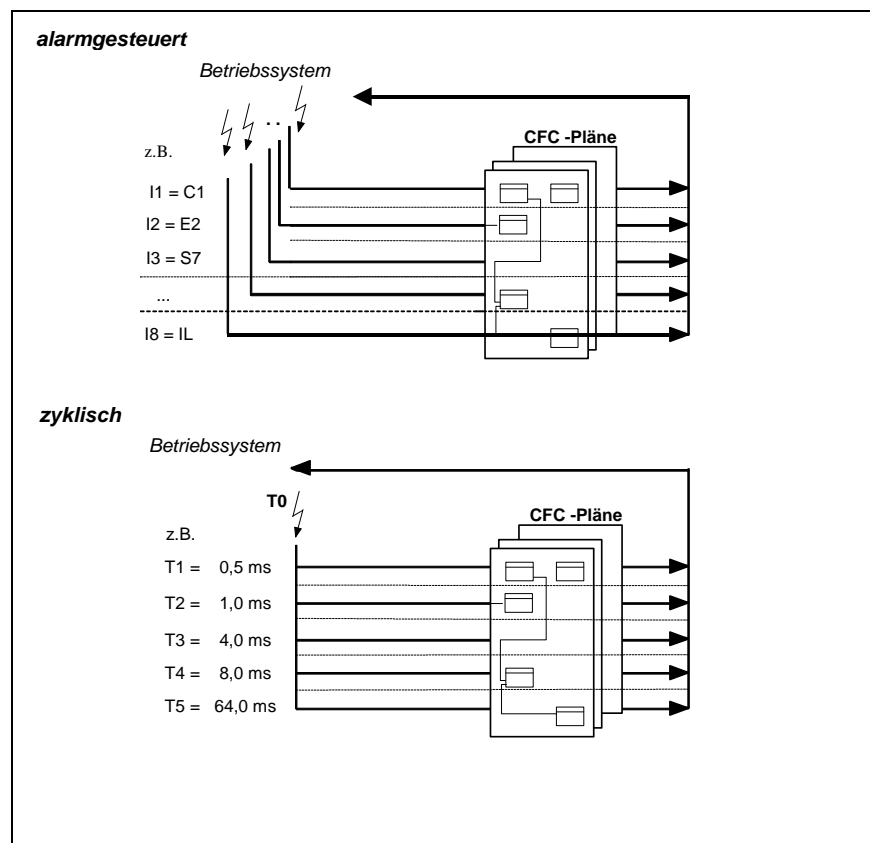


Bild 2-1 Bearbeitung der Funktionsbausteine durch das Betriebssystem

### Bearbeiten der Funktionsbausteine

Durch Verschaltung und Parametrierung der Funktionsbausteine kann die gestellte Steuer- bzw. Regelaufgabe blockschaltbildnah mit SIMATIC TDC/SIMADYN D umgesetzt werden. Ein Funktionsbausteintyp kann dabei beliebig oft eingesetzt werden. Die Parametrierung und Verschaltung der Funktionsbausteine findet an den Bausteinanschlüssen statt.

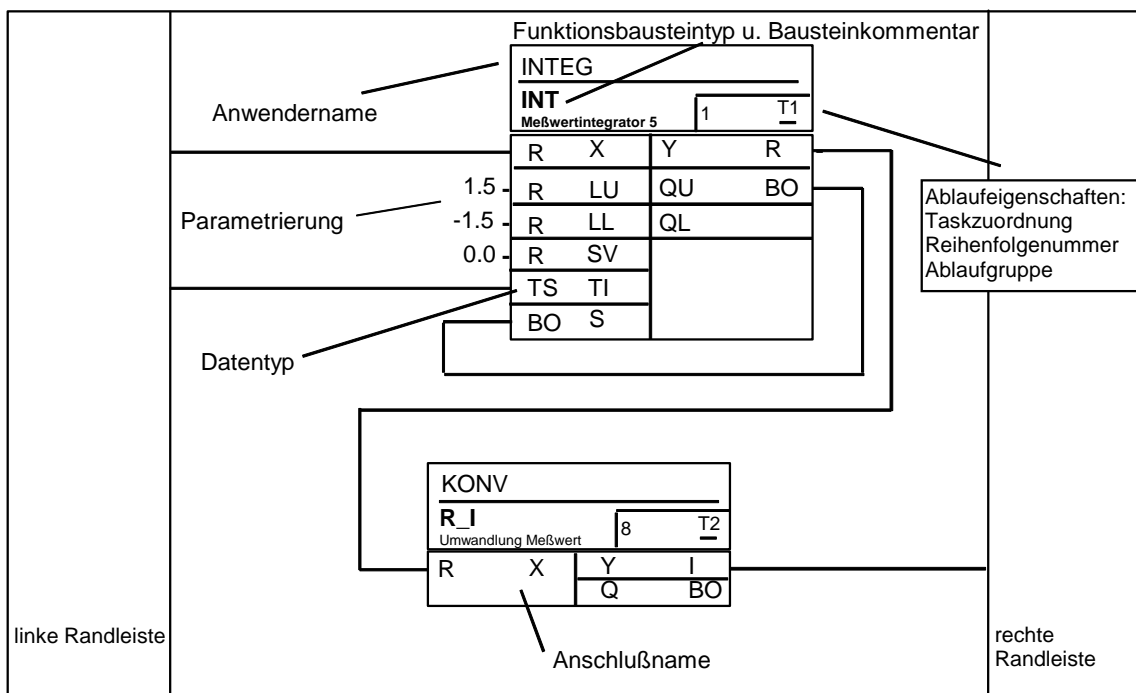


Bild 2-2 CFC Planblatt- Arbeitsfläche

Zur allgemeinen Parametrierung der Funktionsbausteine bzw. zur Verschaltung untereinander gibt es

- Eingangsanschlüsse (Funktionsbausteineingänge) und
- Ausgangsanschlüsse (Funktionsbausteinausgänge).

### Eingangsanschlüsse

Die Eingangsanschlüsse können vom Projektteur mit Konstanten parametrierung oder mit anderen Funktionsbausteinausgängen verschaltet werden. Beim Aufruf der Funktionsbausteine sind Eingänge und Ausgänge dabei bereits mit einer Vorbesetzung belegt, die bei Bedarf noch geändert werden kann.

### Ausgangsanschlüsse

Die Ausgangsanschlüsse können mit anderen Eingängen verschaltet werden oder mit einem vom Vorbesetzungswert abweichenden Initialisierungswert besetzt werden. Dieser Wert steht dann an diesem Anschluß, wenn der Funktionsbaustein im Betriebszustand INIT das erste mal gerechnet wird. Dies ist sinnvoll, wenn eine gezielte Vorbesetzung des Ausgangs eines Flip-Flop-Bausteins vorgenommen werden soll.

### Randleisten

Die Randleisten am linken und rechten Teil eines CFC-Blattes enthalten zum einen die Verweise zu den verschalteten Objekten, z.B. andere Bausteine oder Ablaufgruppen, die sich nicht auf dem aktuellen Blatt befinden. Zum anderen enthalten sie die Nummer des Konnektors (Abbruchstelle), wenn der Autorouter wegen Überfüllung des Blattes die Verbindungslinie zur Randleiste nicht ziehen konnte.

---

<b>Überlaufseiten</b>	Überlaufseiten werden automatisch angelegt, wenn auf einem Blatt mehr Randleisteneinträge erzeugt werden, als der Platz zum Anzeigen auf einer Seite bietet. Eine Überlaufseite besteht ausschließlich aus den Randleisten und enthält keine weiteren Objekte.
<b>Parametrieren</b>	<p>An jedem Eingang oder Ausgang kann statt einer Verschaltung auch eine Konstante, abweichend von der Vorbesetzung, parametriert werden.</p> <p>Ein Bausteinanschluß kann durch einen Pseudokommentar als Parameter gekennzeichnet werden.</p> <p><b>Weitere Informationen</b> zu Parametrieren siehe Handbuch "System- und Kommunikationsprojektierung D7-SYS, Kapitel Parametrieren von SIMATIC TDC/SIMADYN D.</p>
<b>Verschalten</b>	<p>Unter dem Begriff Verschalten versteht man:</p> <ul style="list-style-type: none"><li>• Die Verbindung eines Funktionsbausteinenausganges mit einem anderen Funktionsbausteineingang auf der gleichen CPU,</li><li>• Verschaltungen eines Funktionsbausteinenausganges zu einer Ablaufgruppe,</li><li>• Verschaltung eines Funktionsbausteinenausganges mit einem globalen Operanden bzw. eines globalen Operanden mit einem Funktionsbausteineingang. Ein globaler Operand kann folgendes sein:<ul style="list-style-type: none"><li>– Ein Name mit vorgestelltem Dollar "\$"-Zeichen, d.h. eine Verschaltung eines Signals von oder zu einem Funktionsbaustein einer anderen CPU.</li><li>– Eine virtuelle Verbindungsangabe oder eine virtuelle Verbindung, d.h. eine Übertragung von Prozeßdaten zwischen Funktionsbausteinen über beliebige Kopplungen mit Hilfe des Dienstes Prozeßdaten.</li><li>– Eine symbolische Hardwareadresse. Eine Hardwareadresse ist dabei eine symbolische Bezeichnung einer oder mehrerer zusammengehöriger Klemmen einer Baugruppe, z.B. Binäreingänge einer Ein-/Ausgabebaugruppe. Die symbolische Hardwareadresse wird im Programmteil HWKonfig definiert.</li><li>– Eine Namensreferenz, d.h. der Name eines Meldesystems.</li></ul></li></ul> <p>Alle Arten der Verschaltung, die ein Planblatt verlassen, erzeugen einen entsprechenden Querverweis auf den Randleisten des CFC-Blattes.</p>
<b>Kommentare</b>	Jeder Funktionsbausteinanschluß des CFC-Blattes kann mit einem Kommentar versehen werden.

## Pseudo-kommentare

Es gibt drei Pseudokommentare, die durch ein vorangestelltes @-Zeichen kenntlich gemacht werden und durch Leerzeichen getrennt vor dem normalen Kommentartext enthalten sein können:

### 1. @DATX

- Der Eingangsanschluß ist unter Umgehung der Konsistenzmechanismen verschaltet (siehe Kapitel "Beschreibung und Verwendung von Signaltransporten").

### 2. @TP\_bnnn

- Ein damit gekennzeichnete Anschluß kann auch als Parameter angesprochen werden. Der Parameter ist an Bausteineingängen mit Bediengeräten les- und veränderbar und an Bausteinausgängen nur lesbar.

- Über diese Schnittstellen können die als Parameter definierten Anschlüsse auch über Stromrichterbedienfelder oder SIMOVIS gelesen und abgeändert werden. Die folgenden Variablen werden eingesetzt:

- b: Bereichskennung "H", "L", "c" oder "d"
  - kennzeichnet den Parameternummernbereich
  - "H" oder "L": Anschlüsse können gelesen und verändert werden
  - "c" oder "d": Anschlüsse können nur gelesen werden

- nnn: dreistellige Parameternummer
  - 000 bis 999

### 3. @TC\_nnnn:

- Ein Technologiekonnektor @TC\_nnnn an einem Bausteinausgang kann mit der BICO-Technik mit einem Parameter am Bausteineingang verschaltet werden. Ein Technologiekonnektor wird über seine Nummer identifiziert:

- nnnn: vierstellige Technologiekonnektornummer
  - 0000 bis 9999

### Weitere Informationen

zu Parametern und Technologiekonnektoren siehe Handbuch "System- und Kommunikationsprojektierung D7-SYS, Kapitel Parametrieren von SIMATIC TDC/SIMADYN D".

## 2.1.3.3 Der dritte Schritt: Übersetzen und Laden des Anwenderprogramms in die CPU

Nachdem alle erforderlichen Hardware-Baugruppen mit HWKonfig und die gewünschten Funktionsbausteine auf den einzelnen Plänen mit dem CFC-Editor projiziert wurden, kann die Software durch den Compiler in

den Maschinencode der CPU übersetzt werden. Danach gibt es zwei Möglichkeiten:

**Offline-Laden**

Ein Speichermodul wird mit der PCMCIA-Schnittstelle des Projektierungs-PCs programmiert. Nach dem Stecken aller richtig programmierten Speichermodule aller CPU's eines Baugruppenträgers sind die Baugruppen betriebsbereit.

**Online-Laden**

Das Anwenderprogramm und das Betriebssystem werden über eine serielle Kommunikationsverbindung direkt vom Projektierungs-PC in die CPU geladen.

## 2.1.4 Betriebszustände einer CPU Baugruppe

Im System SIMATIC TDC/SIMADYN D sind die in der folgenden Tabelle gezeigten Systemzustände möglich:

Betriebszustand	Power off	INIT	RUN	STOP		
				Anwenderstop	Initialisierungsfehler	Systemfehler
interner Systemzustand						
Zustandsbeschreibung	spannungsloser Zustand	Systemanlauf (Initialisierung)	zyklischer Betrieb (Normalbetrieb)	durch Anwender ausgelöster Stopbetrieb	Zustand nach Init.-fehler	Zustand nach schwerem Systemfehler
Eigenschaften	System nicht in Betrieb	Hochlauf des Systems --> keine Beeinflussung von außen möglich	Funktionalität gemäß Projektierung	keine zyklische Bearbeitung -> schneller Download	Initialisierung fehlerhaft --> kein Übergang in zyklischen Betrieb	schwerer Systemfehler -> Abbruch der Bearbeitung
Siebensegmentanzeige	aus	'0'	PN-Nummer ('1' ... '8') bzw. 'C', 'E', 'b', 'A'	'd' (blinkend, wenn Download läuft)	'0' (Verursacher blinkend)	'H' (Verursacher blinkend)
rote LED auf T400	aus	aus	blinkt mit kleiner Frequenz	blinkt mit mittlerer Frequenz	blinkt mit schneller Frequenz	an
verfügbare Diagnose-Schnittstellen	--	keine	alle projektierten ( <b>eine muß auf erster CS7-SS sein</b> ) und lokale SS		lokale SS und erste CS7-SS	lokale SS
mögliche Bedienfunktionen	--	keine	volle Funktionalität von CFC-Online	nur Diagnose bzw. Download	nur Diagnose bzw. Download	nur Diagnose bzw. Download
Verwaltung durch Oberfläche (CFC)	--	--	Zustände können per Dialog vom Anwender abgefragt werden			

Tabelle 2-6 Systemzustände einer CPU-Baugruppe

Begriffsdefinition	Begriff	Beschreibung
	erste CS7-SS	SIMATIC TDC: Schnittstelle, die in der ersten CP50M0 / CP51M1 im Baugruppenträger (von links gezählt) ganz oben steckt  SIMADYN D: Schnittstellenmodul (SS4 bzw. SS52), das in der ersten CS7 im Baugruppenträger (von links gezählt) ganz oben steckt
	Diagnose	nur Auslesen von Fehlerfeldern möglich

Tabelle 2-7 Begriffserklärung

Es gibt die Betriebszustände INIT, RUN und STOP, wobei sich der Betriebszustand STOP in drei unterschiedliche Systemzustände aufgliedert.

**Systemzustand Anwenderstop**

Der Systemzustand "Anwenderstop" wurde neu implementiert und dient dazu, über SS52/MPI, SS4/DUST1 (SIMADYN D), CP50M0/CP51M1 (SIMATIC TDC) oder lokale Schnittstelle **schnell** ein Programm zu laden. Schnell bedeutet, daß in diesem Zustand die zyklische Bearbeitung gestoppt wird und die gesamte Rechenzeit der CPU für den Download zur Verfügung steht. In der Siebensegmentanzeige steht ein 'd', das anfängt zu blinken, wenn ein Ladevorgang läuft. Dieser Zustand wird vom Anwender herbeigeführt, wobei die Parametrierung bezüglich der projektierten Diagnoseschnittstellen (SS4, SS5x-Schnittstellenmodule in CS7-Baugruppe, CP50M0/CP51M1) ihre Gültigkeit behält.

**Download im Betriebszustand RUN**

Es ist auch möglich, im Betriebszustand RUN über jeden Dienst ein Download zu fahren, dann aber nur mit deutlich längeren Ladezeiten (Laden parallel zu zyklischer Bearbeitung).

Der Übergang in den Zustand "Anwenderstop" ist nur aus dem Betriebszustand RUN heraus durch ein explizites Anfordern des Anwenders über eine Service-Schnittstelle (lokal bzw. projektiert) möglich. In diesem Zustand sind alle projektierten und die lokale Service-Schnittstelle weiterhin verfügbar, d.h. es kann über alle Service-Schnittstellen Diagnose betrieben bzw. ein Download gefahren werden (erforderlich, wenn mehrere PCs am Rahmen hängen).

**2.1.5 Beschreibung und Verwendung von Signaltransporten**

Unter Signaltransport ist der Austausch von Daten zwischen verschiedenen Bausteinen zu verstehen.

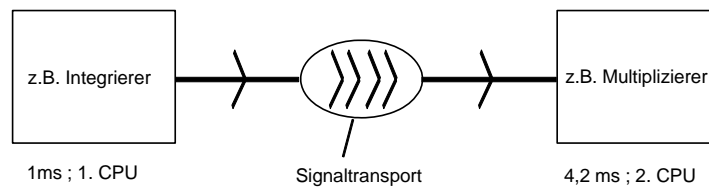


Bild 2-3 Datenaustausch zwischen zwei Tasks

### 2.1.5.1 Datenkonsistenz

Bei Verschaltungen zwischen unterschiedlichen zyklischen Tasks gewährleistet SIMATIC TDC/SIMADYN D die Konsistenz aller übertragenen Daten, d.h. alle aus einer Task übertragenen Daten stammen aus dem selben Rechenzyklus dieser Task. Alle im Laufe eines Abtastzyklus berechneten Werte werden am Ende der Task "exportiert". Beim Start einer Task werden die benötigten Werte "importiert", wobei sichergestellt wird, daß es keine zeitliche Überschneidung zwischen dem Lesen und dem Schreiben der Werte gibt (Puffersystem). Da bei diesem Mechanismus die Entstehung von Totzeiten unvermeidlich ist, sollte ein Signalweg nicht ohne Grund über mehrere Tasks und CPU's geführt werden.

Folgende Fälle des Signaltransports werden unterschieden:

- Datenaustausch innerhalb der gleichen Task einer CPU
- Datenaustausch zwischen verschiedenen Tasks einer CPU
- Datenaustausch zwischen zyklischen Tasks von mehreren CPUs
- Datenaustausch zwischen Alarmtasks mehrerer CPU's

### 2.1.5.2 Datenaustausch innerhalb der gleichen Task einer CPU

Jedem Funktionsbausteinausgang ist im System eine Speicherzelle zugeordnet, in der sein während der Bausteinbearbeitung aktuell berechneter Wert abgelegt wird. Alle Eingänge, die mit Ausgängen in der gleichen Task verschaltet sind, holen ihre Werte aus den entsprechenden, dem jeweils verschalteten Ausgang zugeordneten Speicherzellen. Um Totzeiten zu vermeiden, sollten die Bausteine einer Task möglichst entsprechend dem "Signalfluß" berechnet werden, d.h. zuerst wird der Baustein berechnet, dessen Ausgänge einem nachfolgenden Baustein als Eingänge dienen usw.

### 2.1.5.3 Datenaustausch zwischen verschiedenen Tasks einer CPU

Der Datenaustausch zwischen verschiedenen Tasks einer CPU erfolgt über ein Puffersystem, um die Datenkonsistenz gewährleisten zu können (siehe Kapitel "Datenkonsistenz"). Allerdings ist beim Datenaustausch von einer schnelleren in eine langsamere Task zu beachten, daß Wertänderungen in der langsamen Task nicht erfaßt (siehe Bild 1.6-2) bzw. erst verspätet erfaßt (siehe Bild 1.6-3) werden können. Sofern dies verhindert werden soll, muß die Projektierung z.B. mit Impulsverlängerer-Funktionsbausteinen angepaßt werden.

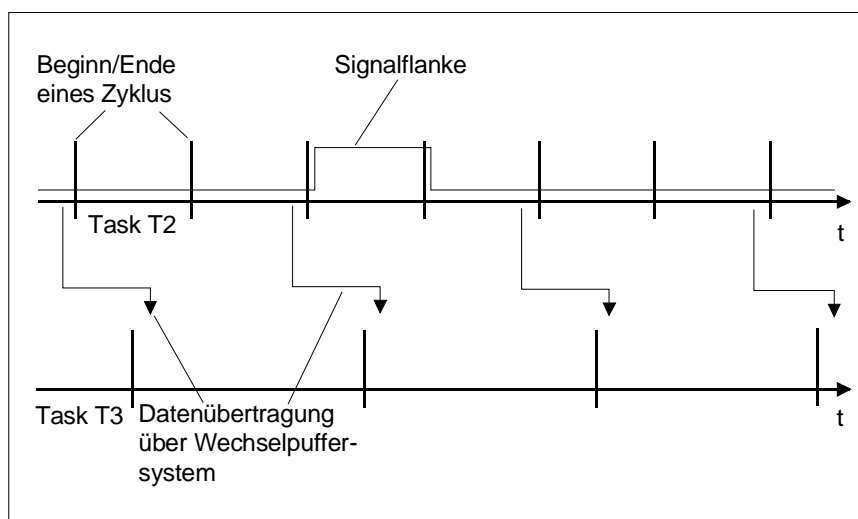


Bild 2-4 Signal in Task 3 nicht erfaßt

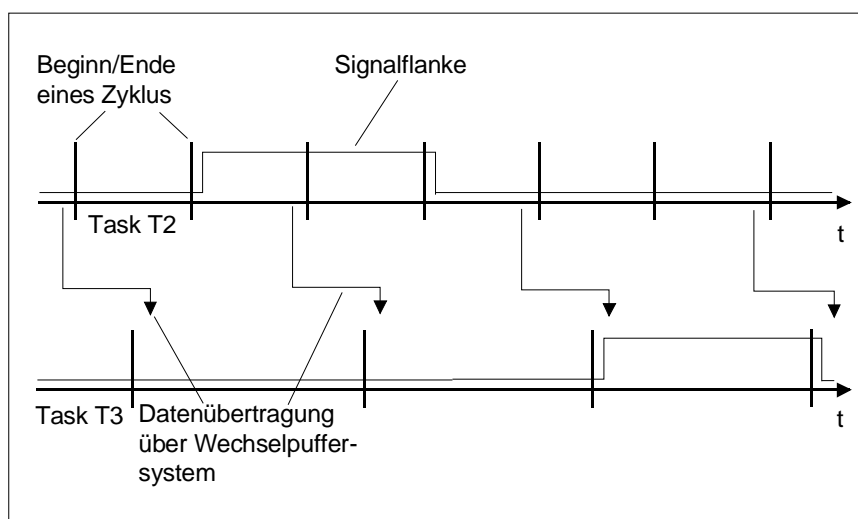


Bild 2-5 Signal verspätet erfaßt



#### 2.1.5.4 Datenaustausch zwischen zyklischen Tasks von mehreren CPUs

Der Signaltransport zwischen den CPUs erfolgt durch die Koppelspeicherbaugruppen MM3, MM4 und MM11 (SIMADYN D) oder CP50M0/CP51M1 (SIMATIC TDC). Die Verschaltung von Funktionsbausteinen, die auf unterschiedlichen CPUs innerhalb derselben Station ablaufen, wird durch \$-Signale geregelt (Menüpunkt "Einfügen-Verbindung zu Operand " im CFC-Editor). Zur Projektierung eines \$-Signals sind folgende Angaben notwendig:

- Signalname
- Typ
- Buszuordnung.

Der Typ des Dollarsignals bestimmt, ob die Datenübertragung

- konsistent ("Standard") oder
- inkonsistent ("Fast-\$-Signal")

erfolgen soll.

Bei einem Fast-\$-Signal kann der Verbraucher immer auf einen aktuellen Wert zugreifen. Die beim Signaltransport entstehende Totzeit ist dann minimal, wenn Erzeuger und Verbraucher in der gleichen Task projiziert sind und evtl. die Tasks synchronisiert werden (siehe Kapitel "Bedeutung und Einsatz der CPU-Synchronisation").

Die Buszuordnung legt fest, ob die Datenübertragung über den L-Bus oder den C-Bus erfolgen soll.

---

**HINWEIS**

Wenn auf den CPUs eines Baugruppenträgers zeitkritische Funktionen bearbeitet werden, dann berücksichtigen Sie folgende Regeln:

- Beschränken Sie die Anzahl der \$-Signale auf ein Minimum.
  - Wählen Sie für die \$-Signale, die in Alarmtasks projiziert werden, den L-Bus.
  - Wählen Sie für \$-Signale, die nicht in Alarmtasks projiziert werden, den C-Bus.
  - Projektieren Sie möglichst alle Kommunikationsverbindungen der Rahmenkopplung auf einer oder maximal zwei CPUs des Baugruppenträgers.
  - Projektieren Sie die CPUs mit den projizierten Kommunikationsverbindungen der Rahmenkopplung möglichst so, daß keine weiteren CPUs zwischen diesen CPUs und der Rahmenkopplungs-Baugruppe stecken.
-

### 2.1.5.5 Datenaustausch zwischen Alarmtasks mehrerer CPU's

#### Fast-\$-Signal

Ein Fast-\$-Signal muß auf jeden Fall dann projiziert werden, wenn das Signal in einer Alarmtask erzeugt oder verbraucht wird, da ein Alarmereignis zu einem beliebigen Zeitpunkt und nur einmalig auftreten kann und daher die Konsistenzmechanismen umgangen werden müssen, um Datenverlust zu vermeiden. Hier könnte ein Konflikt zwischen der Forderung nach Datenkonsistenz und kleiner Totzeit entstehen, der je nach Anwendung entschieden werden muß.

#### HINWEIS

In jedem Fall sollte geprüft werden, ob eine fehlende Datenkonsistenz nicht zu Problemen führen kann.

Die Datenkonsistenz kann durch das "Durchschleifen der Signale" durch eine zyklische Task auf der CPU-Baugruppe, auf der die Alarmtask gerechnet wird, erreicht werden. Die Berechnung der Totzeit ist in der folgenden Tabelle dargestellt.

Zeitspanne	Berechnung
Minimalwert	$1 * T_x$
Maximalwert	$2 * T_x + 1 * T_y + 1 * T_{alarm}$

Tabelle 2-8 Zeitwertberechnung der Totzeiten

- $T_x$  = Abtastzeit der zyklischen Task, durch die die Signale durchgeschleift werden,
- $T_y$  = Abtastzeit der erzeugenden/verbrauchenden CPU und
- $T_{alarm}$  = maximale Alarmwiederholzeit der Alarmtask.

### 2.1.5.6 Minimierung von Totzeiten

Zur Minimierung der Totzeiten kann unter Umgehung der Datenkonsistenz ein Signal direkt übertragen werden. Es wird dann direkt auf den Ausgang des erzeugenden Bausteins "verdrahtet". Zu diesem Zweck stehen zwei Projektierungsmöglichkeiten zur Verfügung:

- Pseudokommentar @DATX bei Verschaltungen zwischen Tasks einer CPU
- Fast-\$-Signale bei Verschaltungen zwischen mehreren CPU's

### 2.1.5.7 Bearbeitungsreihenfolge innerhalb eines CPU-Grundtaktes

Mit dem CPU-Grundtakt  $T_0$  wird der Aufgabenverwalter (siehe "Veranschaulichung der Arbeitsweise des Aufgabenverwalters") des Betriebssystems gestartet. Dieser entscheidet, welche Tasks zu starten sind ( $T_1$  und maximal eine weitere  $T_n$ , mit  $T_n$  aus  $\{T_2...T_5\}$ ).

Innerhalb der Taskbearbeitung können prinzipiell folgende Anteile abzuarbeiten sein:

- Pufferumschaltung für die zu startenden Aufgaben (T1 und gegebenenfalls eine weitere Task Tn)
- Systemmode der Bausteine in T1 entsprechend der Bausteinreihenfolge (siehe Kapitel "Bedeutung und Einsatzmöglichkeiten des Prozeßabbildes")
- Systemmode der Bausteine in Tn entsprechend der Bausteinreihenfolge (siehe Kapitel "Bedeutung und Einsatzmöglichkeiten des Prozeßabbildes");
- **Import von Signalverschaltungen in T1 und Normalmode T1**
- **Export von Signalverschaltungen aus T1**
- **Import von Signalverschaltungen in Tn und Normalmode Tn**
- **Export von Signalverschaltungen aus Tn.**

Die für die Signaltransporte relevanten Anteile sind fett dargestellt.

### 2.1.5.8 Verschaltungsänderungen und Grenzzahl von Verschaltungen

#### **Verschaltungsänderungen während der Testphase der Projektierung**

Änderungen von Verschaltungen über Taskgrenzen hinweg sind durch den Testmodus des CFC-Editors nur eingeschränkt möglich. Der Testmodus ist eine Betriebsart des CFC-Editors zum Testen und Optimieren des Anwenderprogramms, das bereits online auf der CPU abläuft.

Es stehen bei solchen Änderungen mit dem Service nur begrenzte Reserven für zusätzliche Verschaltungen zur Verfügung. Die maximal zur Verfügung stehende Reserve von zusätzlichen Verschaltungen beträgt

- 20% der bereits projektierten Anzahl von Verschaltungen jedoch
- mindestens 10.

#### **Beispiel:**

Von der zyklischen Task T2 zur zyklischen Task T3 existieren bereits 5 Verschaltungen. Dann steht für Verschaltungsänderungen von T2 nach T3 eine Reserve von 10 Verschaltungsänderungen zur Verfügung. Die 20% von 5 = 1 zusätzliche Verschaltung aus Regel 1 würde sonst die Minimalreserve von 10 zusätzlichen Verschaltungen aus Regel 2 unterschreiten.

Bei 100 bestehenden Verschaltungen stehen zusätzlich 20 Reserveverschaltungen zur Verfügung, da 20% von 100 = 20.

#### **Grenzwerte für die Anzahl von Verschaltungen**

Unterschieden werden Verschaltungen innerhalb einer Task, zwischen Tasks einer CPU und zwischen mehreren CPUs einer Station. Im Betrieb

mit mehreren CPU's wird zusätzlich zwischen Standard- und Fast-\$-Signalen unterschieden.

Bei Verschaltungen zwischen Tasks einer CPU kommt das prozessorlokale Wechselpuffersystem zum Einsatz. Eine Beschränkung der maximalen Anzahl von Verschaltungen wird durch den Ausbau des Hauptspeichers begrenzt.

Verbindungen zwischen mehreren CPU's einer Station werden über die Koppelspeicherbaugruppen abgewickelt. Die Anzahl der möglichen Verschaltungen ist von der eingesetzten Koppelspeicherbaugruppe und den verwendeten Signaltypen abhängig.

**Weitere Informationen**

zu Koppelspeicherbaugruppen siehe Handbuch "SIMATIC TDC/SIMADYN D Hardware"

Für die Baugruppe MM11 mit 64 Kbyte Speicher je L- und C-Bus ergibt sich bei Verwendung von:

Signaltyp	Bytes/Verschaltung	Anzahl Verschaltungen
Fast-\$-Signalen	4	ca. 16000 je Bustyp
Standardsignal	max. 36 (Anzahl CPU's + 1)* 4)	min. 1800 je Bustyp

Tabelle 2-9 Berechnung der maximalen Anzahl der Verschaltungen

**HINWEIS**

Werden normale und Fast-Verschaltungen kombiniert, ergibt sich eine entsprechend kleinere Anzahl.

**2.1.6 Bedeutung und Einsatzmöglichkeiten des Prozeßabbildes**

Ein Prozeßabbild ist eine Momentaufnahme aller Schnittstellensignale vom Prozeß zu Beginn einer zyklischen Task.

**Notwendigkeit der Datenkonsistenz**

Bei einem digitalen Steuer- und Regelungssystem besteht die Notwendigkeit, die Schnittstellensignale zu den einzelnen Prozessen konsistent zu verarbeiten. Als Schnittstellensignale werden die binären und analogen Eingangs- oder Ausgangssignale einer Hardwarebaugruppe bezeichnet.

Die Eingangssignale der verschiedenen Tasks müssen während eines Rechenzyklusses konstant gehalten werden, da sonst Schnittstellensignaländerungen während der Bearbeitung einer Task und Laufzeiten der einzelnen Funktionsbausteine das Ergebnis eines Rechenzyklus unüberschaubar beeinflussen.

Im sogenannten Prozeßabbild, realisiert durch den Systemmode der Funktionsbausteine zu Beginn einer Taskbearbeitung, werden die Daten der Hardwareschnittstellen bearbeitet.

Mit dem CPU-Grundtakt T0 wird der Aufgabenverwalter (siehe Kapitel "Veranschaulichung der Arbeitsweise des Aufgabenverwalters") des Betriebssystems gestartet. Dieser entscheidet, welche Tasks zu starten sind (T1 und maximal eine weitere Tn,

mit Tn aus {T2...T5}).

### **Taskbearbeitung**

Innerhalb der Taskbearbeitung können prinzipiell folgende Anteile abzuarbeiten sein:

- Pufferumschaltung für die zu startenden Aufgaben (Task 1 T1 und ggf. eine weitere Task Tn)
- **Systemmode der Funktionsbausteine in T1 entsprechend der Bausteinreihenfolge**
- **Systemmode der Funktionsbausteine in Tn entsprechend der Bausteinreihenfolge**
- Import von Signalverschaltungen in T1 und Normalmode T1
- Export von Signalverschaltungen aus T1
- Import von Signalverschaltungen in Tn und Normalmode Tn
- Export von Signalverschaltungen aus Tn

Die für das Prozeßabbild relevanten Anteile sind fett dargestellt; die anderen Anteile siehe Kapitel "Beschreibung und Verwendung von Signaltransporten".

### **2.1.6.1 Realisierung des Prozeßabbildes**

#### **Systemmode**

Der Systemmode dient der Realisierung des Prozeßabbildes vor der Berechnung der Task. Im folgenden Bild wird dargestellt, in welcher Abfolge im zyklischen Betrieb (CPU im Zustand RUN) die Funktionsbausteine im System- und Normalmode gerechnet werden. In diesem Beispiel werden die Funktionsbausteine 10 und 30 im Systemmode im Rahmen des Prozeßabbildes berechnet, um die Ergebnisse im Normalmode anschließend konsistent zu verwenden.

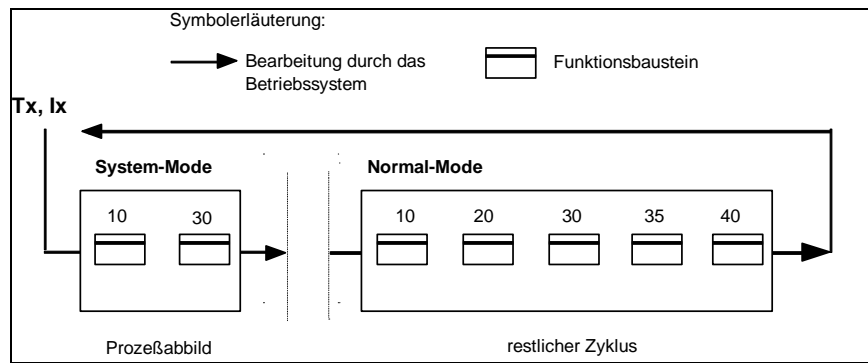


Bild 2-6 Ablauf der Funktionsbausteinberechnung im System- und Normalmode

Der Systemmode startet sofort nach dem Aktivierungsereignis (Prozessalarm o. Grundtakt), um ein Echtzeit-Prozeßabbild zu erzeugen. Dabei ist die Abarbeitung zwischen dem Einsprung ins Betriebssystem bis zum Ende des Systemmodes nur durch höherpriorie Systemmodes unterbrechbar. Es werden u.a. Funktionsbausteine mit Zugriff auf die Peripherie berechnet.

### 2.1.6.2 Prozeßabbild bei zyklischen Tasks

#### Eingabebausteine mit Systemanteil

Bei Eingabebausteinen, die einen Systemanteil besitzen oder deren Systemanteil aktiviert ist, werden die Eingangssignale von der Hardware eingelesen und zwischengespeichert. Die Auswertung der Signale erfolgt im Normalmode der Bausteine des gleichen Zyklus (siehe Bild 2-7).

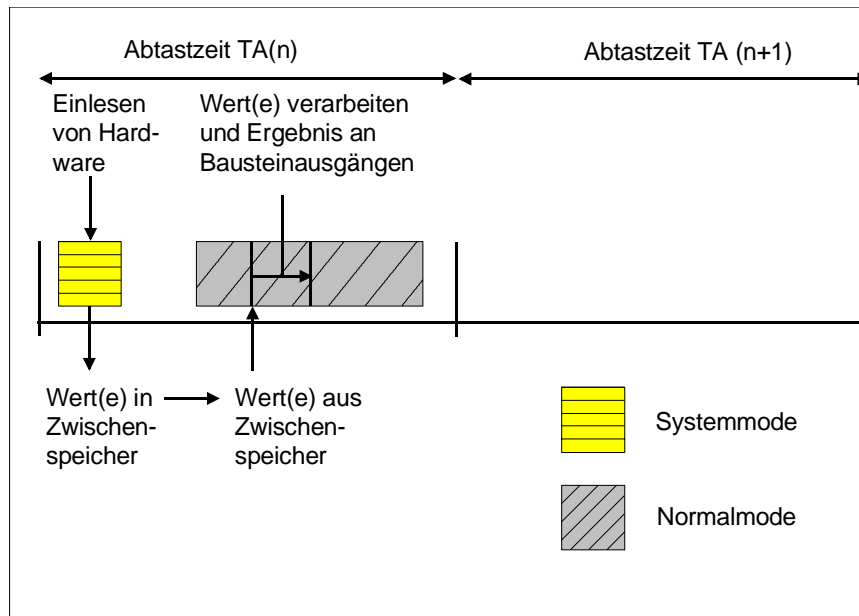


Bild 2-7 Ablauf des Systemmodes bei Eingabebausteinen

### Ausgabebausteine mit Systemanteil

Bei Ausgabebausteinen, die einen Systemanteil besitzen bzw. deren Systemanteil aktiviert ist, werden im Normalmode des vorangegangenen Zyklus die auszugebenden Signale entsprechend der Bausteinfunktion und den aktuellen Anschlußwerten ermittelt und zwischengespeichert. Die Ausgabe an die Hardware erfolgt im Systemmode, zu Beginn des nächsten Abtastzyklus (siehe Bild 2-8).

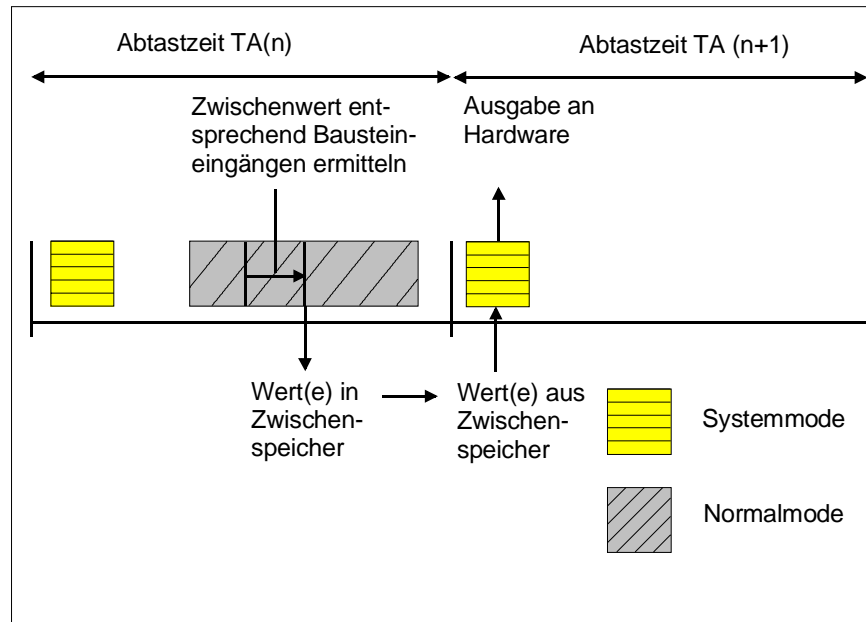


Bild 2-8 Ablauf des Systemmodes bei Ausgabebausteinen

Da sich der Systemanteil größtenteils auf die Ein- und Ausgabe von Hardwaresignalen beschränkt, erfolgt die Bearbeitung des Systemmodes innerhalb weniger Mikrosekunden.

Bei einigen Ein-/Ausgabebausteinen kann über den Bausteineingang „DM“ gesteuert werden, ob eine Ein-/Ausgabe im Systemmode oder im Normalmode vorgenommen werden soll. Bei einer Berechnung im **Normalmode** werden die Schnittstellensignale an den Bausteinen unter Umgehung des Prozeßabbildes innerhalb des **Normalmode** berechnet. Bei Eingangsbausteinen werden dabei die Signale unmittelbar vor ihrer Berechnung eingelesen und bei Ausgangsbausteinen unmittelbar nach ihrer Berechnung ausgegeben.

### 2.1.6.3 Prozeßabbild bei Alarmtasks

Für eine Alarmtask gilt prinzipiell das gleiche Verhalten wie bei einer zyklischen Task.

#### Arbeitsweise der Alarmtask

Eine Alarmtask kann eine laufende zyklische Task im Normalmode unterbrechen, kann jedoch nicht durch zyklische Aufgaben unterbrochen werden. Dadurch kann, z.B. bei längeren Rechenzeiten einer Alarmaufgabe, der Start der zyklischen Aufgaben und damit die Ausgabe an die Hardware verzögert werden, da bei Ausgabebausteinen im Systemmode die Signalausgabe an die Hardware erst nach dem Start der nächsten Aufgabe erfolgt.

Zudem ist die Verwendung von Ein-/Ausgabebausteinen im Systemmode innerhalb einer Alarmtask bei nicht quasizyklischen Alarmen genau zu prüfen. Hier erfolgt die Ausgabe erst nach dem nächsten Alarmereignis, dessen Zeitpunkt aber unbekannt ist. Als Abhilfe kann bei bestimmten Ein-/Ausgabebausteinen, gesteuert über einen Bausteineingang, die Ein-/Ausgabe im Normalmode vorgenommen werden.

### 2.1.7 Bedeutung und Einsatz der CPU-Synchronisation

#### Projektierung der CPU-Synchronisation

Die CPU-Synchronisation wird im Programmteil HWKonfig projektiert. Im SIMATIC Manager wird das Verzeichnis der entsprechenden SIMATIC TDC/SIMADYN D-Station geöffnet und im rechten Fensterteil durch einen Doppelklick auf das Hardware-symbol aktiviert sich HWKonfig. Markieren Sie nun die gewünschte CPU-Baugruppe. Im Menüpunkt **Bearbeiten > Objekteigenschaften** gibt es für die Synchronisation des Grundtaktes der CPU's und der Alarmtasks getrennte Dialogfenster.

#### Synchronisiermechanismen

Folgende Synchronisiermechanismen werden von SIMATIC TDC/SIMADYN D zur Verfügung gestellt:

- Uhrzeitsynchronisation
- Synchronisation des eigenen Grundtaktes auf den Grundtakt einer Master-CPU
- Synchronisation des eigenen Grundtaktes auf Alarmtasks einer Master-CPU
- Synchronisation von eigenen Alarmtasks auf Alarmtasks einer Master-CPU
- Synchronisation von mehreren Stationen
- Reaktion bei Ausfall der Synchronisation
- Projektierung der CPU-Grundtakt-Synchronisation
- Projektierung der Alarmtask-Synchronisation



### 2.1.7.1 Uhrzeitsynchronisation

Die Echtzeituhren aller CPUs in einer SIMATIC TDC/SIMADYN D-Station werden auf die Uhr der CPU auf Steckplatz 1 synchronisiert, um ein auseinanderlaufen der verschiedenen CPU-Uhren zu verhindern. Diese Synchronisation wird alle 10 s automatisch durchgeführt.

### 2.1.7.2 Synchronisation des eigenen Grundtaktes auf den Grundtakt einer Master-CPU

Der Grundtakt kann von einer CPU auf den L- und/oder C-Bus des Baugruppenträgers geschaltet und von anderen CPUs der Station oder bei mehreren SIMATIC TDC/SIMADYN D-Stationen, die mit Hilfe der Rahmenkopplung bzw. GDM-Kopplung gekoppelt werden, auch stationsübergreifend empfangen werden. Bei der Empfänger-CPU ist die Projektierung einer Verschiebung der Grundabtastzeit gegenüber der Grundabtastzeit des Senders möglich. Diese Zeitverschiebung ist dann auch online im Betriebszustand RUN der CPU mit dem Funktionsbaustein DTS änderbar.

### 2.1.7.3 Synchronisation des eigenen Grundtaktes auf Alarmtasks einer Master-CPU

Es besteht die Möglichkeit, am Beginn bzw. am Ende einer Alarmtask einer Sender-CPU einen L- oder C-Businterrupt auszulösen. Dieser kann auf einer oder mehreren anderen Empfänger-CPU's empfangen werden, um dort den Grundtakt zu erzeugen.

### 2.1.7.4 Synchronisation von eigenen Alarmtasks auf Alarmtasks einer Master-CPU

Zur Synchronisation einer Alarmtask ist es möglich, den am Beginn bzw. am Ende einer Alarmtask von einer Sender-CPU ausgelösten L- oder C-Businterrupt zu verwenden. Dieser Interrupt kann auf einem oder mehreren anderen Empfänger-CPU's empfangen werden, um dort eine alarmgesteuerte Task zu starten.

### 2.1.7.5 Synchronisation von mehreren SIMATIC TDC/SIMADYN D-Stationen

Zur stationsübergreifenden Synchronisation der Grundabtastzeit stehen die Baugruppen CS12, CS13 und CS14 (Rahmenkopplung Master) und CS22 (Rahmenkopplung Slave) (SIMADYN D) bzw. CP52M0, CP52IO, CP52A0 (SIMATIC TDC) und CP53M0 (SIMATIC TDC mit SIMADYN D) zur Verfügung. Dabei werden die Bussysteme der beiden Stationen über die Koppelbaugruppen verbunden.

#### **Weitere Informationen**

zur Thematik Synchronisation siehe Handbuch "System- und Kommunikationsprojektierung D7-SYS".

### 2.1.7.6 Reaktion bei Ausfall der Synchronisation

Die Grundtaktüberwachung auf synchronisierten Empfänger-CPU's erfolgt mittels eines Hardware-Timers. Bleibt der Sendetakt 4 Zyklen lang aus, übernimmt der auf der CPU-Baugruppe vorhandene Grundtakttimer die Grundtakterzeugung. Zugrundegelegt wird die in HWKonfig projektierte Grundabtastzeit, welche in diesem Fall als Ersatzabtastzeit dient. Die Umschaltung auf den eigenen Grundtakt der CPU wird durch ein blinkendes "E" auf der Siebensegmentanzeige der CPU-Baugruppe signalisiert und im Fehlerfeld vermerkt. Mittels des Funktionsbaustein "DTS" besteht die Möglichkeit, bei Wiedereinsetzen der externen Taktquelle, den gesendeten Grundtakt auf dem Empfänger erneut zu nutzen.

### 2.1.7.7 Projektierung der CPU-Grundtakt-Synchronisation

Die Projektierung wird im Dialogfenster "Grundtakt" des HWKonfig eingestellt (siehe Kapitel "Bedeutung und Einsatz der CPU-Synchronisation"). Als Voreinstellung ist die Synchronisation abgeschaltet.

#### Grundtakt selbst erzeugen

Soll die CPU einen Grundtakt selbst erzeugen, müssen im Dialogfeld „Grundtakt“ (siehe Bild 1.8-1) folgende Einstellungen vorgenommen werden:

- Button „Erzeugen“ mit Mausclick aktivieren.
- Eingabe der gewünschten Grundabtastzeit von 0,1 bis 16 ms.

Im unteren Fensterteil kann zudem noch festgelegt werden, ob die ausgewählte CPU als Grundtaktquelle genutzt werden soll. Zu diesem Zweck ist der entsprechende Bus einzustellen. Vorbelegung ist „nein“.

#### Grundtakt auf eine Quelle synchronisieren

Soll der Grundtakt auf eine andere Quelle synchronisiert werden, benötigt HWKonfig folgende Einstellungen:

- Button „Synchronisieren“ mit Mausclick aktivieren.
- Auswahl der gewünschten Quelle aus einer Liste, z.B.

L-Bus- oder C-Bus-Grundtakt

L-Bus- oder C-Bus-Alarm (SIMADYN D)

Bus-Alarm (SIMATIC TDC)

- Eingabe einer Ersatzabtastzeit von 0,1 bis 16 ms.

Vorbelegung = 1,0 ms

- Eingabe einer eventuell gewünschten Verzögerungszeit der Synchronisation von 0,1 ms bis zur Ersatzabtastzeit.

Vorbelegung der Verzögerungszeit ist "keine"

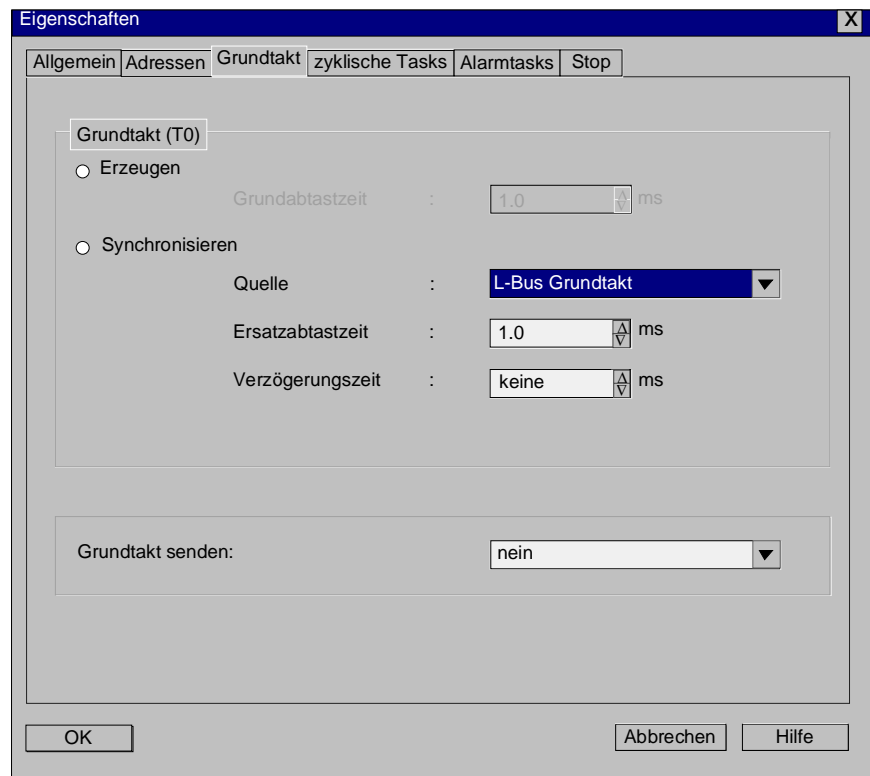


Bild 2-9 Dialogfeld Grundtakt in HWkonfig

### 2.1.7.8 Projektierung der Alarmtask-Synchronisation

Die Projektierung wird im Dialogfeld "Alarmtasks" in HWKonfig eingestellt (siehe Kapitel "Bedeutung und Einsatz der CPU-Synchronisation"). Als Voreinstellung ist die Synchronisation abgeschaltet, d.h. es sind keine Prozeßalarme definiert, und es wird kein Bus-Alarm gesendet.

#### Einstellen der Alarmtask-synchronisation

- Mit einem Mausklick wird festgelegt, welche der 8 möglichen Alarmtasks I1 - I8 definiert werden sollen.
- Auswahl der gewünschten Quelle der definierten Prozeßalarme aus einer Liste, z.B.

C-Bus Alarm oder

CPU Counter C1 oder C2

- Eingabe einer **Ersatzabtaastzeit** von 0,1 bis 16 ms.

#### CPU als Alarmquelle für den Baugruppenträger

Im unteren Fensterteil wird vorgegeben, ob die ausgewählte CPU als Prozeßalarmquelle für den Baugruppenträger arbeiten soll. In diesem Fall muß eine der definierten Alarmtasks I1 - I8 ausgewählt und auf den L- und/oder C-Bus gesendet werden. Dabei kann noch gewählt werden, ob das Senden zu Beginn oder am Ende der Alarmtaskbearbeitung erfolgt.

**Senden zu Beginn der Bearbeitung**

Das Senden zu Beginn ist dann sinnvoll, wenn ein synchroner, verzögerungsfreier Start von Alarmtasks auf mehreren CPU-Baugruppen gewünscht wird. Es kann hier aber z.B. vorkommen, daß die Alarmtask auf der Empfänger-CPU-Baugruppe vor der Alarmtask auf der Sender-CPU-Baugruppe beendet wird, weil die Abarbeitung der Sendetask durch einen höherpriorien Interrupt blockiert wird.

**Senden am Ende der Bearbeitung**

Wird am Ende gesendet, ist sichergestellt, daß die Task auf der Empfangsseite keinesfalls vor Beendigung der Sendetask gestartet wird. Die zweite Möglichkeit kann sinnvoll angewendet werden, wenn ein Datentransfer von der Sende- zur Empfangstask stattfindet.

**2.1.7.9 Beispiel einer Synchronisationskonstellation**

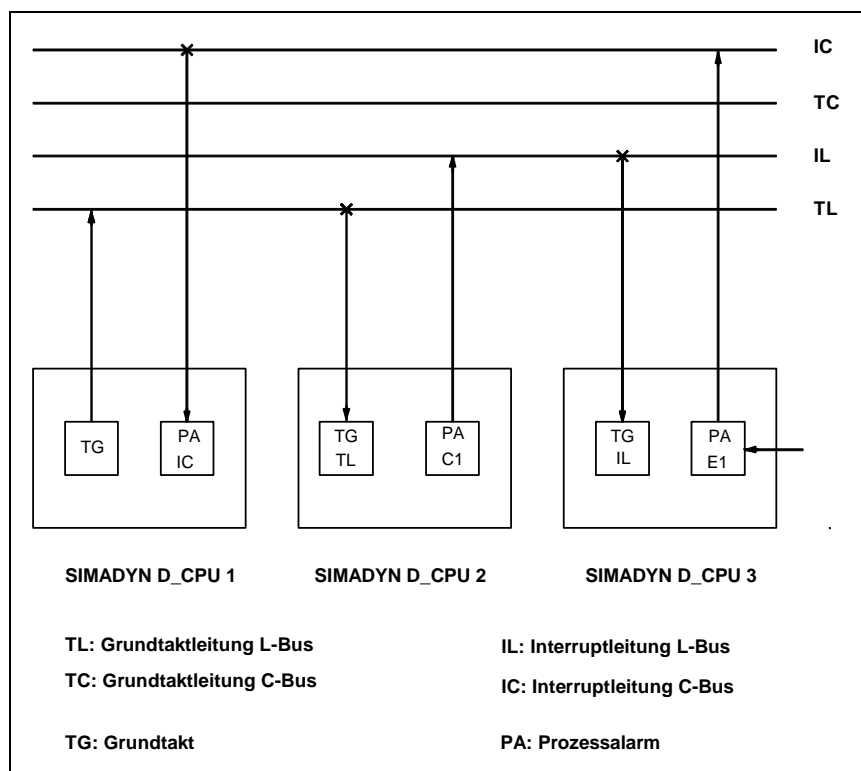


Bild 2-10 Synchronisationskonstellation

**Beschreibung**

Im Bild 2-10 stellt CPU 1 ihren Grundtakt als Sender auf dem L-Bus zur Verfügung. Außerdem wird der C-Businterrupt von einer alarmgesteuerten Task der CPU 1 als Alarmereignis verwendet.

CPU 2 bezieht ihren Grundtakt von der Grundtaktleitung des L-Busses und schaltet den Interrupt vom Counter C1 (Konfiguration mit Funktionsbaustein PAC) auf die L-Bus-Interruptleitung durch.

CPU 3 bezieht ihren Grundtakt von der L-Bus-Interruptleitung und schaltet den über die binäre Eingabe (Konfiguration mit Funktionsbaustein PAI) erhaltenen Interrupt auf die C-Bus-Interruptleitung durch.

## 2.1.8 Bedeutung der Prozessorauslastung

### 2.1.8.1 Ermittlung der ungefähren Prozessorauslastung

Der CFC ermittelt beim Compilieren einen Wert für die Rechenzeitauslastung der CPU. Dazu wird auf eine Liste zugegriffen, in der für jeden Funktionsbausteintyp die Rechenzeit des Bausteins eingetragen ist. Diese Rechenzeiten wurden bei der Entwicklung der Bausteine für den "Worst-Case"-Fall ermittelt und sind in der Benutzerdokumentation Funktionsbaustein-Bibliothek (ab Ausgabe Herbst 97) zu finden.

Bei einigen Funktionsbausteinen, insbesondere bei Bausteinen, die auf Hardware zugreifen, würde der Worst-Case-Fall meist zu einem zu hohen Zeitverbrauch führen, deshalb wird dort mit einer typischen Rechenzeit (z.B. für mittelgroße Busbelastung) gearbeitet. Ausgehend von diesen Richtwerten, kann bei einigen Funktionsbausteintypen die tatsächliche Rechenzeit stark schwanken.

Die im Bausteinkatalog eingetragene Rechenzeit gibt die typische Rechenzeit des Bausteins in  $\mu\text{s}$  auf einer PM5 an. Dieser Wert kann vor allen Dingen bei Kommunikationsbausteinen, abhängig von der Menge der zu transportierenden Daten, stark von der tatsächlich benötigten Zeit abweichen.

Nach dem Übersetzen der Pläne einer CPU mit dem CFC-Editor über den Menüpunkt **Plan > Übersetzen** wird in einem Informationsfenster bzw. im Fehlerfenster der Pfad eines MAP-Listings angegeben. Das MAP-Listing finden Sie im File Die im MAP-Listing eingetragene Prozessorauslastung ist aus den dargestellten Gründen ein Anhaltswert, der im Normalfall auf etwa  $\pm 10\%$  genau sein wird.

### 2.1.8.2 Ermittlung der genauen Prozessorauslastung

**Funktionsbaustein PSL** Die genaue CPU-Auslastung läßt sich nur durch die Projektierung des Funktionsbausteines vom Typ PSL "Permanent System Load" ermitteln. Bei der zu untersuchenden CPU wird der Baustein PSL in eine beliebige zyklische Task der CPU projiziert.

Der PSL-Baustein besitzt 5 Ausgänge (Y1..5), welche die aktuelle Auslastung der einzelnen Tasks in Form eines Belastungsfaktors anzeigt. Der dargestellte Faktor sollte den Wert 1,0 (100%) nicht überschreiten. Werte über 1,0 signalisieren eine überlastete CPU.

Zusätzlich besitzt der PSL-Baustein 5 Eingänge (T1..5), an denen für jede Task eine zusätzliche Belastung in Millisekunden (ms) simuliert werden kann. Es kann an den Ausgängen dann abgelesen werden, wie sich eine solche Belastung auf die Auslastung der einzelnen Tasks auswirkt. Die Auslastung wird dadurch bestimmt, daß die Laufzeit der Tasks gemessen und durch die eigene Abtastzeit dividiert wird. In die Laufzeit einer Task fallen höherpriore Tasks, welche die Laufzeit verlängern und die Auslastung scheinbar erhöhen (siehe Bild 1.9-1).

Deshalb kann durch Addieren dieser Werte auf eine Gesamtbelastung nicht geschlossen werden.

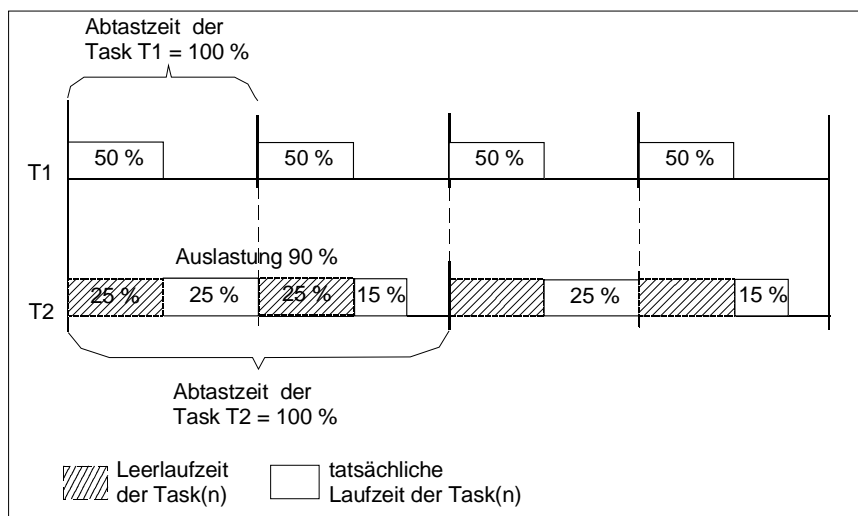


Bild 2-11 Laufzeitenberechnung

### 2.1.8.3 Veranschaulichung der Arbeitsweise des Aufgabenverwalters

Die Arbeitsweise des Aufgabenverwalters wird in diesem Kapitel am Bild 2-12 dargestellt.

Können aufgrund einer kleinen Rechenzeitbelastung alle Tasks innerhalb einer Grundabtastzeit beendet werden, ergibt sich die Darstellung des 1. Zyklus.

Kann eine Task aufgrund erhöhter Rechenzeitbelastung nicht mehr in einer Grundabtastzeit beendet werden, wird sie über die folgenden Grundzyklen bis zum Beenden fortgesetzt. Die Tasks mit kleinen Abtastzeiten werden vor Tasks mit großen Abtastzeiten beendet, d.h. T1 vor T2 vor T3 vor T4 vor ... Diese Aufteilung ist solange zulässig, also ohne Zyklusfehler, solange die geforderten Abtastzeiten eingehalten werden können (siehe 2. und 3. Zyklus).

#### Zyklusfehler

Wird die Rechenzeitbelastung noch größer, tritt irgendwann bei der Task mit der größten Abtastzeit ein Zyklusfehler auf. Das bedeutet, daß innerhalb der projektierten Abtastzeit die Summe der Funktionsbausteine nicht komplett zu Ende gerechnet werden kann.

#### HINWEIS

Überschreitet man eine bestimmte Anzahl von Zyklusfehlern, wird eine Fehlerkennung "E" gesetzt und in der Siebensegmentanzeige auf der Frontplatte der CPU angezeigt, falls dies zu diesem Zeitpunkt der höchstprioräre Fehlerzustand der CPU ist.

Neben den projektierbaren Alarmtasks werden die zyklischen Tasks vor allem von Interrupts der Kommunikation unterbrochen. Diese Interrupts sorgen dafür, daß z.B. die über die seriellen Schnittstellen empfangenen und zu sendenden Daten rechtzeitig vor dem Eintreffen neuer Daten bearbeitet werden. Solche Sende- und Empfangs-Interrupts können quasi unabhängig von der projektierten Zykluszeit der entsprechenden Kommunikationsbausteine an nahezu jedem beliebigen Zeitpunkt auftreten. Dadurch und durch das willkürliche Auftreten von Alarmtasks kann bei sehr hoher Prozessorauslastung prinzipiell jede zyklische Task durch einen Aufgabenstau zu einem oder mehreren Zyklusfehlern führen.

Dies ist besonders dann zu beachten, wenn

- die Auslastung durch die Task mit der kleinsten Abtastzeit schon sehr hoch ist und
- die in dieser Aufgabe gerechneten Funktionen sehr empfindlich bezüglich sporadischer Ausfälle von Abtastzyklen sind (z.B. Wegregelungen).

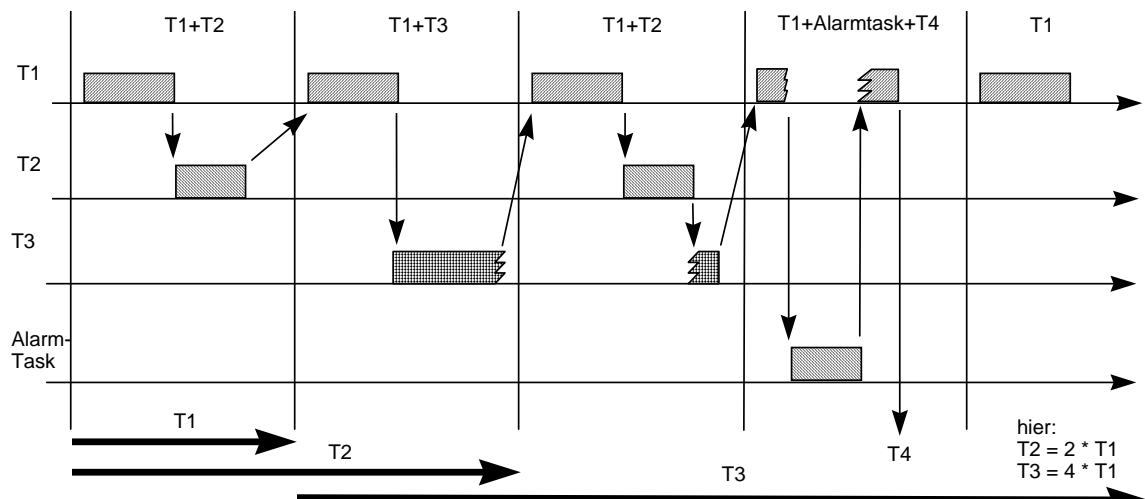


Bild 2-12 Arbeitsablauf einer projektierten Task

#### 2.1.8.4 Beseitigung von Zyklusfehlern

Zur Beseitigung von Zyklusfehlern gibt es durch den modularen Aufbau von SIMATIC TDC/SIMADYN D folgende Möglichkeiten:

- Vergrößern der projektierten Grundabtastzeit

- Verschieben von projektierten Bausteinen von schnellen zu langsamen Tasks
- Verwendung mehrerer bzw. leistungsfähigerer CPUs oder mehrerer SIMATIC TDC/SIMADYN D-Stationen
- Verringerung der Bausteinanzahl oder Veränderung der Bausteintypen
- Überprüfung der Notwendigkeit von Kommunikationsschnittstellen auf dieser CPU
- Überprüfung der Notwendigkeit von Alarmfunktionspaketen auf dieser CPU

**HINWEIS**

Im Einzelfall ist zu überprüfen, welche der Möglichkeiten am wirtschaftlichsten das gewünschte Ergebnis erreicht.

## 2.1.9 Technische Daten des Betriebssystems

### 2.1.9.1 Leistungsmerkmale

Im folgenden werden wichtige Kenndaten und technische Daten des Betriebssystems genannt.

**Anzahl CPU-Baugruppen**

Maximal können bis zu 8 (SIMADYN D) bzw. 20 (SIMATIC TDC) CPU-Baugruppen innerhalb eines Baugruppenträgers gesteckt werden. Eine CPU-Baugruppe benötigt 1 Steckplatz. Die nicht durch CPU-Baugruppen besetzten Steckplätze können durch Peripherie-Baugruppen belegt werden.

**Anzahl der Funktionspläne**

Die maximale Anzahl der Funktionspläne ist projektierungsabhängig, liegt aber ca. bei 65536 ( $2^{16}$ ).

**zyklische Tasks**

Systemplan	automatisch vorhanden
Grundabtastzeit T0 projektierbar	von 0,1 [ms] bis 16 [ms] in Schritten von 0,1 [ms]
Anzahl der projektierbaren zyklischen Tasks	5
von Grundabtastzeit	T0
bis	$T0 * (2^{**} 15)$
projektierbar von	T0 bis $32768 * T0$ z. B. von 1 [ms] bis 32768 [s]

Tabelle 2-10 Technische Daten der zyklischen Tasks



**Alarmtasks**

Anzahl der projektierbaren Alarmtasks	8
Anzahl der verfügbaren Alarmquellen gesamt	54 (SIMADYN D) bzw. 19 (SIMATIC TDC)
davon	
Software-Interrupts	8
CPU-Timer-Interrupts	2
Interrupts für binäre Eingänge	4
Bus-Interrupts(L/C)	2 (SIMADYN D) bzw. 3 (SIMATIC TDC)
LE-Bus-Interrupt	4 (nur SIMADYN D)
LE-Bus-Interrupts erweitert	32 (nur SIMADYN D)
nur T400 ISL, ISR	2

Tabelle 2-11 Technische Daten der Alarmtasks

**Rechenzeiten des Betriebssystems**

Die Durchlaufzeiten des Betriebssystems sind im folgenden auf der Basis der PM5 CPU-Baugruppe angegeben. Für PM6 CPU-Baugruppen verkürzt sich die Rechenzeit in etwa auf ein Drittel der angegebenen Zeitspanne.

Die Signale, die über L- und/oder C-Bus geführt werden, stellen eine nahezu gleichbleibende Systembelastung dar, da der Bus immer mit 8 MHz getaktet wird.

In der folgenden Tabelle wird der minimale Aufwand an Zeit dargestellt, der zur Bearbeitung jedes Zyklus einer Task notwendig ist (Berechnungsgrundlagen siehe oben!):

Zeit für Start	40 µs
Zeit für Beenden	40 µs
zusätzlicher Anteil bei lokalem Puffersystem	20 µs
C-Bus-Puffersystem	20 µs
L-Bus-Puffersystem	20 µs

Tabelle 2-12 Rechenzeiten des Betriebssystems

**Speicherbedarf des Betriebssystems**

Der Code und die Daten des Betriebssystems werden auf der CPU-Baugruppe aus dem Speichermodul ins CPU-RAM kopiert und dabei dekomprimiert. Es entsteht folgender Bedarf an:

- CPU-RAM-Bereich: 400 KByte
- Speichermodul-Bereich: 200 KByte komprimiert

Auf der Koppelspeicher-Baugruppe benutzt das Betriebssystem nach dem Start jeweils 1 Kbyte des C-Bus- und des L-Bus-Koppelspeichers als Bereich zur Verwaltung von Betriebssystemlisten. Dazu kommt noch projektierungsabhängig der entsprechende Bedarf an Speicher für das Puffersystem und weitere Komponenten, z.B. die Kommunikation.

## 2.1.9.2 Die Grundfunktionen des Betriebssystems

### Komponenten des Betriebssystems

Das Betriebssystem setzt sich aus folgenden Komponenten zusammen:

- Aufgabenverwalter für zyklische und alarmgesteuerte Bearbeitung
- Hard- und Software-Initialisierung
- Speicherverwaltung (Pufferverwaltung)
- Betriebssystemdaten und -listen
- Schnittstelle zu den zentralen AMC-Listen
- Kopplung zu den übrigen Komponenten (Systemschnittstellen).

Das Betriebssystem zeichnet sich durch seine Fähigkeit zum Multiprocessing- und Multitaskingbetrieb aus.

Die Grundfunktionen des Betriebssystems sind in das Gesamtsystem eingebettet, womit gleichzeitig die wichtigsten Schnittstellen zur Umgebung dargestellt werden.

Betriebssystemfunktionen	ausgelöst durch
Initialisierung	RESET
zyklische Bearbeitung	Abtastzeit-Timer
alarmgesteuerte Bearbeitung	Prozess-Alarme
Prozessabbild	
Ausnahmebehandlung und Diagnose	Systeminterrupts
Kommunikation, Ein/Ausgabe	Ein/Ausgabe-Interrupts
Service	
Anwenderprogramm	
Dienstprogramme	

Tabelle 2-13 Grundfunktionen des Betriebssystems

### Initialisierung

Die Initialisierung wird durch Einschalten der Stromversorgung oder Betätigung der RESET-Taste und den damit verbundenen Reset-Impuls ausgelöst. Aufgabe der Initialisierung ist es, die Hard- und Software so vorzubereiten, daß das System den Normalbetrieb (Betriebszustand RUN ) aufnehmen kann.

### Zyklische Bearbeitung (Betriebszustand RUN)

Der Aufgabenverwalter sorgt für die zyklische Bearbeitung der den unterschiedlichen Tasks zugeordneten Aufgaben. Die zyklischen Tasks stehen im Verhältnis der 2er-Potenzen zueinander nach

$$T(i) = T(0) * ( 2 ** j )$$

mit T(0) als Grundabtastzeit,  
j bestimmt den Wert der Abtastzeit mit 0 <= j <= 15  
i numeriert die Abtastzeiten mit 1 <= i <= 5.

**Beispiel:**

Bei einer Grundabtastzeit von 1 ms könnten die Abtastzeiten 1 ms, 2 ms, 8 ms, 32 ms und 128 ms sein. Die Grundabtastzeit wird für jede CPU-Baugruppe bei der Projektierung mit Hilfe des Programmteils HWKonfig des SIMATIC Managers festgelegt. Dabei werden auch die Abtastzeiten für die auf der jeweiligen CPU-Baugruppe ablaufenden Tasks projektiert.

Um Häufungspunkte zu vermeiden, werden die Tasks phasenverschoben mit dem Grundtakt gestartet, so daß mit dem Grundtakt jeweils der Start für eine zweite, niederpriore Task vermerkt wird. Infolge der diskreten auf der 2er-Potenz beruhenden Aufteilung der Abtastzeiten werden damit alle niederpriorien Tasks vollständig berücksichtigt, es trifft also niemals mehr als eine niederpriore Abtastzeit auf den Grundtakt (siehe Kapitel "Prozessorauslastung"). Die Prioritäten der Tasks nehmen ab mit zunehmender Abtastzeit.

Mit dem Takt der Grundabtastzeit des Abtastzeit-Timers wird der Aufgabenverwalter gestartet. Dieser ermittelt die zweite, neben der Task T1 zusätzlich zu startenden Task Tn (Tn aus {T2...T5}). Ist die zu startende Task von niedrigerer Priorität als eine unterbrochene Task, so wird ihr Start gepuffert und die unterbrochene Task fortgesetzt. Andernfalls wird die ermittelte Task gestartet. Der Zustand der unterbrochenen Aufgabe wird in einem taskspezifischen Datenbereich festgeschrieben, der die Bearbeitung fortzusetzen erlaubt, sobald keine höherpriore Task mehr ansteht.

In der Darstellung ist der durch das Betriebssystem selbst verbrauchte Zeitanteil nicht berücksichtigt. Bei exakter Darstellung würde sich dann der tatsächliche Startzeitpunkt der Task um diese Anteile verschieben.

**Alarmgesteuerte Bearbeitung**

Neben der zyklischen Bearbeitung verwaltet das Betriebssystem auch Tasks, die von azyklischen Interrupts, insbesondere den Prozessalarmen, gestartet werden. Alarmquellen können sein:

- Software -Interrupts
- CPU-Timer-Interrupts
- L-/C-Bus-Interrupts
- LE-Bus-Interrupts.

Die Prioritäten der Alarmtasks sind durch die Projektierung in HWKonfig festgelegt (I1 > I2...> I8). Der Anwendungsprogrammierer projektiert mit Hilfe von HWKonfig die für seine Anwendung benötigten Alarmquellen und deren Verarbeitung in den alarmgesteuerten Tasks.

**Prozessabbild (Systemmode)**

Vor Beginn der Taskbearbeitung wird untersucht, ob ein zugehöriges Prozessabbild zu aktualisieren ist. Wenn ja, geschieht das vor dem Start der Task durch Aufruf des Systemmodes der Funktionsbausteine (siehe Kapitel "Bedeutung und Einsatzmöglichkeiten des Prozeßabbildes"). Die Aktualisierung bezieht sich auf:

- Binäre Ein-/Ausgaben, wie z.B. die Zustandsabbilder für Reglerfreigaben und die Stellung von Endschaltern.

- Analoge Ein-/Ausgaben, wie z.B. Werte für Temperatur, Drehzahl, u. a.

---

**HINWEIS**

Der Systemmode wird für beide zu startende Tasks vor der Normalmode-Bearbeitung gestartet (siehe Kapitel "Bedeutung und Einsatzmöglichkeiten des Prozeßabbildes").

---

**Fehlerdifferenzierung**

SIMATIC TDC/SIMADYN D differenziert zwischen Fehlern, die während der Initialisierung bzw. während des normalen Ablaufs auftreten.

Fehler aus der Initialisierung (Betriebszustand INIT) führen dazu, daß für das System keine Startfreigabe (Übergang in den Betriebszustand RUN) erfolgt.

Bei Fehlern des Normalbetriebs (Betriebszustand RUN) ist zu unterscheiden, ob sie den weiteren Betrieb fortzuführen erlauben oder aber zwingend einen Abbruch der Bearbeitung erfordern.

Das System informiert den Anwender über seinen Zustand, insbesondere natürlich über die Fehlerzustände mittels einer Siebensegmentanzeige an der CPU-Baugruppe.

Im Fehlerfall sind in Fehlerdatenfeldern des Betriebssystems Detailinformationen hinterlegt, die eine genaue Fehleranalyse erlauben.

Diese Daten können mit Hilfe des Dienstes Service ausgelesen und verändert werden.

**Weitere Informationen**

über die Bedeutung der Fehlersignale sowie Maßnahmen zur Abhilfe siehe Onlinehilfe "D7-SYS, Hilfe zu Ereignissen".

**Kommunikation**

Die Kommunikation realisiert den gesamten Ein-/Ausgabeverkehr zwischen der Hardware sowie den zugehörigen Softwarekomponenten und den Benutzerschnittstellen. Die Schnittstellen und deren Parametrierung sind im Anwenderprogramm mittels CFC zu projektieren.

**Dienst Service**

Der Dienst Service ist die zentrale Schnittstelle der CPU-Baugruppen. Er ist ein Instrument für die Inbetriebnahme, Diagnose und Fehlersuche.

Da die Bearbeitungszeit des Dienstes Service unbestimmt ist, könnte die an ihn gekettete Task sowie die Tasks mit niedrigerer Priorität blockiert werden. Dies wurde so realisiert, daß dem Service eine maximale Bearbeitungszeit innerhalb seines Zyklus zugewiesen wird (maximal ein Grundtakt T0).

Die Benutzerschnittstelle bilden Servicegeräte, die über die Kommunikationssoftware angesteuert werden.

**Anwenderprogramm** Das Anwenderprogramm realisiert die technologischen Aufgaben auf der Zielhardware. Es wird am Programmiergerät in der Programmiersprache CFC unter Zuhilfenahme der verfügbaren Dienstprogramme, wie HWKonfig, CFC-Editor, CFC-Compiler, Linker/Locator und Speichermodul-Treiber erstellt.

Der CFC-Quellcode des Anwenderprogramms wird durch den CFC-Compiler in Datenstrukturen umgesetzt und zur Abarbeitung durch das Betriebssystem auf der Zielhardware bereitgestellt.

**Dienstprogramme** Unter diesem Begriff werden für das Betriebssystem Systemgrundfunktionen zusammengefaßt. Das sind Weckerfunktionen, Funktionen zur Behandlung der Anzeige an der CPU, spezielle Test- und Interruptroutinen für die Behandlung von Systemfehlern.

### 2.1.9.3 Der Dienst Service

Der Dienst Service stellt einen Pool von Auskunftsfunktionen bereit, damit Anwender zu prozessorlokalen Systeminformationen Zugang haben. Der Dienst Service ist als Hilfsmittel für die Einsatzgebiete IBS (Inbetriebsetzung) und Test konzipiert.

**Einsatzgebiet IBS** Hier werden projektierte Daten (Soll-/Ist-Werte) angezeigt und/oder verändert sowie die Projektierung optimiert (z.B. Verschaltungsänderungen, Reglerzeiten verändern etc.).

**Einsatzgebiet Test** Hier lassen sich Ursachen für Anlagenstörungen (Absturz, Hochlaufprobleme) und Störungen, die ihre Ursache innerhalb der eigenen CPU-Baugruppe haben, ermitteln.

Alle Aktivitäten des Dienstes Service werden über Aufträge gesteuert, die über "seine" Datenschnittstelle (entsprechend der Parametrierung der Service-Funktionsbausteinanschlüsse) eintreffen.

Als Bediengerät für den Dienst Service kommen alle Geräte in Frage, die die Auftrags- und Antwortsprache der Dienste verarbeiten können. Im SIMATIC TDC/SIMADYN D Spektrum sind dies die Programme (Tools) CFC im Testmodus und Service IBS.

---

**HINWEIS** Es besteht für Anwender auch die Möglichkeit, eigene Tools einzusetzen. Sie müssen dann die Schnittstellen-Definitionen des Dienstes Service beherrschen. Die Schnittstellenspezifikation ist bei A&D DL R beziehbar.

---

Der Dienst Service wird mit dem Funktionsbaustein "SER" zur Verfügung gestellt. Dieser Funktionsbaustein stellt sicher, daß keine Nachrichten verlorengehen.

**Auftragsbearbeitung** Der Dienst Service unterscheidet zyklische und nichtzyklische Aufträge. Ein nichtzyklischer Auftrag ist abgeschlossen, wenn sein Antworttelegramm abgeschickt ist.

Ein zyklischer Auftrag bleibt solange aktiv, bis er explizit beendet wurde, entweder durch Abbrechen über Reset oder durch einen neuen Auftrag. Zu einem Auftrag gehört immer mindestens ein Antworttelegramm.

**HINWEIS**

Der Dienst Service kann immer nur einen Auftrag bearbeiten. Der nächste Auftrag wird erst bearbeitet, nachdem der vorangegangene beantwortet wurde.

---

**Systembelastung,  
Antwortzeiten**

Die eigentliche Verarbeitung des Dienstes Service passiert in einer Abtastzeit um 32 ms (es wird die nächste Abtastzeit unter 35 ms gewählt; die an den SER-Bausteinen angegebenen Abtastzeiten sind also für die Verarbeitung nicht maßgeblich). In der verwendeten zyklischen Task wird den Servicebausteinen eine gewisse Rechenzeit zur Verfügung gestellt, die maximal den Grundtakt T0 in Anspruch nehmen darf. Das Verhältnis von Grundtakt T0 zur verwendeten Task bestimmt die zur Verfügung stehende CPU-Leistung und damit die Systembelastung.

Beispiel 1:

Grundtakt T0 = 1 ms; gewählte Abtastzeit = 32 ms. Es werden alle 32 ms jeweils 1 ms für Dienst Service reserviert. Somit errechnet sich die Systembelastung aus

$$1 \text{ ms} / 32 \text{ ms} = 0.03125 = 3.125\%$$

Beispiel 2:

Grundtakt T0 = 2 ms; gewählte Abtastzeit = 16 ms. Es werden alle 16 ms jeweils 2 ms für Dienst Service reserviert. Somit errechnet sich die Systembelastung aus

$$2 \text{ ms} / 16 \text{ ms} = 0.125 = 12.5\%$$

Die zur Verfügung stehende Rechenzeit wird von allen Service-Bausteinen gleichberechtigt genutzt, d.h. solange die Zeit reicht, werden möglichst alle SER-Bausteine einmal durchlaufen. Je Takt bearbeitet ein SER-Baustein maximal einen Auftrag. Bei zyklischen Aufträgen kommt je Takt höchstens ein Antworttelegramm. Vorteil dieser Arbeitsweise ist, daß bei zyklischen Aufträgen zeitlich äquidistante Antworten zustande kommen können.

Wird die reservierte Rechenzeit nicht vollständig genutzt, weil z.B. kein Auftrag zur Bearbeitung ansteht, wird sie dem System zur Verfügung gestellt.

Bei Mehrfachprojektierung und gleichzeitigem Zugriff auf nur einmal vorhandene Systemressourcen (z.B. Änderungsspeicher des Speichermoduls), bekommt derjenige die Ressource zugeteilt, der als erster die Anforderung stellt. Alle anderen werden abgewiesen und bringen nach spätestens 1 Sekunde eine Fehlermeldung ("Resource belegt") über die Datenschnittstelle.

**Verhalten im  
Störfall**

Im Störfall (Exceptionfall), also bei Initialisierungsfehlern oder Onlinestörungen, geht das System in den Stopbetrieb. Damit gelten besondere Bedingungen für den Dienst Service. Er wird dann nicht mehr in einer zyklischen Task gerechnet, sondern läuft, gestartet von einem Exceptionhandler, ständig. Im Störfall kann der Dienst Service nicht mit dem projektierten User verbunden werden. Um trotzdem Systemdiagnose zu ermöglichen, wird dann die CPU-eigene Diagnoseschnittstelle angeschlossen. Hier läuft das DUST1-Protokoll (siehe Kapitel "Betriebszustände einer CPU-Baugruppe").

## 2.2 Funktionsbeschreibung und Benutzerhinweise

### 2.2.1 Fatale Systemfehler "H"

Tritt ein fataler Systemfehler auf, so wird die Verarbeitung (Initialisierung oder Normalbetrieb) abgebrochen und in den Stopbetrieb übergegangen. Es steht dann Service zur Diagnose der Fehlerursache zur Verfügung.

#### HINWEIS

Bevor einem fatalen Systemfehler auf den Grund gegangen wird, sollten erst die System-Fehlerfelder INIT\_ERR und SYS\_ERR untersucht werden. Sind dort Fehler (insbesondere Hardware-(Überwachungs-Fehler) eingetragen, können sie die Ursache eines fatalen Systemfehlers sein.

Im lokalen RAM einer jeden CPU-Baugruppe wird im oberen Bereich ein SAVE-Bereich angelegt. Dieser Bereich wird bei einer erneuten Initialisierung nicht gelöscht, wenn der Status der RAM-Kopie entsprechend ist. In diesem SAVE-Bereich wird ein Fehlerpuffer angelegt, der das Fehlerprotokoll, bestehend aus mehreren Meldungen, enthält.

Der Fehlerpuffer besteht aus einem Verwaltungsteil und einem Ringpuffer, in dem die Fehlermeldungen gespeichert werden. Der Ringpuffer ist als überschreibender Puffer realisiert, d.h. wenn der Puffer mit Fehlermeldungen vollgeschrieben ist, werden durch neue Meldungen die ältesten Meldungen überschrieben.

Es gibt 2 verschiedene Typen von Fehlermeldungen. Im Falle eines Non maskable Interrupt NMI wird eine lange Meldung abgesetzt. Im Falle eines Power-OFF eine kurze Meldung.

Zur Diagnose der Fatalen-Systemfehler steht der Kommunikationsdienst Service zur Verfügung (auch wenn nicht projektiert). Er ist nach Drücken der Quittierungstaste über die lokale Diagnose-Schnittstelle erreichbar. Mit Hilfe von Dienst Service werden die Fehlerursachen im Klartext ausgegeben.

Wichtig ist vor allem die unter Kennung und Zusatzkennung angegebene Fehlerursache. Ist im Moment des Systemfehlers gerade ein Funktionsbaustein in Bearbeitung, wird dieser ausgegeben. Zusätzlich werden die Ergebnisse der letzten Buszugriffe angezeigt, diese sind dann wichtig, wenn ein Buszugriff die Fehlerursache ist. Für den Systemspezialisten werden zur genauen Fehleranalyse außerdem noch alle Prozessorregister angezeigt.

#### NMI - Behandlung

Das Auftreten eines Non maskable Interrupts gilt als fataler Fehler und führt zum Abbruch der Initialisierung oder des Normalbetriebs. Die Verarbeitung auf allen im Baugruppenträger steckenden Baugruppen wird abgebrochen.

An der CPU-Baugruppen-Anzeige der gestörten Baugruppe, die den fatalen Fehler verursacht hat, wird ein blinkendes großes *H* angezeigt. Auf den anderen CPU-Baugruppen, die durch die gestörte Baugruppe



einen NMI erhalten, wird ein großes **H** konstant angezeigt. Der Debug Monitor kann durch Betätigen der Quittierungstaste oder durch Setzen des Statuswertes aktiviert werden. Die auf der 7-Segmentanzeige ausgegebenen Symbole bedeuten folgendes:

konstantes **H**: CPU-Baugruppe wurde durch andere Baugruppe abgeschaltet.

blinkendes **H**: Fataler Fehler auf dieser CPU-Baugruppe (Verursacher).

### Beispiel für ein Fehlerprotokoll bei fatalen Systemfehlern

```
Informationen zum letzten Absturz:
Uhrzeit:          01.01.93  04:16:24.9294 h
Kennung:          #5 CPU
Zusatzkennung:   28 (unaligned instruction fetch)
EPC:              0x04C4F19A
Ruecksprungadresse: 0x801201f8
```

```
Laufende Aufgabe: T2
Gestartete Ebenen: T2(NRM),T5(NRM),BACKGROUND
Zuletzt bearbeit. FB: FP-KRUMMS.AY0815 (Typ: ADD8F)
                    (ALE: 0x80107D84  CODE: 0x801201E0)
```

```
Letzter L-Bus-Zugriff:
- Zugriffsart:      q_read_2byte
- BUS-Adresse:     0xB0F25874
- Wiederholungen:  0
Letzter C-Bus-Zugriff:
- Zugriffsart:      q_read_2byte
- BUS-Adresse:     0xB4F400B4
- Wiederholungen:  0
```

```
Es folgt der Prozessorzustand zum Absturzzeitpunkt:
EPC      : 0x04C4F19A  BadVAddr: 0x04C4F19A
Status   : 0xF000FC14  mdlo    : 0x04C4F19A
fpc_csr  : 0x00000F04
```

ACHTUNG: Der Wert von a0, a1 (und evtl. a2) ist ungueltig!

```
r00/0 : 0x00000000  r01/at: 0x00000000  r02/v0: 0x00000001  r03/v1: 0xB8803000
r04/a0: 0x80064FC8  r05/a1: 0x80064F44  r06/a2: 0x0000000A  r07/a3: 0x00000000
r08/t0: 0x80064FC4  r09/t1: 0x80065048  r10/t2: 0x19999999  r11/t3: 0x00000000
r12/t4: 0x04C4F19A  r13/t5: 0x8007FE90  r14/t6: 0x00000000  r15/t7: 0x00000000
r16/s0: 0x800650CC  r17/s1: 0x8006511C  r18/s2: 0x00000000  r19/s3: 0x8006548C
r20/s4: 0x8006548C  r21/s5: 0x00000020  r22/s6: 0x800812E0  r23/s7: 0x80400000
r24/t8: 0xFFFFFFFF  r25/t9: 0x8007FE90  r26/k0: 0x00000210  r27/k1: 0x04C4F19A
r28/gp: 0x80088BA0  r29/sp: 0x80064EA8  r30/s8: 0x04C4F19A  r31/ra: 0x801201f8
```

```
d00:      +Denorm  d02: +6.400000e+01  d04:      +Denorm  d06: +9.999000e-01
d08: +4.687500e+01  d10:      +Denorm  d12:      +Denorm  d14: +2.660285e+154
d16: +3.000001e+03  d18: +4.687500e+01  d20:      QNaN     d22: +8.329648e+298
d24: -1.818767e-12  d26: -1.227518e+306  d28: -3.691391e+249  d30: -2.374690e-237
```

```
f00: +4.787490e+01  f02: +0.000000e+00  f04: +4.687500e+01  f06: +4.764729e+05
f08: +0.000000e+00  f10:      +Denorm  f12: +4.787490e+01  f14: -1.960343e+37
f16: +1.024000e+03  f18: +0.000000e+00  f20: -1.693935e+38  f22: QNaN
f24:      QNaN     f26:      QNaN     f28: -6.835168e-27  f30: -4.550802e-04
```

----- Ende der Diagnose -----

#### Absturzzeitpunkt

Absturzursache  
Absturzadresse und  
Adresse des letzten Funktionsaufrufs

Zum Absturzzeitpunkt laufende und gestartete Tasks sowie der bearbeitete Funktionsbaustein mit Angabe der Daten und Codebereiche

Angaben zu Systembuszugriffen. Interessant sind diese Angaben vor allem, wenn ein fehlerhafter Buszugriff als Absturzursache bei Kennung (NMI) und Zusatzkennung angegeben wird.

#### Registerdump aller Prozessorregister

Wichtig vor allem:

EPC (Absturzadresse, wie oben) und BadVAddr (Bad Virtual Address, Adresse, auf die fehlerhaft zugegriffen wurde (hauptsächlich bei Kennung TLB und CPU))

**Ursachen fataler Fehler**

Ein fataler Systemfehler kann eine folgende Ursachen (Kennungen) haben. Die Zusatzkennung beschreibt die Fehlerursache genauer.

<b>Kennung</b>	<b>Zusatzkennung (genaue Beschreibung)</b>
NMI	ein nicht maskierbarer Interrupt zweites Bus Clear bei auftragsgesteuertem Zugriff Bus Clear bei Direkt-Zugriff Time-Out während L/C-Bus-Arbitrierung/Belegung (Baugruppe fehlt/defekt, Daisy Chain fehlt) Ready Intern vom L/C-Bus (Fehler auf anderer CPU-Baugruppe) Ready Intern vom lokalen Erweiterungsbus (LE-Bus) Overrun des Systembus-Controllers Time-Out bei Zugriff auf lokale Peripherie Spurious Interrupt (Interruptquelle läßt sich nicht ermitteln) Direktzugriff auf L/C-Bus (unter Umgehung der Treiberfunktionen)
CPU	Ausnahmezustand der CPU interner Fehler reserved Instruction unknown Syscall unaligned instruction fetch (Sprung auf nicht durch vier teilbare Adresse) user access to kernel space unaligned load/store to coprozessor 0/2/3 unaligned load/store to L-/C-Bus address space break 6/7 not in div/mul context unknown break value reserved exception Task in Endlosschleife gelaufen
FPU	Ausnahmezustand der FPU fpu fault at non-fpu instruction illegal fpu sub opcode operation on NaNs add/sub/division of infinities mul of infinity and 0
TLB	Ausnahmezustand des TLB TLB modified exception TLB read/write miss (Zugriff auf nicht erlaubte Adresse) UTLB miss (Zugriff auf nicht erlaubte Adresse)
TIME	Ausfall der Grundabtastzeit
OFF	Power down Power down/Reset erfolgte im Normalbetrieb Power down/Reset erfolgte im Stopbetrieb (nach anderer Exception)

## 2.2.2 Hintergrund-Verarbeitung

Stehen während des Normalbetriebs keine Aufgaben zur Bearbeitung durch die CPU an, so bearbeitet sie die Hintergrund-Task.

Als Hintergrundtask stehen gleichzeitig folgende Funktionalitäten zur Verfügung:

- der On-Line-Testmodus und
- ein Dienst Service.

Der On-Line-Testmodus wird normalerweise nach fehlerfreier Beendigung der Initialisierung im Hintergrund bearbeitet. Ist jedoch am Ende der Initialisierung die Quittierungstaste gedrückt, so wird ausschließlich der Kommunikationsdienst Service aktiviert.

Fehler der Hintergrund-Verarbeitung werden in dem Element UEB des Fehlerfeldes SYS\_ERR abgelegt.

### 2.2.2.1 On-Line-Testmodus

Im On-Line-Testmodus wird z.B. ein Batterie-Test, ein Speichermodul-Checksum-Test usw. durchgeführt. Die Speichermodul-Checksum-Routine ermittelt die Speichermodul-Check-Summe und vergleicht sie mit der vom Programmiergerät errechneten und im Speichermodul hinterlegten. Wird im On-Line-Testmodus ein Speichermodul-Check-Sum-Fehler festgestellt, so kann der Anwender durch ein wiederholtes Erzeugen des Speichermoduls den Fehler beseitigen. Beim Batterie-Test-Fehler kann er die Batterie wechseln.

## 2.3 Systemplan @SIMD

**Übersicht** Der Systemplan @SIMD ist ein dem Anwender standardmäßig zur Verfügung gestellter CFC-Plan, der eine Standard-Diagnose der Hardware und der Systemsoftware ermöglicht. Er besteht aus den Teilplänen A und B.

**Programmstruktur** Die Projektierung des Systemplans gliedert sich in die Teile

- Quittung erkennen      Quittierung der Fehleranzeige
- Komponenten auswerten      Ermittlung der Komponente, die einen Fehler meldet
- Anzeige      Ausgabe des erkannten Fehlers

Systemplan @SIMD	
	Funktionsbausteinamen
Quittung erkennen	
Taster	ACK Acknowledge
Servicebedienung	ACK
Komponenten auswerten	
Erstfehlerfeld	FER First error
Kommunikationsfehlerfeld	CER Communication error
Aufgabenverwalterfehlerfeld	TER Task management error
HW-Ausfall Überwachungsfehlerfeld	HER HW error
Anwenderfehlerfeld	UER User error
Fehler auswerten	DER Display error
Anzeige	
Ausgabe Siebensegmentanzeige	DST Display status
Ausgabe Diagnose LED	DST
Ausgabe Statuswort SIMS, Statusbit SIMD	SIMS, SIMD

Tabelle 2-14 Detailangaben über den Systemplan @SIMD

**Beschreibung** Das Betriebssystem überwacht die Hardware und die Systemsoftware. Erkennt die Überwachung einen Fehler, gibt sie ihn dadurch bekannt, daß sie im Systemfehlerfeld entsprechende Bits (Flags) setzt.

Der Systemplane @SIMD macht diese Flags dem Anwender zugänglich. Dazu erscheint eine Ausgabe auf der Siebensegmentanzeige der CPU-Baugruppe, wenn ein Flag einer Komponente gesetzt wurde.

Werden mehrere Meldungen für die Siebensegmentanzeige erzeugt, wird die höchstpriorie Meldung ausgegeben.

Fehlerbezeichnung	Fehleranzeige	Priorität
Kommunikationsfehler	C	hoch
Aufgabenverwalterfehler	E	
HW-Ausfall Überwachungsfehler	b	
Anwendererzeugtes Fehlerkennzeichen	A	
Kein Fehler steht an	CPU-Nummer	niedrig

Tabelle 2-15 Fehlerprioritäten für die Meldungsanzeige

### Rücksetzen des Flags

Drücken der Quittierungstaste an der CPU-Baugruppe bzw. Quittierung über ein Servicegerät bewirkt das Rücksetzen der Flags des angezeigten Fehlers und die Anzeige des nächstpriorien Fehlercodes. Als niedrigstpriorie Meldung auf der Siebensegmentanzeige erscheint im fehlerfreien Fall die CPU-Nummer. Um kenntlich zu machen, daß die angezeigte Fehlermeldung als erste aufgetreten ist, erscheint ihre Anzeige blinkend.

### Funktionsweise

Das Ablaufdiagramm (siehe Bild 3-1) stellt den globalen Programmablauf des Systemplans dar. Er besteht aus den drei funktionalen Komponenten

- Quittung erkennen
- Komponenten auswerten und
- Anzeige.

### Quittung erkennen

Das Quittierungssignal ist ein Impuls, der von dem Funktionsbaustein ASI eingelesenen Tasterzustand oder durch eine Servicebedienung am Anschluss ACK000.I (setzen von 1 auf 0) abgeleitet ist. Mit diesem Impuls werden prioritätsgesteuert Fehlerfelder und damit ihre Anzeige quittiert. Durch Ändern des Anschlusses ACK050.I von 0 auf 1 kann die Ausgabe der Fehlercodes "C" und "E" unterdrückt werden.

### Komponenten auswerten

Die Komponenten werden mit dem Funktionsbaustein SYF1 bzw. SYF4 ausgewertet. Die entsprechenden Nummern der Fehlerfelder sind in der Funktionsbausteinbeschreibung (siehe Referenzhandbuch SIMADYN D Funktionsbaustein-Bibliothek) dokumentiert. Das Quittieren eines Fehlerfeldes ist nur möglich, wenn für die jeweilige Komponente ein Fehler erkannt und dieser dargestellt wurde.

### Auswertung des Erstfehlerfeld

Die Erstfehlerfeldauswertung stellt fest, welcher Fehlereintrag als erster vom System erkannt wurde. Der Fehler im Erstfehlerfeld wird als blinkend auf der Siebensegmentanzeige dargestellt.

Nacheinander werden alle Komponenten nach ihrer Priorität ausgewertet. Das Kommunikationsfehlerfeld kann nicht quittiert werden, da zur Beseitigung dieses Fehlers eine Projektierungsänderung erforderlich ist. Im Systemanlauf kann es zu erhöhten Auslastungen der CPU kommen.

Mit einer Zähllogik werden während des Systemanlaufs Aufgabenverwaltungsfehler automatisch quittiert.

**Steuerung durch Prioritätslogik**

Eine Prioritätslogik stellt sicher, daß jeweils nur die höchstpriorie Komponente zur Anzeige kommt. Die niedrigstpriorie Komponente liefert ein Bitsignal, welches die Anzeige von der CPU-Nummernanzeige auf Fehleranzeige umschaltet (UER070.Q). Ist die höchstpriorie Fehlerkomponente zusätzlich im Erstfehlerfeld eingetragen, so wird die Fehleranzeige mit einem Blinktakt ausgegeben.

Ein Quittierimpuls setzt immer nur einen Fehlerstatus einer Komponente und deren Anzeige zurück.

**HINWEIS**

Wird ein angezeigter Fehler quittiert, so ist die Fehlerquelle immer noch vorhanden. Bevor ein Fehler bestätigt wird, ist in jedem Fall die Fehlerursache zu ermitteln und der Fehler zu beseitigen.

**Anzeige**

Die Ansteuerung der Siebensegmentanzeige erfolgt im fehlerfreien Fall durch Ausgabe der eigenen Prozessornummer. Meldet eine Komponente einen Fehler, so wird der entsprechende Fehlercode ausgegeben.

Die Statusanzeige auf einer T400 erfolgt über eine Diagnose-LED. Liegt ein Erstfehler vor, so wird der Blinktakt erhöht.

Die Statusanzeige auf einer FM 458 erfolgt über acht LED-Anzeigen an ihrer Frontseite (siehe Benutzerhandbuch "Applikationsbaugruppe FM 458").

**Ablaufdiagramm**

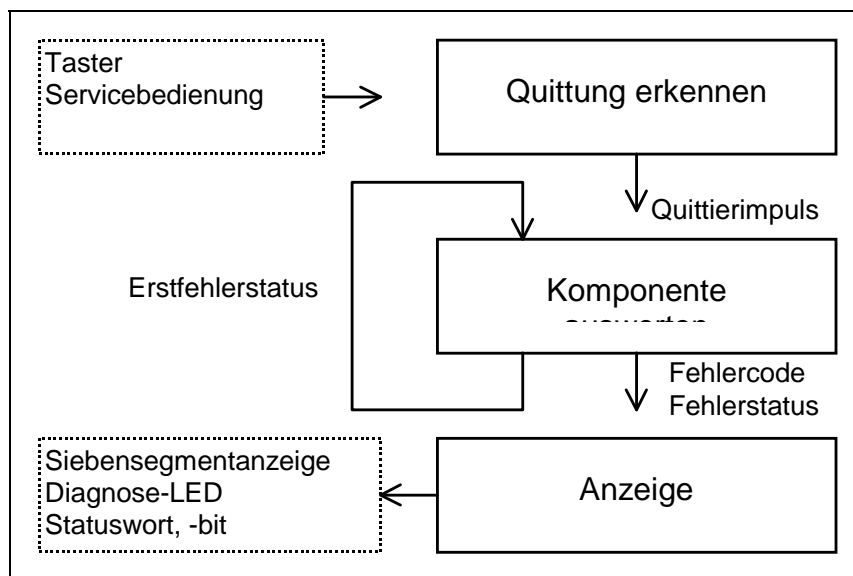


Bild 2-13 Ablaufdiagramm

**Schnittstellen**

Als externer Eingang des Systemplanes ist die Quittierungstaste des CPU-Moduls bzw. eine Quittierungsmöglichkeit über die Serviceschnittstelle vorgesehen. Als externe Ausgänge für die Anwenderanzeige stehen die Siebensegmentanzeige des CPU-Moduls oder die Diagnose-LED (T400) bzw. Zustandsanzeigen (FM 458) zur Verfügung.

Für eine Fehlerverarbeitung im Anwenderprogramm können die beiden Anschlüsse SIMS.QS und SIMD.Q ausgewertet werden. Die Fehlerausgänge der einzelnen Komponenten sind über den Funktionsbaustein SIMS zu einem Fehlerstatuswort zusammengefaßt. Der Ausgangsanschluss SIMD.Q stellt einen generellen Fehlerstatus dar.

Das Fehlerstatuswort am Bausteinanschluß SIMS.QS hat folgende Bitbelegung:

Bit	Bitbelegung
Bit1	nicht benutzt
Bit2	nicht benutzt
Bit3	nicht benutzt
Bit4	Aufgabenverwalterfehler
Bit5	nicht benutzt
Bit6	HW-Ausfall
Bit7	Kommunikationsfehler
Bit8	nicht benutzt
Bit9	nicht benutzt
Bit10	nicht benutzt
Bit11	Anwendererzeugtes Fehlerkennzeichen
Bit12	nicht benutzt
Bit13	nicht benutzt
Bit14	nicht benutzt
Bit15	nicht benutzt
Bit16	nicht benutzt

Tabelle 2-16 Bitbelegung des Funktionsbausteinanschlusses SIMS.QS





# 3 Kommunikationsprojektierung

<b>Kapitelübersicht</b>	3.1	Einführung	3-2
	3.2	Baugruppenträger-lokale Kopplungen	3-21
	3.3	Baugruppenträger-Kopplung CP52M0	3-24
	3.4	Baugruppenträger-Kopplung CP53M0	3-30
	3.5	Kopplung TCP/IP (CP51M1)	3-38
	3.6	Kopplung TCP/IP (CP5100)	3-47
	3.7	Kopplung PROFIBUS DP (CP50M1)	3-55
	3.8	Kopplung PROFIBUS DP (CP50M0)	3-68
	3.9	Kopplung MPI	3-105
	3.10	Tabellenfunktion	3-106
	3.11	Kommunikations-Dienst Meldesystem	3-144
	3.12	Kommunikations-Dienst Prozessdaten	3-159
	3.13	Kommunikations-Dienst Service	3-178
	3.14	Kommunikations-Dienst Uhrzeitsynchronisation	3-181
	3.15	Kommunikation mit SIMATIC Operator Panels	3-183
	3.16	WinCC- Anbindung an SIMATIC TDC über Standard-Kanal (SIMATIC S7 Protocol Suite.CHN)	3-191
	3.17	Kommunikation mit WinCC (TCP/IP)	3-225
	3.18	Kommunikations-Dienst Trace	3-235

## 3.1 Einführung

### 3.1.1 Grundlagen der Kommunikation

<b>Allgemeines</b>	Die Kommunikation ermöglicht den Informationsaustausch zu anderen Systemen und Geräten.  Voraussetzungen für die Kommunikation sind: <ul style="list-style-type: none"> <li>• die Projektierung eines Kommunikations-Dienstes und einer Kopplung</li> <li>• das Vorhandensein einer Kommunikationsschnittstelle</li> </ul>
<b>Kommunikations-Dienst</b>	Der Kommunikations-Dienst bestimmt den Informationsinhalt (z.B. Prozessdaten) bei der Kommunikation.
<b>Kopplung</b>	Die Kopplung legt die Hardware (z.B. CP50M0) und das Übertragungsprotokoll (z.B. PROFIBUS DP) für die Kommunikation fest.
<b>Kopplungen und Kommunikations-schnittstellen</b>	Der Anwendungsfall und die Kommunikationsmöglichkeiten des Partners bestimmen die Kommunikationsschnittstelle und die Kopplung.

#### 3.1.1.1 Übersicht Kopplungen

**Allgemeines** Kopplungen werden in der Applikation CFC durch die Kopplungs-Zentralbausteine projektiert.

#### **CPU-lokale Kopplung**

Anwendung bei Kommunikations-partner	<ul style="list-style-type: none"> <li>• CPU-intern zum Testen von Sendern/Empfängern</li> </ul>
notwendige Hardware	<ul style="list-style-type: none"> <li>• CPU</li> </ul>
Kommunikations-Dienste	<ul style="list-style-type: none"> <li>• Prozessdaten</li> </ul>
Kopplungs-Zentralbaustein	<ul style="list-style-type: none"> <li>• @LOCAL</li> </ul>
Merkmale	<ul style="list-style-type: none"> <li>• SIMATIC TDC interne Speicherkopplung</li> </ul>

Tabelle 3-1 CPU-lokale Kopplung

**Direkte CPU-CPU  
Kopplung**

Anwendung bei Kommunikations-partner	<ul style="list-style-type: none"> <li>• CPU-CPU-Kommunikation bei größeren Datenmengen als Alternative zu \$-Signalen</li> </ul>
notwendige Hardware	<ul style="list-style-type: none"> <li>• CPU</li> </ul>
Kommunikations-Dienste	<ul style="list-style-type: none"> <li>• Prozessdaten</li> </ul>
Kopplungs-Zentralbaustein	<ul style="list-style-type: none"> <li>• @LOCAL</li> </ul>
Merkmale	<ul style="list-style-type: none"> <li>• SIMATIC TDC interne Speicherkopplung</li> </ul>

Tabelle 3-2 CPU-CPU Kopplung

**Koppelspeicher-  
Kopplung**

Anwendung bei Kommunikations-partner	<ul style="list-style-type: none"> <li>• CPU-CPU-Kommunikation bei größeren Datenmengen als Alternative zu \$-Signalen</li> </ul>
notwendige Hardware	<ul style="list-style-type: none"> <li>• Koppelspeicher (CP50M0 / CP50M1 / CP51M1 oder CP53M0)</li> </ul>
Kommunikations-Dienste	<ul style="list-style-type: none"> <li>• Prozessdaten</li> </ul>
Kopplungs-Zentralbaustein	<ul style="list-style-type: none"> <li>• @GLOB</li> </ul>
Merkmale	<ul style="list-style-type: none"> <li>• SIMATIC TDC interne Speicherkopplung</li> </ul>

Tabelle 3-3 Koppelspeicher-Kopplung

**SIMATIC TDC-  
Baugruppenträger-  
Kopplung**

Anwendung bei Kommunikationspartner	<ul style="list-style-type: none"> <li>• SIMATIC TDC</li> </ul>
notwendige Hardware	Kommunikationsbaugruppen für Masteranschaltung: <ul style="list-style-type: none"> <li>• CP52M0 (GDM-Memory)</li> <li>• CP52I0 (GDM-Memory )</li> </ul>
notwendige Hardware	Kommunikationsbaugruppe für Slaveanschaltung: <ul style="list-style-type: none"> <li>• CP52A0</li> </ul>
Kommunikations-Dienste	<ul style="list-style-type: none"> <li>• Prozessdaten, Meldesystem, Trace</li> </ul>
Kopplungs-Zentralbaustein	<ul style="list-style-type: none"> <li>• @SRACK</li> </ul>
Merkmale	<ul style="list-style-type: none"> <li>• Lichtwellenleiter</li> <li>• parallele Kopplung von bis zu 44 SIMATIC TDC Baugruppenträgern</li> <li>• Synchronisation aller Baugruppenträger möglich</li> <li>• einheitliche Systemuhrzeit möglich</li> <li>• schnell</li> <li>• maximale Entfernung zwischen 2 Baugruppenträgern beträgt 200 m</li> <li>• Abschalten der Baugruppenträger jederzeit erlaubt</li> </ul>

Tabelle 3-4 SIMATIC TDC-Baugruppenträger-Kopplung

**Baugruppenträger-  
Kopplung TDC mit  
SIMADYN D**

Anwendung bei Kommunikationspartner	SIMATIC TDC mit SIMADYN D
notwendige Hardware	Kommunikationsbaugruppe für SIMATIC TDC: CP53M0
notwendige Hardware	Kommunikationsbaugruppe für SIMADYN D: CS12/CS13/CS14/CS22
Kommunikations-Dienste	Prozessdaten
Kopplungs-Zentralbaustein	@CS1 (Mastermodus) @CS2 (Slavemodus)
Merkmale	Lichtwellenleiter parallele Kopplung von SIMATIC TDC an SIMADYN D parallele Kopplung von bis zu 3 SIMATIC TDC Baugruppenträgern Synchronisation aller Baugruppenträger möglich einheitliche Systemuhrzeit möglich schnell maximale Entfernung zwischen 2 Baugruppenträgern beträgt 200 m Abschalten der Baugruppenträger jederzeit erlaubt

Tabelle 3-5 Baugruppenträger-Kopplung SIMATIC TDC mit SIMADYN D

**Kopplung TCP/IP**

Anwendung bei Kommunikationspartner	<ul style="list-style-type: none"> <li>• SIMATIC TDC</li> <li>• SIMATIC S5/S7</li> <li>• Fremdsysteme</li> <li>• WinCC</li> </ul>
notwendige Hardware	<ul style="list-style-type: none"> <li>• Kommunikationsbaugruppe CP5100 / CP51M1</li> </ul>
Kommunikations-Dienste	<ul style="list-style-type: none"> <li>• Prozessdaten und Meldesystem</li> <li>• Service</li> <li>• S7-Kommunikation</li> </ul>
Kopplungs-Zentralbaustein	<ul style="list-style-type: none"> <li>• @TCPIP</li> </ul>
Merkmale	<ul style="list-style-type: none"> <li>• standardisierter Bus nach Ethernet (IEEE 802.3)</li> <li>• Baudrate: 10 bzw. 100 Mbaud (Autosensing)</li> </ul>

Tabelle 3-6 Kopplung TCP/IP

**PROFIBUS DP**

Anwendung bei Kommunikationspartner	<ul style="list-style-type: none"> <li>• SIMATIC S5/S7</li> <li>• Stromrichtergeräte SIMOVERT/SIMOREG</li> <li>• ET200</li> <li>• SIMATIC TDC</li> <li>• zertifizierte Fremdgeräte</li> </ul>
notwendige Hardware	<ul style="list-style-type: none"> <li>• Kommunikationsbaugruppe CP50M1 / CP50M0</li> </ul>
Kommunikations-Dienste	<ul style="list-style-type: none"> <li>• Prozessdaten</li> <li>• Parameterbearbeitung</li> </ul>
Kopplungs-Zentralbaustein	<ul style="list-style-type: none"> <li>• @PRODP</li> </ul>
Merkmale	<ul style="list-style-type: none"> <li>• standardisierter Multi Master Bus für die Kommunikation von SIMATIC TDC mit maximal 123 Kommunikationspartnern</li> <li>• Master Slave Prinzip (CP50M1 / CP50M0 ist Master und/oder Slave)</li> <li>• PROFIBUS-Norm nach EN 50170</li> <li>• schnell</li> <li>• max. 12 Mbaud</li> <li>• maximale Nutzdatenlänge 244 Bytes</li> <li>• Bus-Parametrierung mit COM-PROFIBUS-Software (nur CP50M0)</li> </ul>

Tabelle 3-7 Kopplung PROFIBUS DP

**MPI**

Anwendung bei Kommunikationspartner	<ul style="list-style-type: none"><li>• CFC</li><li>• WinCC</li><li>• SIMATIC-OPs</li></ul>
notwendige Hardware	<ul style="list-style-type: none"><li>• Kommunikationsbaugruppe CP50M1 / CP50M0</li></ul>
Kommunikations-Dienste	<ul style="list-style-type: none"><li>• Service</li><li>• S7-Kommunikation</li></ul>
Kopplungs-Zentralbaustein	<ul style="list-style-type: none"><li>• @MPI</li></ul>
Merkmale	<ul style="list-style-type: none"><li>• Multi-Master-Bus mit maximal 126 Teilnehmern</li><li>• 187,5 kBaud / 1,5 MBaud</li><li>• Standard bei SIMATIC S7</li></ul>

Tabelle 3-8 Kopplung MPI

### 3.1.2 Übersicht Kommunikations-Dienste

#### Allgemeines

Über die Kommunikationsschnittstellen können verschiedene Informationen übertragen werden, wie z.B. Prozessdaten oder Meldungen.

Die Kommunikations-Dienste bestimmen, welche Informationen übertragen werden. Durch Projektierung der Kommunikationsbausteine werden die Kommunikations-Dienste festgelegt.

#### Kommunikations-Dienste

Kommunikations-Dienst	Beschreibung	zu projektierende Kommunikationsbausteine
Meldesystem	Aufbau von Warnungs- und Störmeldesystemen	spezielle Meldebausteine: @MSC, MER..., MSI...
Prozessdaten	Übertragung von Prozessdaten (Soll- und Istwerte)	Sende- und Empfangsbausteine: CTV, CRV, CCC4, CDC4
Service	Diagnose und Analyse von CPU-Programmen / CFC	Service-Funktionsbaustein: SER
Uhrzeitsynchronisation	Uhrzeitsynchronisation aller eingesetzten CPUs (z.B. um Meldungen mit Zeitstempel zu vergleichen).	spezielle Funktionsbausteine: RTC...
Datenaufzeichnung (Trace)	Verlauf von Prozeßgrößen aufzeichnen	@TCP, TR...
S7-Kommunikation	Bedienen und Beobachten von CPU-Programmen / CFC	Kommunikations-Funktionsbaustein: S7OS

Tabelle 3-9 Übersicht Kommunikations-Dienste

### 3.1.3 Anschlüsse der Kommunikationsbausteine

#### 3.1.3.1 Initialisierungsanschluss CTS

Kommunikationsbausteine, die auf eine Datenschnittstelle zugreifen, besitzen einen CTS-Anschluss.

#### Angaben am Initialisierungsanschluss

Am CTS-Anschluss wird angegeben:

- der projektierte Baugruppenname der Kommunikationsbaugruppe

Regeln für Baugruppennamen:

- die Namenslänge beträgt 1 - 6 Zeichen
- 1. Zeichen: A - Z
- 2. - 6. Zeichen: A - Z, 0 - 9, \_

2. Stecker der Datenschnittstelle, wenn die Datenschnittstelle auf einer Kommunikationsbaugruppe CP50M0, CP50M1 oder CP51M1 liegt

Regeln für Steckerbezeichnung:

- Eingabe von „.“ nach Baugruppenname
- die Namenslänge nach „.“ beträgt 3 Zeichen
- „X01“, „X02“ oder „X03“

**Beispiel Angaben an CTS**

Beispiel-Projektierung eines Baugruppenträgers:

Steckplatz	Baugruppe	projektierte Baugruppennamen in HWKonfig	mögliche Angabe am Anschluss CTS
S01	CPU550	„D01P01“	„D01P01“
S03	CP50M1	„KOPPEL“	„KOPPEL“
S04	CP50M1	„KOMM1“	„KOMM1.X01“ „KOMM1.X02“
S06	CP51M1	„TCPIP“	„TCPIP.X01“

Tabelle 3-10 Beispiel-Projektierung eines Baugruppenträgers

### 3.1.3.2 Adressanschlüsse AT, AR und US

**Allgemeines**

Kommunikationsbausteine, die auf eine Datenschnittstelle zugreifen können, besitzen einen Adressanschluss.

**Typen der Adressanschlüsse**

Je nach Bausteintyp wird zwischen drei Typen der Adressanschlüsse unterschieden:

- Anschluss AT: bei Sendern vorhanden
- Anschluss AR: bei Empfängern vorhanden
- Anschluss US: bei Funktionsbausteinen vorhanden, die einen Send- und einen Empfangskanal bearbeiten

**Mögliche Angaben am Adressanschluss**

Die Angaben am Adressanschluss sind vom Typ AT, AR oder US unabhängig. Mögliche Angaben sind:

- „Kanalname“
- „Kanalname.Adressstufe 1“
- „Kanalname.Adressstufe 1.Adreßstufe 2“

**Kanalname**

- Der Kanalname adressiert einen Kanal auf einer Datenschnittstelle.
- Sender und Empfänger, die mit gleichen Kanalnamen auf eine Datenschnittstelle zugreifen, kommunizieren miteinander.
- Der Kanalname besteht aus maximal 8 ASCII-Zeichen, ausgenommen sind „Punkt“ und „@“.



**HINWEIS**

Die Mehrfach-Projektierung eines Kanalnamens wird nicht überprüft. Der Projektteur muss den Kanalnamen auf einer Datenschnittstelle für jeden Sender/Empfänger an den Anschlüssen AT, AR oder US eindeutig vergeben. Wird diese Bedingung nicht erfüllt, dann:

- kommt es zu unkoordinierter Mehrfachbenutzung von Sender/Empfänger
- meldet sich der Sender/Empfänger eventuell mit einem Kommunikationsfehler ab

Ausnahmen:

- Beim Übertragungsmodus „Select“ sind mehrere Sender erlaubt.
  - Beim Übertragungsmodus „Multiple“ sind mehrere Empfänger erlaubt.
-

### Adressstufen

- Es gibt die Adressstufe 1 und die Adressstufe 2.
- Einige Kopplungen, wie z.B. PROFIBUS und Industrial Ethernet benötigen zur Datenübertragung die Angabe von Adressstufen. Bei der Baugruppenträgerkopplung werden z.B. keine Adressstufen angegeben.
- Adressstufe 1 ist maximal 14 Zeichen lang, Adressstufe 2 maximal 20 Zeichen.
- Bedeutung und Inhalt der Adressstufen sind bei der jeweiligen Kopplung beschrieben.

### 3.1.3.3 Übertragungsmodus, Anschluss MOD

#### Übersicht

Für die verschiedenen Anforderungen in der Kommunikation gibt es fünf verschiedene Übertragungsmodi:

- Handshake
- Refresh
- Select
- Multiple
- Image

#### Auswahl des Übertragungsmodus Übertragungsmodus „Handshake“

Der Übertragungsmodus wird am Anschluss MOD des entsprechenden Senders oder Empfängers angegeben.

Der Übertragungsmodus „Handshake“ wird eingesetzt,

- wenn kein Informationsverlust durch Datenüberschreibung auftreten darf und
- wenn es zu jedem Sender genau einen Empfänger geben soll.

„Handshake“ beschreibt eine sequentielle Kanalbearbeitung. Der Sender legt erst einen neuen Datensatz im Kanal ab, nachdem der Empfänger den Erhalt des letzten quittiert hat. Für den Datenaustausch gibt es einen Nutzdatenpuffer.

Die Nutzdaten werden vom Sender in einem Arbeitszyklus in den Kanal eingetragen und vom Empfänger in einem Arbeitszyklus ausgelesen.

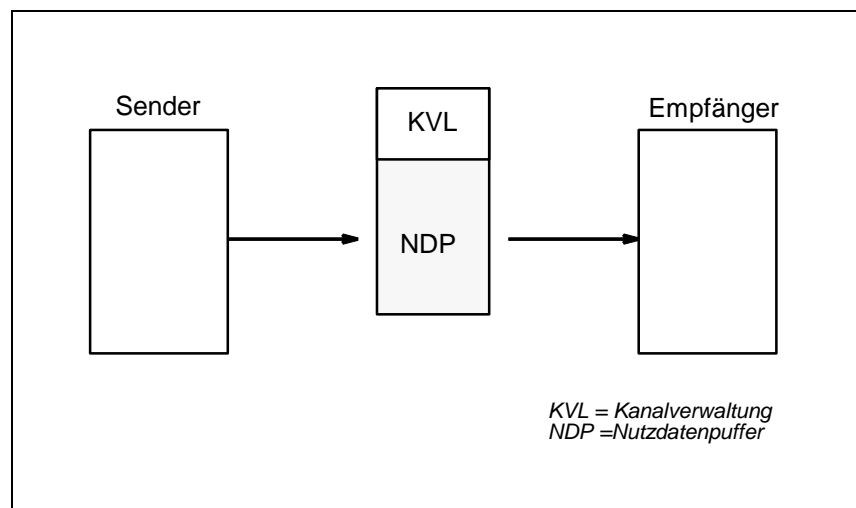


Bild 3-1 Datenübertragung im Übertragungsmodus „Handshake“

### Übertragungsmodus „Refresh“

Der Übertragungsmodus „Refresh“ wird eingesetzt,

- wenn einem Empfänger immer die neuesten Daten zur Verfügung gestellt werden sollen und
- wenn es zu jedem Sender genau einen Empfänger geben soll.

„Refresh“ ist ein überschreibender Datenaustausch. Der Sender legt immer den neuesten Datensatz im Kanal ab, ohne dass der Empfänger den Erhalt des letzten Datensatzes quittiert. Für den Datenaustausch gibt es zwei Nutzdatenpuffer, die als Wechselpuffersystem genutzt werden. Der Sender teilt mit, in welchem Puffer die neuesten Daten stehen.

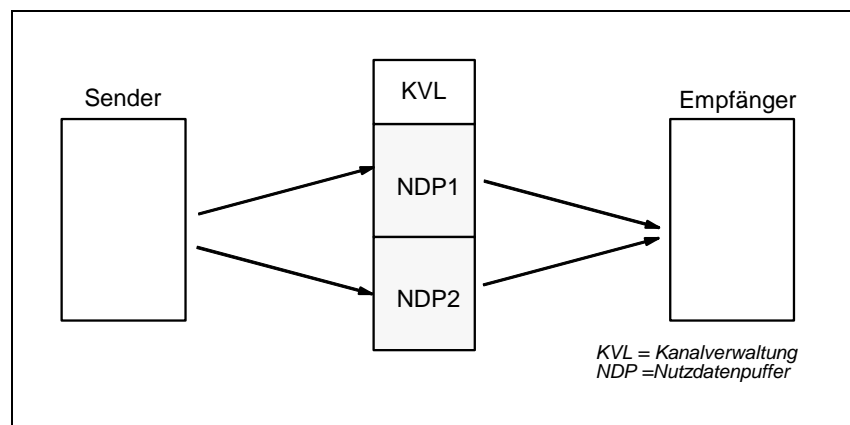


Bild 3-2 Datenübertragung im Übertragungsmodus „Refresh“

### Übertragungsmodus „Select“

Der Übertragungsmodus „Select“ wird eingesetzt,

- wenn kein Informationsverlust durch Datenüberschreibung auftreten darf und
- wenn es zu einem Empfänger beliebig viele Sender geben soll.

„Select“ beschreibt eine sequentielle Kanalbearbeitung. Wenn der Empfänger den Erhalt des letzten Datensatzes quittiert, dann legt ein Sender einen neuen Datensatz im Kanal ab. Für den Datenaustausch gibt es einen Nutzdatenpuffer. Eine Kanalverwaltung regelt den Datenverkehr.

Alle projektierten Sender benutzen denselben Nutzdatenpuffer. Es gibt keine festgelegte Sendereihenfolge für die Sender. Wer zuerst kommt, darf senden. Um eine kontrollierte Datenübertragung zu erreichen, darf nur an einem Sender am Anschluss EN „1“ angegeben werden.

Die Sender müssen in einer kleineren Abtastzeit als der Empfänger projektiert werden.

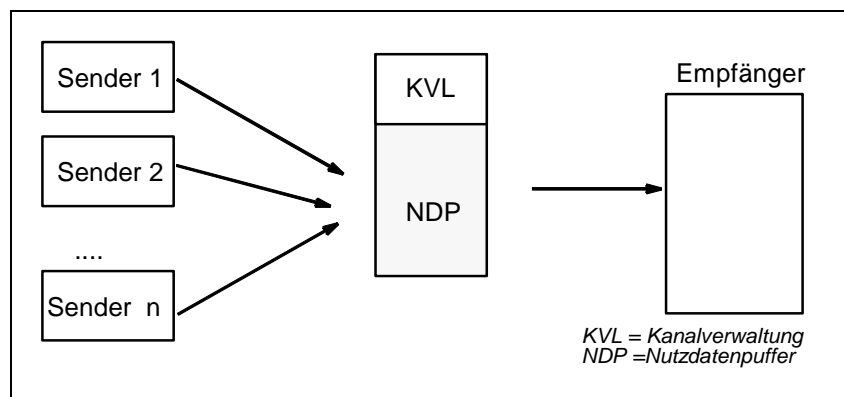


Bild 3-3 Datenübertragung im Übertragungsmodus „Select“

### Übertragungsmodus „Multiple“

Der Übertragungsmodus „Multiple“ wird eingesetzt,

- wenn Empfängern immer die neuesten Daten zur Verfügung gestellt werden sollen und
- wenn es zu jedem Sender beliebig viele Empfänger geben soll.

„Multiple“ ist ein überschreibender Datenaustausch. Der Sender legt immer den neuesten Datensatz im Kanal ab, ohne dass der Empfänger den Erhalt des letzten Datensatzes quittiert.

Wenn ein Sender einen Puffer überschreibt, aus dem ein Empfänger gerade liest, dann verwirft der Empfänger die zuletzt empfangenen Daten. Der Empfangsbetrieb wird im nächsten Arbeitszyklus wiederholt.

Für den Datenaustausch gibt es zwei Nutzdatenpuffer, die als Wechselpuffersystem genutzt werden. Der Sender teilt mit, in welchem Puffer die neusten Daten stehen.

Die Empfänger müssen in der gleichen oder kleineren Abtastzeit wie der Sender projektiert sein (Empfänger müssen also schneller arbeiten).

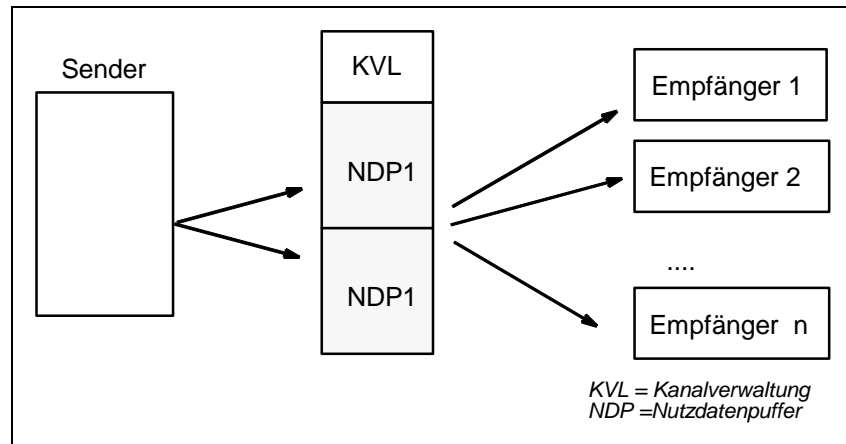


Bild 3-4 Datenübertragung im Übertragungsmodus „Multiple“

### Übertragungsmodus „Image“

Der Übertragungsmodus „Image“ wird bei der FM 458-1 DP zur Kommunikation über die PROFIBUS DP-Schnittstelle eingesetzt,

- wenn allen Empfängern, die in einer Task projektiert sind, Daten zur Verfügung gestellt werden sollen, die aus dem gleichen DP-Zyklus stammen und
- wenn alle Sender, die in der gleichen Task projektiert sind, ihre Daten im gleichen DP-Zyklus an die DP-Slaves senden sollen.

Dazu synchronisieren sich Sende- bzw. Empfangs-FBs innerhalb einer Task, um in sich konsistente Daten abzuliefern. Sie bilden jeweils eine sogenannte **„Konsistenzgruppe“**. Alle Empfangs-FBs einer solchen Konsistenzgruppe holen ihre Nutzdaten aus einem gemeinsamen Wechselpuffer und alle Sende-FBs legen ihre Daten in einem solchen ab.

„Image“ ist ein überschreibender Datenaustausch (siehe Refresh). Für den Datenaustausch gibt es zwei Nutzdatenpuffer, die als Wechselpuffersystem genutzt werden

Dieser Übertragungsmodus ist nur für die PROFIBUS DP-Schnittstelle der Applikationsbaugruppe FM458-1 DP erlaubt

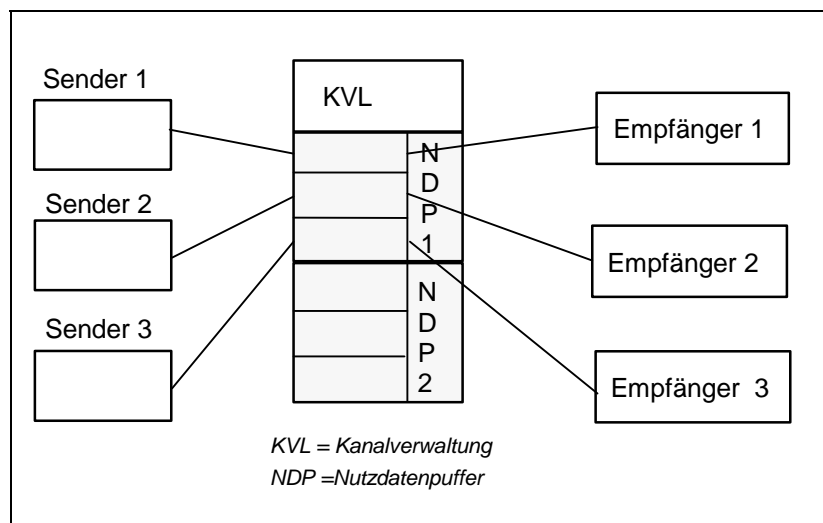


Bild 3-5 Datenübertragung im Übertragungsmodus „Image“

### 3.1.3.4 Firmwarestatus, Anschluss ECL, ECO

#### Allgemeines

Kopplungs-Zentralbausteine, die mit einer Firmware kommunizieren (z.B. @PRODP oder @TCPIP), besitzen die Ausgänge ECL und ECO.

#### Funktion

Die Ausgänge ECL und ECO zeigen den Zustand der entsprechenden Firmware an:

- ECL=0 und ECO=0:  
Die Firmware befindet sich im fehlerfreien Zustand
- ECL=0 und ECO>0:  
Es liegt ein Fehler der Firmware vor, der vom Projektteur bzw. Anwender behoben werden kann. Die Fehlerursache ist in den Kapiteln zu den einzelnen Kopplungen beschrieben
- ECL>0 und ECO>0:  
Es liegt ein irreparabler Fehler der Firmware vor

### 3.1.3.5 Zustandsanzeige, Ausgang YTS

#### Allgemeines

Der Baustein gibt am Ausgang YTS einen Fehlercode oder den momentanen Übertragungszustand aus.

#### Angezeigte Fehler

- echte (schwere) Laufzeit-Fehler
- Projektierungsfehler, die bei der Systeminitialisierung erkannt werden und an der Siebensegmentanzeige der CPU durch ein blinkendes „C“ angezeigt werden.
- vorübergehende Zustandsanzeigen und Warnungen

#### Fehlerdiagnose

Mit dem CFC kann der Wert am Ausgang YTS als Dezimalzahl gelesen werden.

**Weitere Informationen**

zur Bedeutung der Dezimalzahl siehe Onlinehilfe „SIMADYN D, Hilfe zu Ereignissen“. (Drücken Sie die F1-Taste im CFC und rufen Sie das Topic „Hilfe zu Ereignissen“ unter „CFC für SIMADYN D“ auf.)

**3.1.4 Funktionsweise der Kopplungen****Allgemeines**

Eine Kopplung funktioniert wie folgt:

- CPUs tauschen mit einer Kopplungsbaugruppe Daten über den Rückwandbus aus
- Bei seriellen Kopplungen (z.B. bei TCP/IP) werden die Daten dann noch von der Firmware auf der Kopplungsbaugruppe so „umstrukturiert“ und „verpackt“, dass sie dem geforderten Telegrammaufbau und Protokoll entsprechen
- Wenn der Kommunikationspartner ebenfalls SIMATIC TDC ist (Baugruppenträgerkopplung, Koppelspeicherkopplung), dann werden die Daten nicht aufbereitet

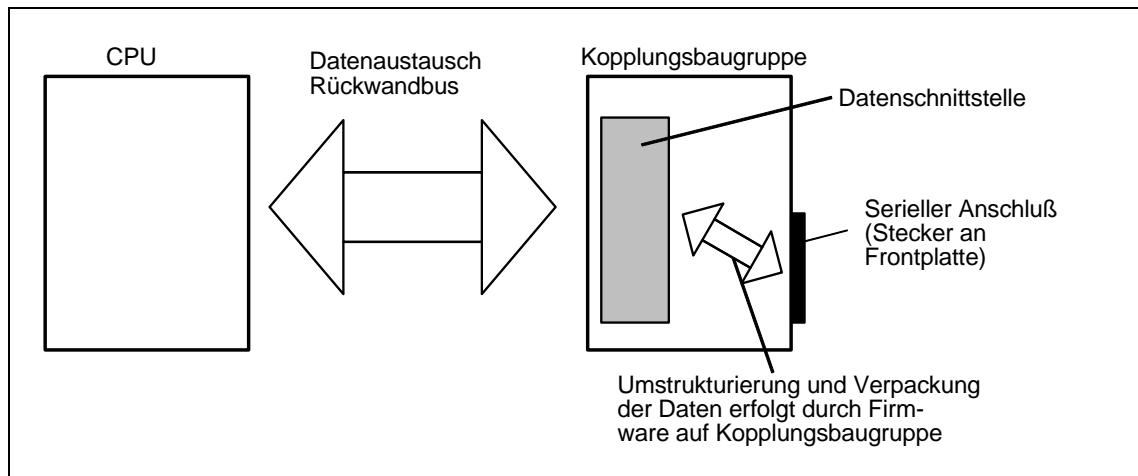


Bild 3-6 Datenaustausch zwischen CPU und Kopplungsbaugruppe

**Zugriff auf Datenschnittstelle**

Da die Datenschnittstellen auf externen Kopplungsbaugruppen und nicht CPU-lokal auf einer CPU liegen, können sie auch von allen CPUs eines Baugruppenträgers benutzt werden. Voraussetzung für die Benutzung einer Datenschnittstelle ist, dass die CPU und die Kopplungsbaugruppe denselben Busanschluss besitzen.

**HINWEIS**

Bei der CPU-lokalen Kopplung liegt die Datenschnittstelle auf dem CPU-lokalen RAM der CPU. Für alle anderen CPUs eines Baugruppenträgers ist diese Datenschnittstelle nicht zugänglich. Sie kann ausschließlich von der CPU benutzt werden, auf der sie projiziert wurde.

**Grund-  
initialisierung**

Die Grundinitialisierung einer Kopplungsbaugruppe erfolgt immer während des Systemanlaufs.

Die erste CPU im führt folgende Aufgaben durch:

- Überprüfen, ob die Kopplungsbaugruppe „ansprechbar“ ist
- Formatieren der Datenschnittstelle

**Projektieren der  
Kopplungs-  
baugruppe**

Die gewünschte Kopplungsbaugruppe wird in HWKonfig projektiert. Für die Grundinitialisierung einer Kopplung müssen keine expliziten Projektierungsschritte durchgeführt werden.

### 3.1.4.1 Kopplungs-Zentralbausteine

**Funktionen der  
Kopplungs-  
Zentralbausteine**

Die Kopplungs-Zentralbausteine haben für eine Kopplung folgende Funktionen:

- Initialisierung:
  - Kopieren der projektierten Initialisierungsinformation (werden an den Initialisierungsanschlüssen projektiert) auf die Datenschnittstelle
  - Feststellen, ob sich die Datenschnittstelle in einem fehlerfreien Zustand befindet
- Freigabe:
  - Nach der Initialisierung durch die Kopplungs-Zentralbausteine und durch die Firmware der Kopplungsbaugruppe gibt der Kopplungs-Zentralbaustein die Kopplung für alle Sender und Empfänger auf dem gleichen Baugruppenträger frei. Die Datenübertragung kann dann beginnen
  - Die Freigabe einer Kopplung erfolgt aus zeitlichen Gründen immer erst im Betriebszustand RUN nach mehreren Abtastzeiten
- Überwachung:
  - Die Kopplungs-Zentralbausteine geben an ihren Baustein-  
ausgängen Auskunft über den Zustand der Kopplung sowie ggf.  
über den Zustand der Firmware

**Projektieren der  
Kopplungs-  
Zentralbausteine**

Beim Projektieren müssen folgende Punkte beachtet werden:

- Für jede Kopplung muss genau ein Kopplungs-Zentralbaustein projektiert werden
- Die Kopplungs-Zentralbausteine können alle auf einer CPU eines Baugruppenträgers projektiert werden oder auf verschiedenen CPUs eines Baugruppenträgers verteilt projektiert werden
  - Projektierung aller Kopplungs-Zentralbausteine auf einer CPU erleichtert z.B. die Diagnose



- Kopplungs-Zentralbausteine haben keinerlei Sender- oder Empfängerfunktionalität
- Alle Kopplungs-Zentralbausteine müssen in einer Abtastzeit  $32 \text{ ms} \leq TA \leq 256 \text{ ms}$  projiziert werden

#### Fehler

Der Kopplungs-Zentralbaustein macht einen Eintrag ins Kommunikations-Fehlerfeld und bearbeitet die Kopplungsbaugruppe nicht mehr, wenn

- ein Kopplungs-Zentralbaustein während der Initialisierung einen Fehler feststellt,
- eine Firmware keine Reaktion oder Fehlverhalten zeigt,
- der Kopplungs-Zentralbaustein auf der falschen Kommunikationsbaugruppe arbeitet.

### 3.1.4.2 Sender und Empfänger

#### Allgemeines

Sender und Empfänger sind:

- Funktionsbausteine, die schreibend und/oder lesend auf die Datenschnittstelle einer Kopplung zugreifen

Teil des Kommunikations-Dienstes, der die Kopplung benutzt Beispiele für Sender:

- Meldeauswerte-Funktionsbaustein MSI:  
kopiert Meldungen aus dem Meldepuffer in eine Datenschnittstelle

Prozessdaten-Sendebaustein CTV Beispiel für Empfänger:

- Prozessdaten-Empfangsbaustein CRV

#### Angaben an den Anschlüssen

Da Sender und Empfänger nicht zwischen den einzelnen Kopplungen unterscheiden, muss an den Bausteineingängen der Sender und der Empfänger kein Kopplungstyp angegeben werden.

#### Anschlüsse der Sender/Empfänger

- Anschluss CTS für die Angabe der Kopplungsbaugruppe
- Adressanschluss AR, AT oder US für die Angabe von Kanalnamen und kopplungsspezifischen Adressen

**Synchronisieren zwischen Sendern und Empfängern**

Bevor Sender und Empfänger Daten austauschen können, müssen sie sich zuerst „erkennen“ und aufeinander „synchronisieren“:

- Die Erkennung erfolgt über die Projektierungsangaben an den Anschlüssen CTS und AT, AR oder US
- Die Synchronisation ist nur dann möglich, wenn
  - ein Sender seinen Partner als Empfänger identifiziert (oder umgekehrt),
  - die Länge des reservierten Datenbereichs übereinstimmt,
  - die Nutzdatenstruktur kompatibel ist,
  - der Übertragungsmodus identisch ist (Angabe am Anschluss MOD bei Sendern/Empfängern).

Wird eine dieser Bedingungen nicht erfüllt, dann meldet sich der synchronisierende Sender/Empfänger mit einem Kommunikationsfehler ab.

**3.1.4.3 Kompatible Nutzdatenstrukturen**

**Allgemeines**

Die Nutzdatenstrukturen beinhalten die Information über den Aufbau der übertragenen Nutzdaten:

- Angaben über Lage und Datentypen der zugehörigen Nutzdaten

Für den Datenaustausch zwischen Sender und Empfänger müssen die Nutzdaten von Sender und Empfänger gleich strukturiert sein.

**Datentypen**

Als Datentypen werden folgende standardisierte Datentypen verwendet:

standardisierter Datentyp	SIMATIC TDC Datentyp	Länge in Bytes
Integer 16	Integer	2
Integer 32	Double Integer	4
Unsigned 8	Bool, Byte	1
Unsigned 16	Word	2
Unsigned 32	Double Word	4
Floating Point	Real, SDTIME	4
Octet-String	-	1
Time and Date	-	6

*Tabelle 3-11 Standardisierte Datentypen*

**HINWEIS**

Es werden nicht die Anschlusstypen von SIMATIC TDC (z.B. SDTIME) als Datentypen verwendet, da der Koppelpartner nicht immer ein Funktionsbaustein von SIMATIC TDC sein muss.

- **Octet-String**

Ein Octet-String ist ein unstrukturierter Datentyp, der nicht an Bausteinanschlüssen erscheint (siehe Kapitel "Kanal-Rangierbausteine CCC4 und CDC4").

- **Time and Date**

Zusammengesetzter Datentyp für die Uhrzeit, der nicht an Bausteinanschlüssen erscheint (siehe Kapitel "Kommunikations-Dienst Meldesystem").

Wertebereich

- **1. Octet und 2. Octet:**

Geben das Datum relativ zum 1.1.1984 an.

Granularität = 1 Tag

$0 \text{ Tage} \leq d \leq 65535 \text{ Tage}$

- **3. Octet bis 6. Octet:**

Geben die Zeit zwischen 00:00 Uhr und 24:00 Uhr an.

Granularität = 1ms

$0 \text{ ms} \leq x \leq 86400000 \text{ ms}$

Im sechsten Octet sind die ersten 4 Bits un belegt

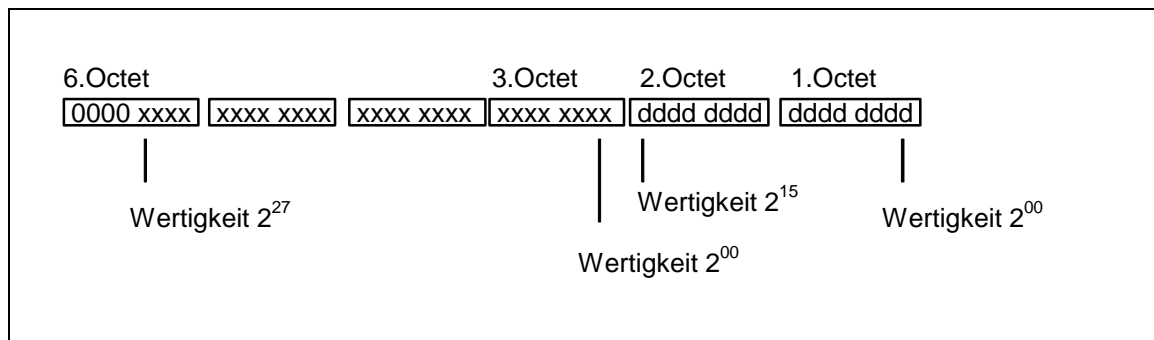


Bild 3-7 Time and Date

### 3.1.4.4 Anzahl der Kopplungsbaugruppen in einem Baugruppenträger

#### Übersicht

Die Anzahl der CS-Kopplungsbaugruppen (CP50M0, CP51M1, CP5100 und CP52A0) wird durch zwei Systemgrenzen beschränkt:

- **Baugruppenträgergröße**  
Der größte Baugruppenträger im System SIMATIC TDC umfasst 21 Steckplätze. Da mindestens eine CPU in einem Baugruppenträger stecken muß, verbleiben theoretisch noch 20 Steckplätze.
- **verfügbarer Adressraum**  
Diese Grenze wird in der Praxis jedoch selten erreicht. Für die CP-Baugruppen steht auf dem Bus je 144 MByte Adressraum zur Verfügung.

#### Belegter Adressraum

Die einzelnen CP-Baugruppen belegen immer einen konstanten Adressraum auf dem Rückwandbus.

- **Beispiel**  
Die CP52A0 belegt immer 2 MByte auf dem Bus.

Baugruppentyp	Belegter Adressraum
CP52A0	2 MByte
CP5100	4 MByte
CP50M0 CP50M1	2 MByte bzw. 10 MByte als Koppelspeicher
CP51M1	1 MByte bzw. 9 MByte als Koppelspeicher

Tabelle 3-12 Belegter Adressraum

### 3.1.4.5 Reorganisation einer Datenschnittstelle

**Allgemeines** Eine Datenschnittstelle kann neu reorganisiert werden, ohne dabei den Betriebszustand RUN zu unterbrechen oder zu beeinträchtigen.

**Reorganisieren der Datenschnittstelle** Einige Kopplungs-Zentralbausteine besitzen einen Anschluss CDV (Datentyp Bool). Bei positiver Flanke am Anschluss CDV sperrt der Kopplungs-Zentralbaustein die Kopplung und formatiert nach ca. 10 Sekunden die Datenschnittstelle. Anschließend wird die Datenschnittstelle wieder freigegeben.

Alle Sender/Empfänger gehen während der Sperrung und Reorganisation in einen Wartezustand über. Nach erneuter Freigabe erfolgt die Neuansmeldung und Synchronisation von Kanälen wie bei einem Systemanlauf.

Für lokale Datenschnittstellen auf einer CPU wird diese Angabe ignoriert.

## 3.2 Baugruppenträger-lokale Kopplungen

**Übersicht** Zu den Baugruppenträger-lokalen Kopplungen zählen:

- CPU-lokale Kopplung
- Direkte CPU-CPU-Kopplung
- Koppelspeicher-Kopplung

### 3.2.1 CPU-lokale Kopplung

**Allgemeines** Die CPU-lokale Kopplung benötigt keine Kopplungsbaugruppe. Dieser Kopplungstyp kann nur von den Funktionsbausteinen benutzt werden, die sich auf derselben CPU wie die Datenschnittstelle befinden. Die Datenschnittstelle liegt immer auf der jeweiligen CPU und hat eine Größe von 1 MByte.

**Anwendung** Die Kopplung wird hauptsächlich dazu eingesetzt, in sich abgeschlossene Aufgaben (z.B. eine Regelung) mit definierten Schnittstellen versehen zu können. Damit ist es auf einfache Weise möglich, falls im Verlaufe einer Projektierung ein CPU „überlastet“ wird, ohne größeren Projektierungsaufwand die komplette Aufgabe auf eine andere CPU zu übertragen und dann z.B. über die Datenschnittstelle im Koppelspeicher zu kommunizieren. Dazu muss dann lediglich an allen Kommunikations-Funktionsbausteinen die Angabe am Anschluss CTS geändert werden.

**Initialisierung und Überwachung** Die Kopplungs-Initialisierung und -Überwachung wird vom Zentralbausteinen im zyklischen Betrieb durchgeführt. Die @LOCAL Kopplung ist also nicht mit Beginn des zyklischen Betriebs für alle Sender/Empfänger freigegeben, sondern verzögert sich um mehrere Arbeitszyklen. Nach Freigabe der Kopplung überwacht der Zentralbaustein @LOCAL die Kopplung.

**Projektierung** Für die Kopplungs-Initialisierung und -Überwachung ist ein Kopplungs-Zentralbaustein @LOCAL zu projektieren.

An den Anschlüssen AT, AR oder US der Sende-/Empfangsbausteine ist bei der CPU-lokalen Kopplung nur der Kanalname anzugeben. Angaben für Adressstufe 1 und 2 sind nicht zu projektieren. Sender und Empfänger mit gleichen Kanalnamen kommunizieren miteinander.

### 3.2.2 Direkte CPU-CPU-Kopplung

**Allgemeines** Die Datenschnittstelle für die direkte CPU-CPU-Kopplung liegt lokal auf einer CPU und hat eine Größe von 1 MByte.

Die Datenschnittstelle muss auf der CPU angelegt werden, welche die Empfangsbausteine enthält, da die Empfangsbausteine über den Rückwandbus nicht auf den Speicher der anderen CPU zugreifen können. Dies ist nur von den Sendebausteinen aus möglich.

Eine Projektierung der Datenschnittstelle auf der Sendeseite führt zu einem Kommunikationsfehler („C“) am Empfangsbaustein.

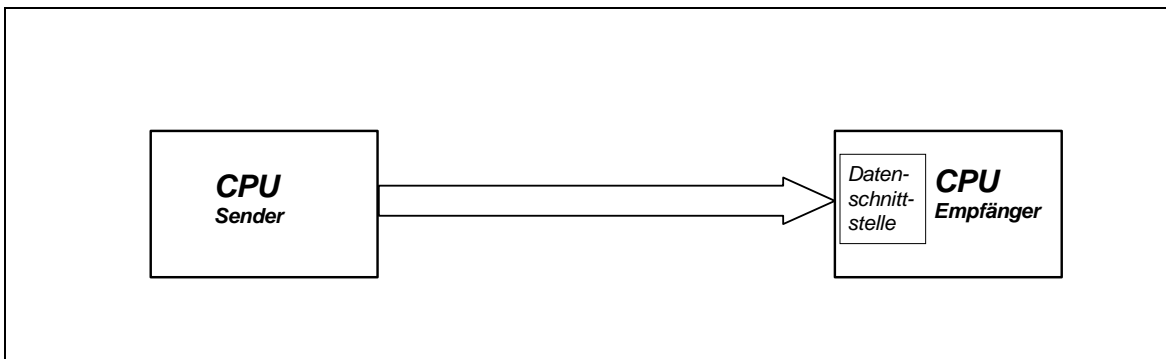


Bild 3-8 Direkte CPU-CPU-Kopplung

**Anwendung** Die direkte CPU-CPU-Kopplung wird für die Übertragung von Prozessdaten zwischen verschiedenen CPUs eines Baugruppenträgers eingesetzt. Im Unterschied zur Koppelspeicherkopplung werden größere Datenmengen effektiver übertragen. Bei dieser Kopplung erreicht eine Seite (Sender) die Datenschnittstelle über den Rückwandbus, während die andere zeitsparend lokal zugreifen kann.

**Initialisierung und Überwachung** Für die Kopplungs-Initialisierung und -Überwachung ist auf der CPU, auf der eine Datenschnittstelle angelegt werden soll, ein Zentralbaustein @LOCAL zu projektieren.

**Projektierung** Die Kopplungs-Initialisierung und -Überwachung wird vom Zentralbaustein @LOCAL im zyklischen Betrieb durchgeführt. Die Kopplung ist also nicht mit Beginn des zyklischen Betriebs für alle Sende-/Empfangsbausteine freigegeben, sondern verzögert sich um mehrere Arbeitszyklen. Nach Freigabe der Kopplung überwacht der Zentralbaustein @LOCAL die Kopplung.

Die direkte CPU-CPU-Kopplung kann nur von Sende-/Empfangsbausteinen benutzt werden, die auf dem gleichen Baugruppenträger projektiert sind.

An den Anschlüssen AT- oder AR der Sende-/Empfangsbausteine ist bei der direkten CPU-CPU-Kopplung der Kanalname und zusätzlich ein „E“ als Adressstufe 1 anzugeben. Bei der CPU, die über den Rückwandbus zugreifen muss (Sender), ist die Angabe optional.

Angaben für Adressstufe 2 sind nicht zu projektieren. Sender und Empfänger mit gleichen Kanalnamen kommunizieren miteinander.

Als Übertragungsmodus sind nur „Handshake“ oder „Refresh“ möglich.

### 3.2.3 Koppelspeicher-Kopplung

#### Allgemeines

Die Datenschnittstelle für die Koppelspeicher-Kopplung liegt auf einem Koppelspeicher und hat eine Größe von 1 MByte.

Der Hardwareaufbau wird mittels der Baugruppe CP50M0, CP50M1, CP51M1 oder CP53M0 realisiert.

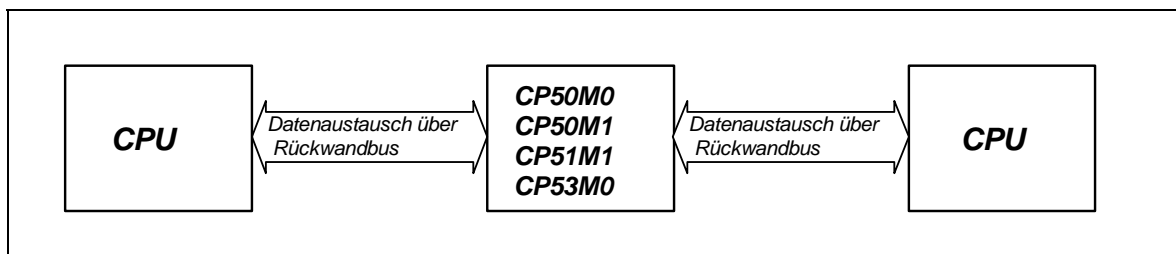


Bild 3-9 Koppelspeicher-Kopplung

#### Anwendung

Die Koppelspeicher-Kopplung wird für die Übertragung von Daten zwischen verschiedenen CPUs eines Baugruppenträgers eingesetzt. Im Unterschied zu \$-Signalen werden größere Datenmengen effektiver übertragen.

#### Initialisierung und Überwachung

Für die Kopplungs-Initialisierung und -Überwachung ist auf einer beliebigen CPU des Baugruppenträgers ein Zentralbaustein @GLOB zu projektieren.

#### Projektierung

Die Kopplungs-Initialisierung und -Überwachung wird vom Zentralbaustein @GLOB im zyklischen Betrieb durchgeführt. Die Kopplung ist also nicht mit Beginn des zyklischen Betriebs für alle Sende-/Empfangsbausteine freigegeben, sondern verzögert sich um mehrere Arbeitszyklen. Nach Freigabe der Kopplung überwacht der Zentralbaustein @GLOB die Kopplung.

Die Koppelspeicher-Kopplung kann nur von Sende-/Empfangsbausteinen benutzt werden, die auf demselben Baugruppenträger projektiert sind.

An den Anschlüssen AT-, AR- oder US der Sende-/Empfangsbausteine ist bei der Koppelspeicher-Kopplung nur der Kanalname anzugeben. Angaben für Adressstufe 1 und 2 sind nicht zu projektieren. Sender und Empfänger mit gleichen Kanalnamen kommunizieren miteinander.

### 3.3 Baugruppenträger-Kopplung CP52M0

#### 3.3.1 Anwendungsgebiete

Über den Speicher im **GlobalDataMemory (GDM)** können Daten baugruppenträgerübergreifend zwischen allen im System befindlichen CPU-Baugruppen, ausgetauscht werden.

Da bis zu 44 Baugruppenträger über den zentralen Speicher synchron gekoppelt werden können, sind damit 836 CPU-Baugruppen im Maximalausbau einsetzbar.

Für GDM wird ein eigener Baugruppenträger verwendet (im folgenden **GDM-Baugruppenträger** genannt). In den Baugruppenträger UR5213 mit 21 Steckplätzen werden die **Speicherbaugruppe CP52M0 (Steckplatz 1)** und eine entsprechende Anzahl **Schnittstellenbaugruppen CP52IO (Steckplätze 2-12)** eingesetzt.

In jedem Baugruppenträger, der mit GDM gekoppelt ist (im folgenden **Koppel-Baugruppenträger** genannt), muss sich eine **Zugriffsbau- gruppe CP52A0** befinden. Über Glasfaser-LWL-Kabel werden diese Baugruppenträger sternförmig an GDM angeschlossen.

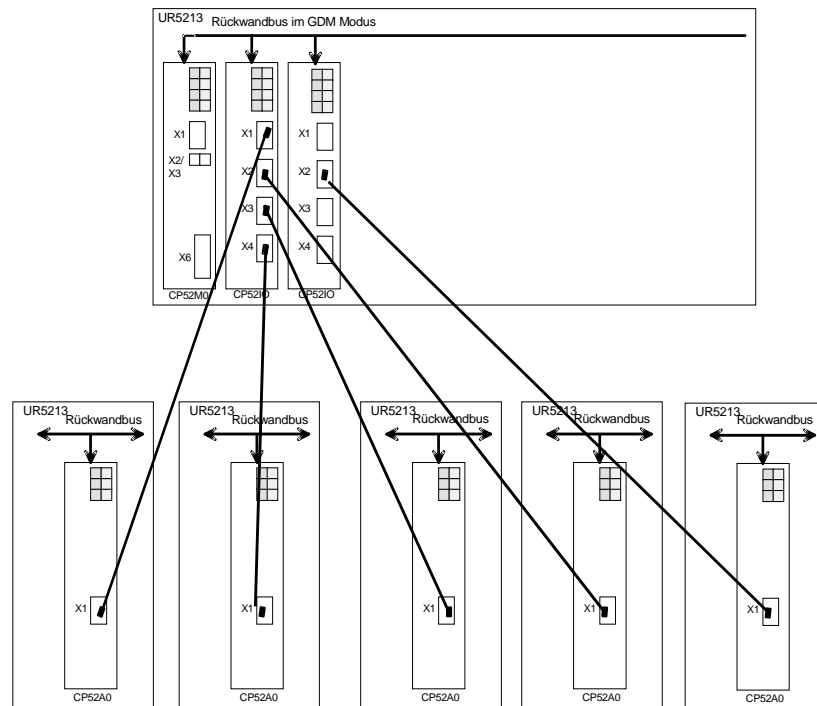


Bild 3-10 Beispiel eines GDM-Systems mit 5 LWL-gekoppelten Baugruppenträgern



**Funktion der CP52M0**

Auf der GDM-Speicherbaugruppe CP52M0 befindet sich der zentrale Speicher des GlobalDataMemory-Systems mit einer Größe von 2MByte. Über ihn wird der gesamte Datenverkehr zwischen den Prozessoren in den gekoppelten Baugruppenträgern abgewickelt.

Der Datenaustausch zwischen den GDM-Schnittstellenbaugruppen CP52IO und der CP52M0 erfolgt über den Rückwandbus.

**Literatur zu diesem Thema:**

- GDM-Hardwareokumentation

**3.3.2 Ein- und Ausschaltverhalten**

- Die Einschaltreihenfolge der Baugruppenträger ist beliebig.
- Wird der GDM-Baugruppenträger im laufenden Betrieb abgeschaltet, werden in den Koppel-Baugruppenträgern alle Telegrammkanäle abgebaut. Nachdem der GDM-Baugruppenträger wieder eingeschaltet ist, werden die Telegrammkanäle wieder initialisiert.

**HINWEIS**

Wird bei laufenden Koppel-Baugruppenträgern der GDM-Baugruppenträger zugeschaltet, ist kurzzeitig mit einem erhöhten Rechenzeitbedarf für den Verbindungsaufbau bei den über die Baugruppenträgerkopplung kommunizierenden CPUs zu rechnen. Dies kann bei bereits sehr stark ausgelasteten CPUs zur Anzeige eines ‚E‘ (Fehler im Aufgabenverwalter) an der Anzeige führen.

- Alle Koppel-Baugruppenträger können aus dem laufenden Betrieb aus- und wieder zugeschaltet werden. Die Koppelpartner können dann keine Daten mehr von diesem Baugruppenträger empfangen oder an ihn senden.
- Wird ein Koppel-Baugruppenträger aus- und wieder zugeschaltet, so ist die Kommunikation zwischen den verbleibenden Teilnehmern (GDM-Baugruppenträger und maximal 43 Koppel-Baugruppenträger mit je einer CP52A0) davon unberührt.
- Abgeschaltete Koppel-Baugruppenträger können umprojektiert und wieder zugeschaltet werden. Auch kann die Anzahl der Sender und Empfänger verändert werden (z.B. falls ein Sender zu wenig projektiert worden ist).

**HINWEIS**

Änderungen der Datenlänge führen bei bestehender Kopplung zu Kommunikationsfehlern im Baugruppenträger der geänderten Projektierung. In diesem Fall muss auch der GDM-Baugruppenträger zurückgesetzt werden.

- Sobald der abgeschaltete Baugruppenträger wieder zugeschaltet wird, erfolgt ein neuer Verbindungsaufbau zwischen dem neu zugeschalteten Koppelpartner und dem GDM- Baugruppenträger. Die Telegrammkanäle werden wieder initialisiert und die Kommunikation wird wieder aufgenommen, ohne die Koppelpartner rücksetzen zu müssen.

### 3.3.3 Synchronisations- und Triggermöglichkeiten

Es stehen auf der Zugriffsbaugruppe CP52A0 folgende Synchronisations- und Triggersignale zur Verfügung:

1. Grundtakt
2. Busalarm
3. Uhrzeit

Die Synchronisationssignale Uhrzeit bzw. Grundtakt können unabhängig voneinander nur von jeweils einer CP52A0 gesendet werden. Sie können auf einer CP52A0 entweder gesendet oder empfangen werden. Die Festlegung zum Senden oder Empfangen (Grundtakt) bzw. nur zum Senden (Uhrzeit) erfolgt in HWKonfig.

Wurde die Uhrzeit bzw. der Grundtakt auf mehreren Zugriffsbaugruppen zum Senden projektiert, wird das Signal immer von der CP52A0 gesendet, die mit einer CP52IO verbunden ist, deren LWL-Schnittstelle sich im Baugruppenträger am weitesten links oben befindet.

Die Signale SYSFAIL\* und Bus-Alarm können von jeder CP52A0 gleichzeitig gesendet und empfangen werden.

### 3.3.4 Projektierung

#### Regeln

1. Die gesamte Anwenderprojektierung erfolgt in den Koppel-Baugruppenträgern
2. Die Namen der Koppel-Baugruppenträger (UR5213) müssen mit einem **eindeutigen** Namen (Register „Allgemein“ des Objekteigenschaften-Dialogs) belegt werden.
3. Im HWKonfig ist als Kommunikationsbaugruppe pro Koppel-Baugruppenträger eine GDM-Zugriffsbaugruppe CP52A0 auf einem beliebigen Steckplatz zu projektieren.
4. Pro Koppel-Baugruppenträger darf nur **eine** GDM-Zugriffsbaugruppe CP52A0 projektiert werden.
5. Pro CP52A0 muss ein Zentralbaustein @SRACK projektiert werden.

#### 3.3.4.1 Projektierung in HWKonfig

In HWKonfig wird die Hardware-Konfiguration eines Baugruppenträgers definiert. Die Baugruppe CP52A0 befindet sich im Baugruppenkatalog von SIMATIC TDC unter Kommunikationsbaugruppen.

Auf der Registerkarte „Synchronisierung“ wird angegeben, ob die Signale Grundtakt, Bus-Alarm und Uhrzeit gesendet, empfangen oder inaktiv geschaltet werden sollen:

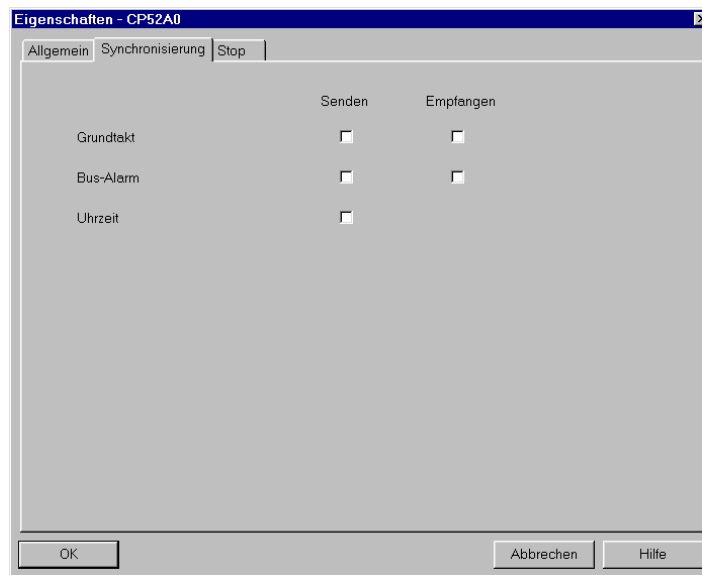


Bild 3-11 Registerkarte „Synchronisierung“

Für das Signal **Grundtakt** gibt es folgende Zustände:

1. Signal empfangen **oder**
2. Signal senden **oder**
3. inaktiv

Für das Signal **Bus-Alarm** gibt es folgende Zustände:

1. Signal empfangen **und / oder**
2. Signal senden **oder**
3. inaktiv

Für das Signal **Uhrzeit** gibt es folgende Zustände:

1. Signal senden **oder**
2. Inaktiv

---

#### HINWEIS

Das Empfangen des Uhrzeit-Signals wird am Baustein RTCM eingestellt.

---

Inaktiv geschaltet werden die Signale dann, wenn keine der Schaltflächen angewählt wird.

#### Defaulteinstellung

Als **Defaulteinstellung** sind die Signale **inaktiv** geschaltet (nichts angewählt). Durch Mausklick auf die Schaltfläche gibt der Anwender für jedes der Signale an, ob er es senden, empfangen oder die Grundeinstellung beibehalten möchte.

Auf der Registerkarte „Stop“ kann eingestellt werden, ob die Baugruppe den ganzen Rahmen stoppen soll, wenn das Signal Systemfehler (SYSFAIL\*) von anderen Baugruppenträgern empfangen wird. Gleichzeitig kann noch festgelegt werden, ob dieses Signal gesendet werden soll.

#### HINWEIS

Solange ein Koppel-Baugruppenträger mit „H“ blinkt (SYSFAIL aktiv) und dieser Rahmen aufgrund der Projektierung das SYSFAIL-Signal sendet, laufen auch die anderen Koppel-Baugruppenträger nicht mehr hoch. In diesem Fall muss der Master-Baugruppenträger ausgeschaltet und die Koppel-Baugruppenträger, welche das SYSFAIL-Signal verursachen, zurückgesetzt werden.

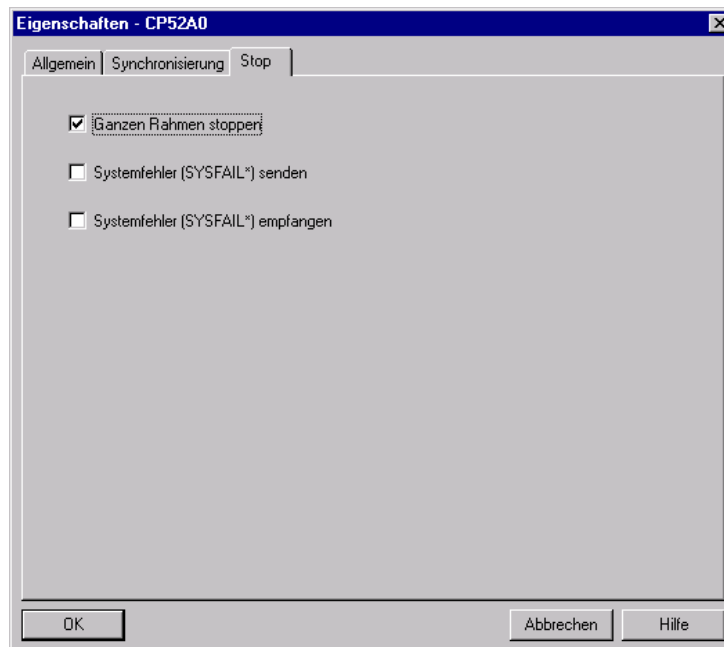


Bild 3-12 Registerkarte „Stop“

### 3.3.4.2 Projektierung in CFC

Für die Baugruppe CP52A0 des Koppel-Baugruppenträgers muss der Zentralbaustein @SRACK projektiert werden.

Hierbei muss auf jeden Fall der Anschluss „CTS“ (CP52A0-Baugruppenname) projektiert werden.

An den Anschlüssen „Nxx“ (Name Baugruppenträger xx) kann von N01 beginnend (und weiter lückenlos aufsteigend mit N02 etc.), angegeben werden, welche Baugruppenträger überwacht werden sollen. Der Baustein wertet die Anschlussbelegung bis zum ersten Anschluss mit einem Leerstring aus.

Der Datenaustausch wird über Sende- und Empfangsbausteine projektiert. An den AT- und AR-Eingängen ist bei der Baugruppenträgerkopplung nur der Kanalname anzugeben. Angaben für Adressstufe 1 und 2 sind nicht zu projektieren. Sender und Empfänger mit gleichen Kanalnamen kommunizieren miteinander.

### 3.3.5 Leistungsdaten

#### 3.3.5.1 Übertragungsraten

Die Datenübertragungsrate zwischen den einzelnen Baugruppenträgern ist stark konfigurationsabhängig.

Bei einer Konfiguration von ca. 20 Baugruppenträgern mit jeweils 4 CPUs kann von einem Datendurchsatz von **60 Bytes/ms** pro CPU für Senden und Empfangen ausgegangen werden.

#### 3.3.5.2 Leitungslängen

Die Performance der Baugruppenträger-Kopplung hängt mit der Länge der verwendeten Lichtwellenleiter zusammen. Es wird deswegen die Verwendung möglichst kurzer Leitungslängen empfohlen. Größere Leitungslängen verlangsamen den Speicherzugriff und führen einerseits zu einer höheren Rechenbelastung der CPU und verringern andererseits die maximal mögliche Datenübertragungsrate.

Die maximale Leitungslänge für die Verbindung zwischen CP52A0 und CP52IO beträgt 200 m.

#### 3.3.5.3 Schnittstellenbelegung

Der letzten belegten Schnittstelle jeder CP52IO steht ein etwas größeres Zeitfenster für den GDM-Zugriff zur Verfügung. Dies führt, bei gleichem Datendurchsatz, zu einer niedrigeren Rechenzeitbelastung der CPU550.

#### Beispiel

Sieben Baugruppenträger sind über zwei CP52IO an den GDM-Baugruppenträger angeschlossen.

Auf der ersten CP52IO sind die Schnittstellen X1 bis **X4**, auf der zweiten Baugruppe die Schnittstellen X1 bis **X3** belegt.

Somit erzeugen die Schnittstellen **X4** (erste CP52IO) und **X3** (zweite CP52IO) eine geringere CPU-Rechenbelastung auf den angeschlossenen Baugruppenträgern.

### 3.4 Baugruppenträger-Kopplung 7 D) ' A \$

#### Allgemeines

Mit der Baugruppe CP53M0 können folgende Kopplungen realisiert werden:

Anschluss eines SIMATIC TDC- Baugruppenträgers an ein SIMADYN D-System mit:

- SIMADYN D als Master:

An jede beliebige Stelle kann anstatt einer CS22 eine CP53M0 an die CS12/CS13/CS14 angeschlossen werden. Dabei ist die CP53M0 im Slavemodus zu parametrieren.

- SIMATIC TDC als Master:

In einem SIMATIC TDC-Baugruppenträger ist eine CP53M0 im Mastermodus parametrierbar. An die beiden Lichtwellenleiterschnittstellen kann jeweils eine CS22 oder eine CP53M0 im Slavemodus angeschlossen werden.

Somit ist es möglich, neben der Kopplung zu SIMADYN D auch bis zu drei SIMATIC TDC- Baugruppenträger miteinander zu koppeln. Hierzu wird in einem Baugruppenträger die CP53M0 im Mastermodus parametrierbar und in den beiden anderen Baugruppenträgern der Slavemodus der CP53M0 genutzt.

Im Folgenden wird als Master-Baugruppenträger der Baugruppenträger bezeichnet, in dem die CP53M0-Baugruppe im Mastermodus steckt.

Als Slave-Baugruppenträger wird der Baugruppenträger bezeichnet, in dem die CP53M0-Baugruppe im Slavemodus steckt.

Die Kopplung über eine CP53M0 im Mastermodus oder Slavemodus wird wie bei einer Kopplung auf SIMADYN D-Seite projektiert.

#### Initialisierung und Überwachung

Für die Kopplungs-Initialisierung und Kopplungs-Überwachung sind in jedem Baugruppenträger auf einer beliebigen CPU jeweils ein Zentralbaustein @CS1 (Mastermodus) bzw. @CS2 (Slavemodus) zu projektieren.

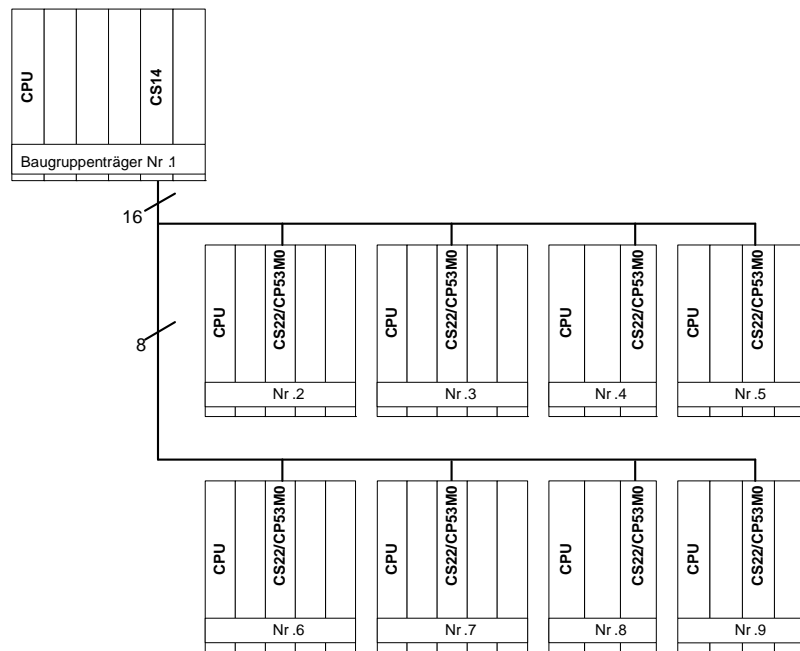


Bild 3-13 Maximalkonfiguration für 8 Slave mit CS14

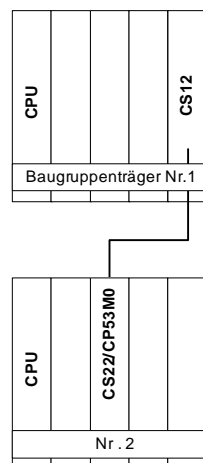


Bild 3-14 Punkt zu Punkt Verbindung mit CS12

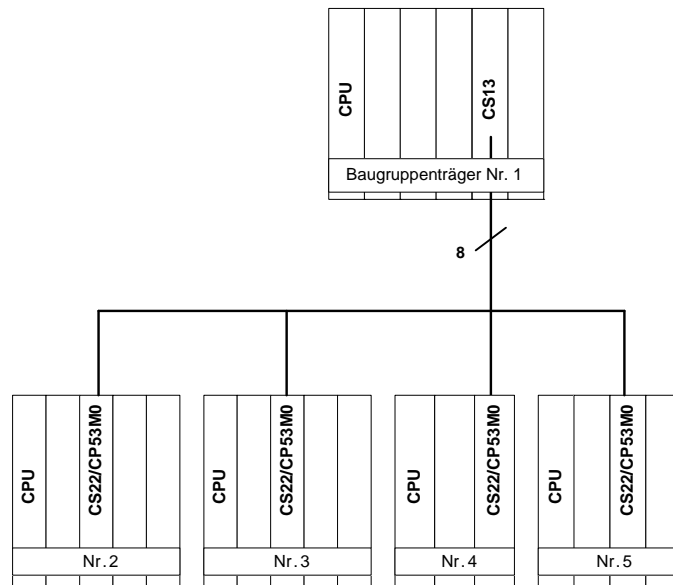


Bild 3-15 Konfiguration für vier Slaves mit CS13

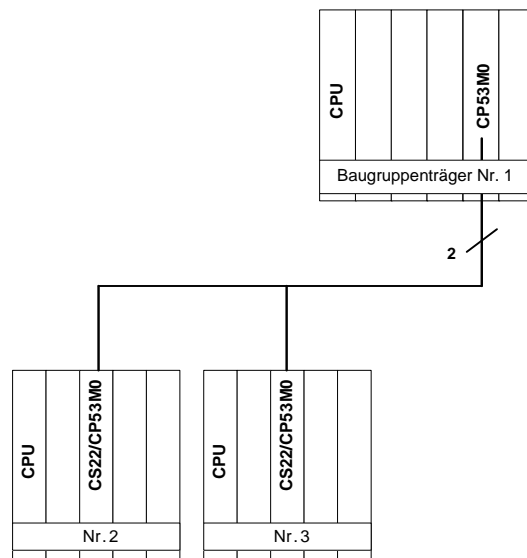


Bild 3-16 Maximalkonfiguration mit CP53M0 als Master



### 3.4.1 Hardware-Aufbau

#### Übersicht

- Es können nur SIMADYN D-Baugruppenträger mit C-Bus-Anschluss an die CP53M0 gekoppelt werden (z.B. SR24).
- Der Master-Baugruppenträger besitzt, abhängig von der Anzahl der anzuschließenden Slaves, eine oder mehrere CS12-, CS13-, CS14 oder CP53M0-Baugruppen. Die Slave-Baugruppenträger besitzen jeweils eine CS22- oder CP53M0-Baugruppe.
- An eine CP53M0 im Mastermodus können maximal 2 Slavebaugruppen (CS22 oder CP53M0 im Slavemodus) gekoppelt werden.
- In einem SIMATIC TDC-Baugruppenträger können mehrere CP53M0 im Slavemodus oder Mastermodus projektiert sein. Es können also mehrere unterschiedliche Baugruppenträgerkopplungen in einem Baugruppenträger projektiert werden.
- Die CP53M0-Baugruppen einer Baugruppenträgerkopplung müssen alle in unterschiedlichen Baugruppenträgern projektiert sein.

### 3.4.2 Leistungsumfang

#### Übersicht

Alle Slave-Baugruppenträger sind fest mit dem Master-Baugruppenträger gekoppelt, weil ein Slave-Baugruppenträger laufend auf den Speicher des Master-Baugruppenträgers zugreift.

- Die Einschaltreihenfolge der Master-/Slave-Baugruppenträger ist beliebig.
- Alle Baugruppenträger können aus dem laufenden Betrieb aus- und wieder zugeschaltet werden.
- Wird ein Slave-Baugruppenträger aus- und wieder zugeschaltet, so ist die Kommunikation zwischen den verbleibenden Teilnehmern davon unberührt.
- Abgeschaltete Slave-Baugruppenträger können umprojektiert und wieder zugeschaltet werden. Auch kann die Anzahl der Sender und Empfänger verändert werden (z.B. falls ein Sender zu wenig projektiert worden ist).
- Sobald der abgeschaltete Slave-Partner wieder zugeschaltet wird, erfolgt ein neuer Verbindungsaufbau zwischen dem neu zugeschalteten Partner und allen anderen Partnern. Dies gilt auch für die Slave-Slave-Kommunikation, wenn also die CS12-, CS13-, CS14- oder CP53M0 (Mastermodus) Baugruppe nur als Datenaustausch-Bereich und nicht als Kommunikationspartner eingesetzt wird. Die Slave-Slave-Kommunikation ist bei abgeschaltetem Master-Baugruppenträger aber unterbrochen.

---

#### HINWEIS

Das Abziehen der Lichtwellenleiter während des laufenden Betriebs ist nicht zulässig und kann zum Absturz der CPU führen.

---

### 3.4.3 Reaktion bei "Abschaltung" eines Koppelpartners

<b>Reaktionen des Master-Baugruppenträgers</b>	<p>Der Master-Baugruppenträger wird abgeschaltet:</p> <p>Der @CS2-Zentralbaustein kann nicht mehr auf den Master-Baugruppenträger zugreifen und bereitet einen Wiederanlauf vor (zusätzlich wird CDM-Bausteinausgang auf "gestört" gesetzt, vgl. @CS2-Maske). Dann wird auf das Zuschalten des Master-Baugruppenträgers gewartet.</p> <p>Alle Slave-Sende-/Empfangsbausteine können nicht mehr auf den Master-Baugruppenträger zugreifen und starten eine neue Kanalanmeldung.</p>
<b>Reaktionen des Slave-Baugruppenträgers</b>	<p>Der Slave-Baugruppenträger wird abgeschaltet:</p> <p>Der @CS1-Zentralbaustein und die maximal sieben weiteren @CS2-Zentralbausteine dekrementieren ihren jeweiligen NCP-Anschluss (d.h. die Anzahl der aktiven Slave-Baugruppenträger ist um eins verringert). Ansonsten erfolgt keine Reaktion und die NCP-Anschlüsse werden nach Wiederanlauf des entsprechenden Slave-Baugruppenträgers wieder inkrementiert.</p> <p>Alle projektierten Send-/Empfangsbausteine, deren Koppelpartner auf dem abgeschalteten Baugruppenträger liegen, warten auf den Wiederanlauf des abgeschalteten Baugruppenträgers.</p>

### 3.4.4 Reaktion bei "Zuschaltung" des Master-Baugruppenträgers

<b>Reaktion</b>	<p>Wird bei laufenden Slave-Baugruppenträgern der Master-Baugruppenträger zugeschaltet, ist kurzzeitig mit einem erhöhten Rechenzeitbedarf für den Verbindungsaufbau bei den über die Baugruppenträgerkopplung kommunizierenden CPUs zu rechnen. Dies kann bei bereits sehr stark ausgelasteten CPUs zur Anzeige eines 'E' (Fehler im Aufgabenverwalter) an der Siebensegmentanzeige führen.</p>
-----------------	--

#### 3.4.4.1 Quittierung

	<p>Die Quittierung des 'E' kann auf zwei Arten erfolgen:</p>
<b>Quittierung von Hand</b>	<p>Nach dem Verbindungsaufbau kann das 'E' bei der betroffenen CPU durch die Betätigung des Quittiertasters quittiert werden.</p>
<b>Automatische Quittierung</b>	<p>Für die automatische Quittierung ist die folgende Projektierung auf allen über die Baugruppenträgerkopplung kommunizierenden CPUs im Slave-Baugruppenträger zu projektieren. Die automatische Quittierung kann mit Hilfe dieser Projektierung auf zwei verschiedene Arten realisiert werden:</p> <ol style="list-style-type: none"><li>1. Alle YEV-Ausgänge der über die Baugruppenträgerkopplung kommunizierenden Funktionsbausteine werden mit Hilfe einer Projektierung (siehe Bild 3-3) überwacht. Ist der Wert aller YEV-Ausgänge kleiner 9 (d.h. Initialisierung abgeschlossen) wird der Eingang NOT.I auf '1' gesetzt. Mit Hilfe des CDM-Ausgangs des @CS2 Zentralbausteins wird sichergestellt, dass eine automatische Quittierung nur dann erfolgt, wenn der Master-Baugruppenträger auch tatsächlich eingeschaltet ist. Mit Hilfe der Zeitbegrenzung (Eingang T</li></ol>

am PCL) wird die Zeitdauer begrenzt, innerhalb derer die automatische Quittierung erfolgen soll. Mit Hilfe des SYF4-Funktionsbausteins erfolgt nun eine automatische Quittierung des an der Siebensegmentanzeige anstehenden 'E'.

2. Falls nicht alle YEV-Ausgänge überwacht werden können oder sollen, ist der Eingang OR.I2 auf den Wert "1" zu setzen und der Eingang NOT.I nicht zu beschalten. In diesem Fall erfolgt die automatische Quittierung innerhalb der am Anschluss T am PCL eingestellten Zeit nach Einschalten des Master-Baugruppenträgers (Ausgang @CS2.CDM).

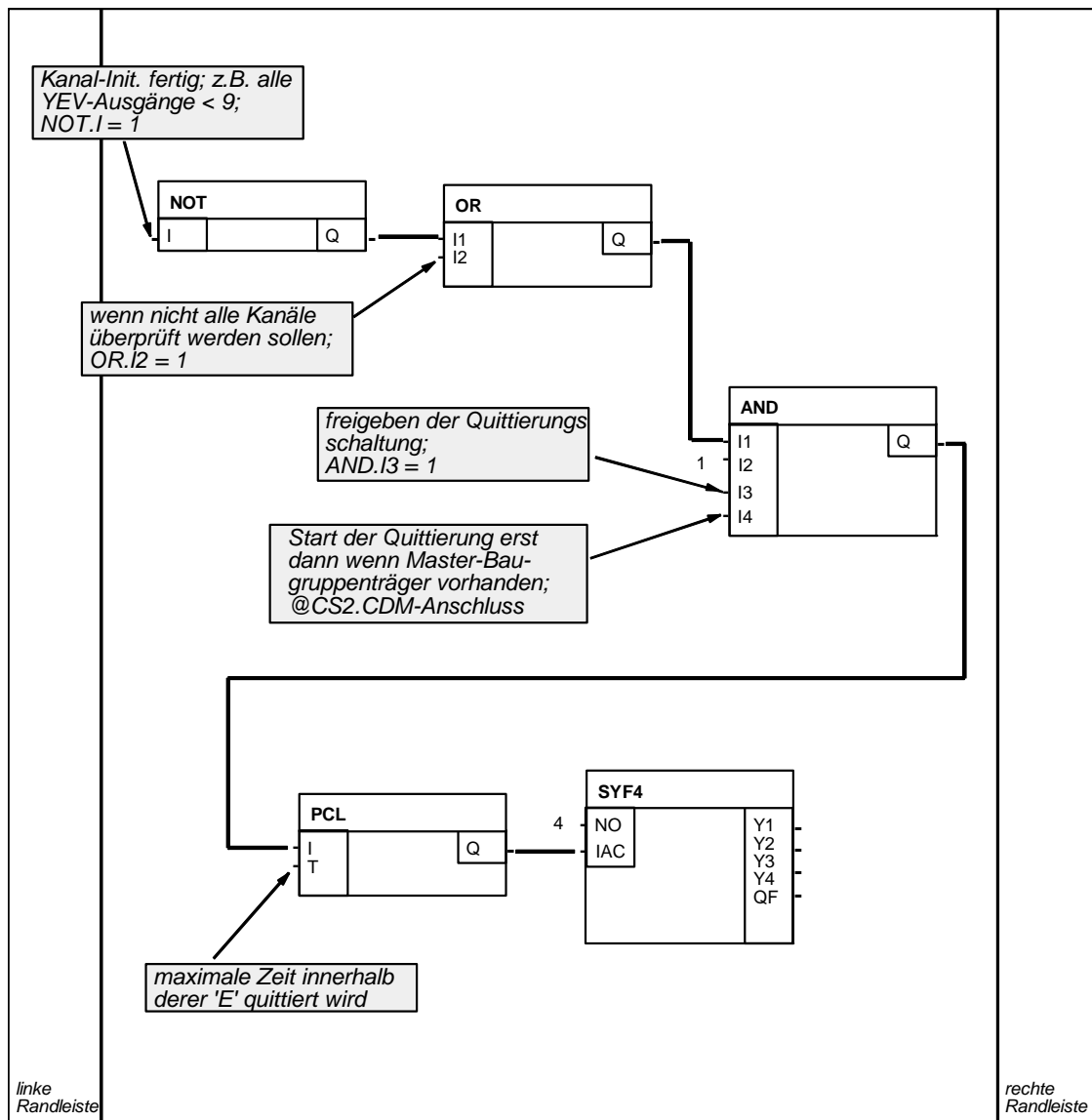


Bild 3-17 Automatische Quittierung des 'E'

### 3.4.5 Wiederanlauffähigkeit

#### Synchronisieren von Sender und Empfänger

Ein weiteres wichtiges Kommunikationsmerkmal bei externen Kommunikationsschnittstellen ist die Wiederanlauffähigkeit von Sendern/Empfängern. Sender/Empfänger finden immer ihren alten Kanal wieder und synchronisieren sich wieder darauf.

Baugruppenträger können in beliebiger Reihenfolge an- und wieder abgeschaltet werden. Die Sender/Empfänger der Baugruppenträger, in denen jeweils die CP53M0 (Slavemodus) steckt, synchronisieren sich bei jedem Wiederanlauf auf den alten Kanal.

Wenn ein Sender/Empfänger bei der Anmeldung einen "passenden" Kanal identifiziert, dann kann er nicht feststellen,

- ob dieser Kanal schon einmal "früher" von ihm benutzt wurde oder
- ob dieser Kanal vielleicht von einem ganz anderen Sender/Empfänger (ebenfalls Sender oder Empfänger) gerade benutzt wird.

### 3.4.6 Projektierung

#### Regeln

- Bei einer Baugruppenträger-Kopplung über Lichtwellenleiter müssen alle CP53M0-Baugruppen (Slavemodus) unterschiedliche Namen haben. Sind Namen doppelt vergeben, so melden sich die entsprechenden Zentralbausteine mit Mehrfachprojektierung ab (FB-Abschaltung).
- Alle CP53M0-Baugruppen einer Baugruppenträgerkopplung müssen in unterschiedlichen Baugruppenträgern stecken.
- Der für Kopplungs-Zentralbausteine gültige Abtastzeitbereich  $32 \text{ ms} \leq T_A \leq 256 \text{ ms}$  gilt auch für die Baugruppenträgerkopplungs-Zentralbausteine @CS1 und @CS2. Zusätzlich ist zu beachten, dass die @CS2-Zentralbausteine in einer schnelleren Abtastzeit als der @CS1-Zentralbaustein projektiert werden müssen (bei Grundtakt-synchronisation sind auch gleiche Abtastzeiten zulässig). Entscheidend ist die tatsächliche Abtastzeit (in Millisekunden) und nicht die zyklische Task (T1, T2 etc.)

#### Datenschnittstelle

Die Datenschnittstelle befindet sich auf dem Dual-Port-RAM der CP53M0-Baugruppe (Mastermodus). Der Datenaustauschbereich ist 128 kByte groß.

#### Initialisierung und Überwachung

Die Kopplungs-Initialisierung und -Überwachung wird von den Zentralbausteinen @CS1 und @CS2 im Betriebszustand RUN durchgeführt. Die Kopplung ist also nicht mit Beginn des zyklischen Betriebs für alle Sende-/Empfangsbausteine freigegeben, sondern verzögert sich um mehrere Abtastzyklen. Die Kopplung wird grundsätzlich zuerst im Master-Baugruppenträger und anschließend in den Slave-Baugruppenträgern freigegeben.

Nach Freigabe der Kopplung überwachen die Zentralbausteine @CS1 und @CS2 die Kopplung. Dabei wird an den Bausteinausgängen der Zentralbausteine die Anzahl der aktiven Koppelpartner ausgegeben.

**Angaben an den  
AT- und AR-  
Eingängen**

An den AT- und AR-Eingängen der Sende-/Empfangsbausteine ist bei der Baugruppenträgerkopplung nur der Kanalname anzugeben. Angaben für Adressstufe 1 und 2 sind nicht zu projektieren. Sender und Empfänger mit gleichen Kanalnamen kommunizieren miteinander.

**3.4.7 Einschränkungen**

Werden Daten mit einem SIMADYN D Baugruppenträger ausgetauscht, der mit STRUC projektiert ist, sind folgende Punkte zu beachten:

- Es dürfen in der Projektierung bei SIMATIC TDC nur die Bausteine CTV / CRV für den Datenaustausch verwendet werden. Bei Einsatz der Bausteine CTV\_P / CRV\_P (Zeigerbausteine) kommt es zu einem Kommunikationsfehler (,C').
- Es kann nur zu den Datentypen I2, I4 und Real (NF) unter STRUC gekoppelt werden. Bei allen anderen Datentypen kommt es zu einem Kommunikationsfehler (,C'). In diesem Fall kann aber die Typumschaltung in STRUC verwendet werden.

Für die Zugriffszeiten auf die Baugruppe CP53M0 über den Rückwandbus ist folgender Unterschied zwischen Master und Slave zu beachten:

- Master: ca. 1  $\mu$ s (4 Byte)
- Slave: ca. 8  $\mu$ s (4 Byte)

Die Auslastung der CPUs im Slave-Baugruppenträger ist bei gleicher Anzahl an Daten damit entsprechend höher.

## 3.5 Kopplung TCP/IP (CP51M1)

### Einleitung

Anhand dieser Musterprojektierung soll das grundsätzliche Vorgehen eines Projektors bei der Realisierung einer Kopplung eines SIMATIC TDC Baugruppenträgers über TCP/IP bzw. UDP beschrieben werden.

Die hier beschriebene Musterkonfiguration beinhaltet die Grundausstattung an Hardware sowie Funktionsbausteinen und zeigt deren prinzipielle Anwendung. Der Funktionsumfang der Musterprojektierung wurde bewusst sehr klein gehalten, um einen schnellen Einstieg in die Thematik zu schaffen. Erweiterungen der Funktionalität und/oder der HW-Komponenten sind ohne weiteres möglich. Es sind hierbei jedoch die Angaben in der jeweiligen Funktionsbaustein-Dokumentation zu beachten.

Alle verwendeten Bezeichnungen der Projektierung sind willkürlich gewählt und nur für diese Musterkonfiguration verbindlich.

Der Aufbau dieser Projektierungsanleitung gibt den zeitlichen Ablauf der Arbeitsschritte wieder, mit der die gesamte Projektierung erstellt werden kann. Dies ist allerdings nur als Empfehlung zu verstehen und muss nicht zwingend eingehalten werden.

### Anwendungsfälle

Die CP51M1 kann für folgende Anwendungsfälle eingesetzt werden:

- Austausch von Prozessdaten mit anderen CP51M1/CP5100 und SIMATIC Industrial Ethernet-Baugruppen (z.B. CP443-1)
- Visualisierung von Prozessdaten mittels WinCC
- Visualisierung von Meldungen mittels WinCC
- Austausch von Prozessdaten mit Fremdsystemen (z.B. Prozessrechner)
- Zentrale Inbetriebsetzung und Diagnose aller CPU-Baugruppen im Baugruppenträger
- Uhrzeitsynchronisation zur Verwendung einer einheitlichen Uhrzeit innerhalb einer Anlage
- Routing wird ab der FW-Version V1.1 und mit D7-SYS ab V7.0 unterstützt

---

### HINWEIS

Bei der Integration von neuen Baugruppen in ein bestehendes TCP/IP-Netz sind vom Netzwerkadministrator immer die IP-, Subnetz- und Routeradressen sowie die Portnummern für Applikationen zu erfragen.

---

### Literatur zu diesen Themen:

- TCP/IP-Grundlagen (z.B. W. Richard Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley-Verlag)
- CP51M1-Hardwareokumentation

### 3.5.1 TCP/IP und UDP im Vergleich

**TCP/IP** Die Protokollart TCP/IP ist verbindungsorientiert. Die Daten können somit nur dann gesendet werden, wenn der Koppelpartner auch erreichbar ist. Solange eine Verbindung zum Koppelpartner besteht, sorgt das TCP/IP-Protokoll dafür, dass die gesendeten Daten auch beim Koppelpartner ankommen. Im Fehlerfall werden die Daten gegebenenfalls mehrfach übertragen.

Bei einem einseitigen Verbindungsabbau kann protokollbedingt ein **Datenverlust** im TCP/IP-Stack der CP51M1 oder des Koppelpartners (z.B. PC) auftreten.

---

**HINWEIS** Bei wichtigen Informationen muss die Zustellung der Daten auf **Applikationsebene** überwacht werden.

---

**UDP** Die Protokollart UDP ist nicht verbindungsorientiert. Die Daten werden somit auch dann gesendet, wenn der Koppelpartner temporär nicht erreichbar ist und **die Daten gehen somit verloren**. Empfangene Daten sind aber aufgrund der Datensicherungsmechanismen korrekt.

UDP erzielt im Vergleich mit TCP/IP eine höhere Kommunikationsgeschwindigkeit zwischen den Koppelpartnern. Dies geht allerdings einher mit einer eingeschränkten Datenübertragungssicherheit.

**Kanalmodi** Mit den Protokollen TCP/IP bzw. UDP sind folgende Kanalmodi möglich:

- Handshake
- Refresh
- Multiple
- Select

**Typische Anwendungsfälle** Die Kanalmodi **Refresh** und **Multiple** sind überschreibend (d.h. ältere Daten können durch neuere überschrieben werden). Für diese beiden Kanalmodi eignet sich deshalb besonders **UDP** als Übertragungsprotokoll.

Die Kanalmodi **Handshake** und **Select** sind nicht überschreibend; als Übertragungsprotokoll ist deshalb hier **TCP/IP** zu empfehlen.

**Client oder Server** Folgende Tabelle, gibt einen Überblick, wann SIMATIC TDC als TCP/IP-Client bzw. Server arbeitet:

<b>SIMATIC TDC (TCP/IP) Prozessdaten (CRV/CTV)</b>		<b>Kommunikationspartner</b>
Angabe von IP-Adr. und Port in Adressstufe 2	Connect → ← accept Send data →	muss Server sein
Keine Angabe in Adressstufe 2	← Connect accept → ← Send data	muss Client sein

### 3.5.2 Musterkonfiguration

Für eine Kopplung von SIMATIC TDC über TCP/IP werden pro Station mindestens folgende Hardwarekomponenten benötigt:

- Baugruppenträger **UR5213**
- CPU-Baugruppe **CPU551** mit Speichermodul **MC5xx**
- Kommunikationsbaugruppe **CP51M1**

---

**HINWEIS** In einem Baugruppenträger können maximal vier CP51M1 parallel betrieben werden.

---

### 3.5.3 Projektierungsschritte

Im folgenden werden nur die Projektierungsschritte für eine Station gezeigt. Alle weiteren Stationen sind analog zu handhaben.

#### 3.5.3.1 Projektierung im HWKonfig

**Parametrierung der CP51M1** Die Parametrierung der CP51M1 erfolgt über die entsprechenden Register des Objekteigenschaften-Dialogs.

Zur Parametrierung der Ethernet Schnittstelle müssen im Unterregister „Eigenschaften von Ethernet Schnittstelle der CP51M1“ folgende relevante Einstellungen angegeben werden:

- **IP- Adresse (Register „Parameter“)**  
IP-Adresse der Baugruppe in Dot-Notation, hier: **141.20.135.197**
- **Subnetzmaske (Register „Parameter“)**  
Subnetzmaske zur Kennzeichnung des Netzwerkanteils, hier: **255.255.0.0** für ein Klasse B Netz



- **IP-Adresse des Default-Routers (Register „Parameter“)**  
hier: „keinen Router verwenden“

**HINWEIS**

Es ist darauf zu achten, dass IP-Adresse und Subnetzmaske harmonieren. So ist z.B. bei einer Klasse C-IP-Adresse (193.x.y.z) die Klasse B-Subnetzmaske 255.255.0.0 nicht zulässig. Detailliertere Informationen zur Wahl der IP-Adresse und deren Wertebereiche, finden sich in der Onlinehilfe der CP51M1.

Folgende Abbildung zeigt die oben ausgeführten Einstellungen im Register „Parameter“:

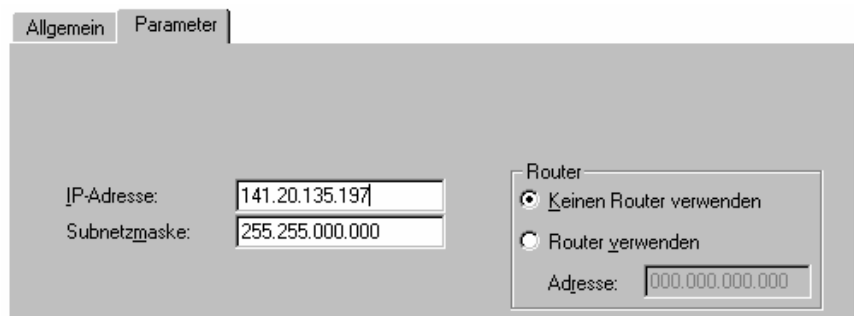


Bild 3-18: Netzwerkeinstellungen der CP51M1

**3.5.3.2 Projektierung mit CFC**

Im folgenden werden nur die wichtigsten Anschlüsse behandelt. Abweichungen, die sich bei der Projektierung einer UDP-Kopplung ergeben, werden an den entsprechenden Stellen beschrieben.

**Benötigte FBs**

Für die Musterkonfiguration werden folgende Funktionsbausteine benötigt:

- Zentralbaustein **@TCPIP** zur Initialisierung und Überwachung der Baugruppe CP51M1 (**wird immer benötigt**)
- Empfangsbaustein **CRV** (hier optional für Prozessdatenkopplung)
- Sendebaustein **CTV** (hier optional für Prozessdatenkopplung)

**3.5.3.2.1 Zentralbaustein @TCPIP**

**Projektierung**

Anschluss	Anschlussbelegung (Bedeutung)
CTS	<b>D1800C.X01</b> (Baugruppenname und Steckerbezeichner der projektierten CP51M1)

### 3.5.3.2.2 Empfangsbaustein CRV

#### Anschluss „AR“

Initialisierungsanschluss für Adressangaben. Hier ist neben dem Kanalnamen die **Adressstufe 1** und die Adressstufe 2 (nur wenn Client) anzugeben.

Die beiden Adressstufen werden in der Notation durch einen „.“ getrennt.

#### Regeln für Adressstufe 1:

- das erste Zeichen (Buchstabe) legt das gewünschte Protokoll („**T**“ = TCP/IP, „**U**“ = UDP) fest (hier „**T**“).
- das zweite Zeichen muss ein Bindestrich „-“ sein.
- durch die nächsten **5 Ziffern** wird die Portnummer des Kanals festgelegt wobei führende Nullen angegeben werden müssen (hier: „**01024**“ für Portnummer 1024).

Als Portnummer sollten in Abstimmung mit dem Systemadministrator nur Ports von 1024 bis 65535 verwendet werden. Die Portnummern bis einschließlich 1023 sind normalerweise für „well known services“ und „Unix-specific services“ reserviert.

#### Regeln für Adressstufe 2:

- die ersten **12 Ziffern** legen die IP-Adresse des remoten Koppelpartners fest. Die Angabe erfolgt in der sogenannten „Dot-Notation“ allerdings ohne Angabe des trennenden Punktes. Führenden Nullen sind anzugeben (hier: „**141020135198**“ für die IP-Adresse 141.20.135.198).
- das 13-te Zeichen muss ein Bindestrich „-“ sein
- durch die nächsten **5 Ziffern** wird die Portnummer des remoten Koppelpartners festgelegt; führende Nullen sind anzugeben (hier: „**01024**“ für die Portnummer 1024).

#### Projektierung

Anschluss	Anschlussbelegung (Bedeutung)
CTS	<b>D1800C.X01</b> (Baugruppenname und Steckerbezeichner der projektierten CP51M1)
AR	<b>RXKAN1.T-01024</b> (Adressparameter Empfangen)
MOD	<b>H</b> (Modus Empfangen)
EN	<b>1</b> (Freigabe)

### 3.5.3.2.3 Sendebaustein CTV

#### Anschluss „AR“

Initialisierungsanschluss für Adressangaben. Hier ist neben dem Kanalnamen die **Adressstufe 1** und die **Adressstufe 2** (nur wenn Client) anzugeben.

Die beiden Adressstufen werden in der Notation durch einen „.“ getrennt.

**Regeln für Adressstufe 1:**

- das erste Zeichen (Buchstabe) legt das gewünschte Protokoll („**T**“ = TCP/IP, „**U**“ = UDP) fest (hier „**T**“).
- das zweite Zeichen muss ein Bindestrich „-“ sein.
- durch die nächsten **5 Ziffern** wird die Portnummer des Kanals festgelegt wobei führende Nullen angegeben werden müssen (hier: „**01024**“ für Portnummer 1024).

Als Portnummer sollten nur Ports von 1024 bis 65535 verwendet werden. Die Portnummern bis einschließlich 1023 sind normalerweise für „well known services“ und „Unix-specific services“ reserviert.

**Regeln für Adressstufe 2:**

- die ersten **12 Ziffern** legen die IP-Adresse des remoten Koppelpartners fest. Die Angabe erfolgt in der sogenannten „Dot-Notation“ allerdings ohne Angabe des trennenden Punktes. Führende Nullen sind anzugeben (hier: „**141020135198**“ für die IP-Adresse 141.20.135.198).
- das 13-te Zeichen muss ein Bindestrich „-“ sein
- durch die nächsten **5 Ziffern** wird die Portnummer des remoten Koppelpartners festgelegt; führende Nullen sind anzugeben (hier: „**01024**“ für die Portnummer 1024).

**Projektierung**

Anschluss	Anschlussbelegung (Bedeutung)
CTS	<b>D1800C.X01</b> (Baugruppenname und Steckerbezeichner der projektierten CP51M1)
AR	<b>TXKAN1.T-01024.141020135198-01024</b> (Adressparameter Senden)
MOD	<b>H</b> (Modus Senden)
EN	<b>1</b> (Freigabe)

**3.5.4 Anwendungs-Hinweise**

**3.5.4.1 Kanalanzahl**

Es können maximal **128 Kanäle** auf der TCP/IP-Baugruppe durch Sende- und Empfangsbausteine (z.B. CTV und CRV) eingerichtet werden.

Die tatsächlich mögliche Anzahl der Kanäle hängt einerseits von der Größe andererseits von der Anzahl der Nutzdaten und vom Zugriffsmechanismus (Handshake, Refresh) ab. Für die Datenschnittstelle stehen 254 KByte RAM auf der CP51M1 zur Verfügung.

**Berechnung**

Die Anzahl der Kanäle kann grob nach folgender Formel berechnet werden:

- **pro Refresh/Multiple-Kanal:** 150 Byte + 2 \* Anzahl der Nutzdatenbytes (Größe der virtuellen Verbindung)
- **pro Handshake/Select-Kanal:** 150 Byte + 1 \* Anzahl der Nutzdatenbytes

### 3.5.4.2 Telegrammlänge

Die Telegrammlänge ist folgendermaßen beschränkt:

- für den Empfangsbetrieb auf 32767 Bytes
- für den Sendebetrieb auf 32767 Bytes

### 3.5.4.3 „Ping“ auf CP51M1

Die Kommunikation mit der CP51M1 kann mit einem „Ping“ geprüft werden. Die Baugruppe antwortet auf einen „Ping“ nur dann, wenn der Kommunikationspartner sich innerhalb des eigenen Netzes befindet oder ein Default-Router (innerhalb des eigenen Netzes) die Verbindung herstellen kann.

### 3.5.4.4 Performance

Die Performance der TCP/IP-Kopplung ist abhängig von der verwendeten Konfiguration. Die folgenden Angaben zeigen die Performancewerte für eine typische Konfiguration (stellen nicht unbedingt die Maximalwerte dar).

#### Voraussetzung

- Bei den Telegrammlängen 192 und 1024 Bytes wurden je CP51M1 32 Sende- und 32 Empfangs-UDP-Verbindungen (Refresh) projektiert.
- Bei der Telegrammlänge 4096 Bytes wurden 8 Sende- und 8 Empfangsverbindungen projektiert.

#### Ergebnisse

Die Anzahl der Sende-/Empfangsaufträge pro CP51M1 beträgt ca. **1270**.

Der maximale Datendurchsatz beträgt ca. 1 MByte/s (Telegrammlänge von 1024 Bytes), der auch trotz größerer Kanallängen nicht mehr gesteigert werden kann.

Datendurchsatz [kBytes/s]	Aufträge/s	Sende-/Empfangszyklus [ms] / Verbindungen	Telegrammlänge [Bytes]
302,260	1270	64 / 40	192
974,556	911	64 / 56	1024
888,16	208	64 / 16	4096

## 3.5.5 Kommunikation mit WinCC

Es gibt zwei Möglichkeiten der Kommunikation zu WinCC:

- Standardkopplung (ohne Zusatzsoftware in WinCC, nur Zugriff Prozessdaten möglich)

- Kopplung über TDC PMC TCP Kanal-DLL (zusätzliche Kanal-DLL für WinCC, Zugriff auf Prozessdaten und Meldungen)

### 3.5.5.1 Standardkopplung

Bei der Projektierung kann dabei auf TDC-Seite wie in Kapitel für MPI beschrieben vorgegangen werden. Der Unterschied besteht darin, dass auf WinCC-Seite statt einer MPI-Verbindung eine Industrial Ethernet Verbindung angelegt werden muss.

---

**HINWEIS** Der Zugriff auf Anschlüsse im System darf nur über Datenbausteinadressierung erfolgen. Zugriff über Merker ist nicht zulässig und führt in WinCC zu einer Fehlermeldung!

---

### 3.5.5.2 Kopplung über TDC PMC Kanal-DLL

Bei der Projektierung ist wie in Kapitel 3.16 beschrieben vorzugehen.

Es wird mit D7-SYS auch ein Beispielprojekt für die CP5100 („D7-TDC-WinCC“) mit entsprechender Dokumentation („D7-SYS – SIMATIC TDC WinCC Anbindung ) mitgeliefert, das für die CP51M1 angepasst werden muss (gemäß 3.5.8).

### 3.5.6 Zentraler Service

In Kapitel 3.13 ist beschrieben, wie vorzugehen ist, um zentralen Service (z.B. CFC Online) über die CP51M1 zu ermöglichen.

### 3.5.7 Uhrzeitsynchronisation

Das Thema Uhrzeitsynchronisation wird in Kapitel 3.14 beschrieben.

### 3.5.8 Umstieg von CP5100 auf CP51M1

Wird eine Projektierung mit CP5100 auf die neue Baugruppe CP51M1 umgestellt, so ist folgende Vorgehensweise einzuhalten:

- In HWKonfig CP5100 löschen und CP51M1 projektieren (dabei darauf achten, dass die gleichen Daten wie bei der CP5100 angegeben werden, z.B. IP-Adresse).
- In den CFC-Plänen muss der Name der CP5100 an allen CTS-Anschlüssen von CTV-/CRV-Bausteinen durch den Baugruppennamen der CP51M1 und den Steckerbezeichner ersetzt werden (z.B. D1800C durch D1800C.X01). Der Anwender wird dadurch unterstützt, indem pro CPU ein beliebiger Plan geöffnet und Extras --> CTS-Anschluss konvertieren... (engl.: Convert CTS connection ...) aufgerufen wird. In dem aufgeblendeten Dialog müssen nur die beiden Namen angegeben werden (an CTS-Anschlüssen vorhandener und neu einzutragender Name). Sind die Namen korrekt

eingegeben und können sie ersetzt werden, wird der Austausch im gesamten Plancontainer vorgenommen. Im Fehlerfall wird eine entsprechende Fehlermeldung ausgegeben.

- Händisch muss dann noch eingegriffen werden, wenn UDP-Telegramme mit einer Länge > 2048 Byte projiziert waren. Die CP51M1 kann bei UDP nur 2048 Byte übertragen!

### 3.6 Kopplung TCP/IP (CP5100)

**Einleitung** Anhand dieser Musterprojektierung soll das grundsätzliche Vorgehen eines Projektors bei der Realisierung einer Kopplung eines SIMATIC TDC Baugruppenträgers über TCP/IP bzw. UDP beschrieben werden.

Die hier beschriebene Musterkonfiguration beinhaltet die Grundausstattung an Hardware sowie Funktionsbausteinen und zeigt deren prinzipielle Anwendung. Der Funktionsumfang der Musterprojektierung wurde bewusst sehr klein gehalten, um einen schnellen Einstieg in die Thematik zu schaffen. Erweiterungen der Funktionalität und/oder der HW-Komponenten sind ohne weiteres möglich. Es sind hierbei jedoch die Angaben in der jeweiligen Funktionsbaustein-Dokumentation zu beachten.

Alle verwendeten Bezeichnungen der Projektierung sind willkürlich gewählt und nur für diese Musterkonfiguration verbindlich.

Der Aufbau dieser Projektierungsanleitung gibt den zeitlichen Ablauf der Arbeitsschritte wieder, mit der die gesamte Projektierung erstellt werden kann. Dies ist allerdings nur als Empfehlung zu verstehen und muss nicht zwingend eingehalten werden.

**Anwendungsfälle** Die CP5100 kann für folgende Anwendungsfälle eingesetzt werden:

- Austausch von Prozessdaten mit anderen CP5100 und SIMATIC Industrial Ethernet-Baugruppen (z.B. CP443-1)
- Visualisierung von Prozessdaten mittels WinCC
- Visualisierung von Meldungen mittels WinCC
- Austausch von Prozessdaten mit Fremdsystemen (z.B. Prozessrechner)

---

**HINWEIS** Bei der Integration von neuen Baugruppen in ein bestehendes TCP/IP-Netz sind vom Netzwerkadministrator immer die IP-, Subnetz- und Routeradressen sowie die Portnummern für Applikationen zu erfragen.

---

**Literatur zu diesen Themen:**

- TCP/IP-Grundlagen (z.B. W. Richard Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley-Verlag)
- CP5100-Hardwareokumentation

### 3.6.1 TCP/IP und UDP im Vergleich

**TCP/IP** Die Protokollart TCP/IP ist verbindungsorientiert. Die Daten können somit nur dann gesendet werden, wenn der Koppelpartner auch erreichbar ist. Solange eine Verbindung zum Koppelpartner besteht, sorgt das TCP/IP-Protokoll dafür, dass die gesendeten Daten auch beim Koppelpartner ankommen. Im Fehlerfall werden die Daten gegebenenfalls mehrfach übertragen.

Bei einem einseitigen Verbindungsabbau kann protokollbedingt ein **Datenverlust** im TCP/IP-Stack der CP5100 oder des Koppelpartners (z.B. PC) auftreten.

---

**HINWEIS** Bei wichtigen Informationen muss die Zustellung der Daten auf **Applikationsebene** überwacht werden.

---

**UDP** Die Protokollart UDP ist nicht verbindungsorientiert. Die Daten werden somit auch dann gesendet, wenn der Koppelpartner temporär nicht erreichbar ist und **die Daten gehen somit verloren**. Empfangene Daten sind aber aufgrund der Datensicherungsmechanismen korrekt.

UDP erzielt im Vergleich mit TCP/IP eine höhere Kommunikationsgeschwindigkeit zwischen den Koppelpartnern. Dies geht allerdings einher mit einer eingeschränkten Datenübertragungssicherheit.

**Kanalmodi** Mit den Protokollen TCP/IP bzw. UDP sind folgende Kanalmodi möglich:

- Handshake
- Refresh
- Multiple
- Select

**Typische Anwendungsfälle** Die Kanalmodi **Refresh** und **Multiple** sind überschreibend (d.h. ältere Daten können durch neuere überschrieben werden). Für diese beiden Kanalmodi eignet sich deshalb besonders **UDP** als Übertragungsprotokoll.

Die Kanalmodi **Handshake** und **Select** sind nicht überschreibend; als Übertragungsprotokoll ist deshalb hier **TCP/IP** zu empfehlen.



**Client oder Server**

Folgende Tabelle, gibt einen Überblick, wann SIMATIC TDC als TCP/IP-Client bzw. Server arbeitet:

SIMATIC TDC (TCP/IP) Prozessdaten		Kommunikationspartner
Sendebaustein CTV (Client)	Connect → ← accept Send data →	Empfänger
Empfangsbaustein CRV (Server)	← Connect accept → ← Send data	Sender

**3.6.2 Musterkonfiguration**

Für eine Kopplung von SIMATIC TDC über TCP/IP werden pro Station mindestens folgende Hardwarekomponenten benötigt:

- Baugruppenträger **UR5213**
- CPU-Baugruppe **CPU550** mit Speichermodul **MC5xx**
- Kommunikationsbaugruppe **CP5100** (hier auf Steckplatz 18)

**HINWEIS**

Die CP5100 darf **generell** nur auf den **Steckplätzen 18 bis 21** projektiert werden. Damit können in einem Baugruppenträger maximal vier CP5100 parallel betrieben werden.

Je nach ausgewähltem Steckplatz müssen auf der Baugruppe die BCD-Codierschalter S1 bis S3 (Lage auf der Baugruppe siehe Bild 3-19) nach untenstehender Tabelle eingestellt werden. Für die Musterkonfiguration ergibt sich die Kombination „C-8-0“.

Steckplatz	S1	S2	S3
18	C	8	0
19	C	8	4
20	C	8	8
21	C	8	C

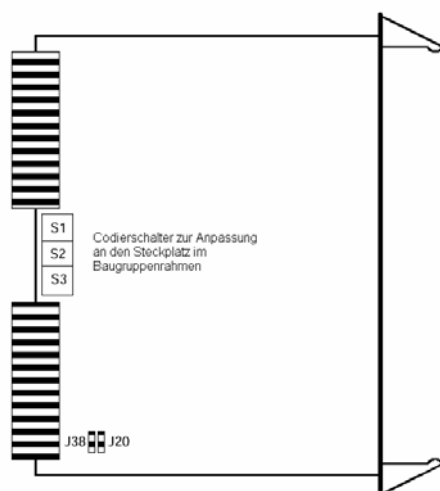


Bild 3-19: Lage der Codierschalter auf der CP5100 (Seitenansicht)

### 3.6.3 Projektierungsschritte

Im folgenden werden nur die Projektierungsschritte für eine Station gezeigt. Alle weiteren Stationen sind analog zu handhaben.

#### 3.6.3.1 Projektierung im HWKonfig

##### Parametrierung der CP5100

Die Parametrierung der CP5100 erfolgt über die entsprechenden Register des Objekteigenschaften-Dialogs.

Zur Parametrierung der Ethernet Schnittstelle müssen im Unterregister „Eigenschaften von Ethernet Schnittstelle der CP5100“ folgende relevante Einstellungen angegeben werden:

- **IP- Adresse (Register „Parameter“)**  
IP-Adresse der Baugruppe in Dot-Notation, hier: **141.20.135.197**
- **Subnetzmaske (Register „Parameter“)**  
Subnetzmaske zur Kennzeichnung des Netzwerkanteils, hier: **255.255.0.0** für ein Klasse B Netz
- **IP-Adresse des Default-Routers (Register „Parameter“)**  
hier: **„keinen Router verwenden“**

---

##### HINWEIS

Es ist darauf zu achten, dass IP-Adresse und Subnetzmaske harmonisieren. So ist z.B. bei einer Klasse C-IP-Adresse (193.x.y.z) die Klasse B-Subnetzmaske 255.255.0.0 nicht zulässig. Detailliertere Informationen zur Wahl der IP-Adresse und deren Wertebereiche, finden sich in der Onlinehilfe der CP5100.

---

Folgende Abbildung zeigt die oben ausgeführten Einstellungen im Register „Parameter“:

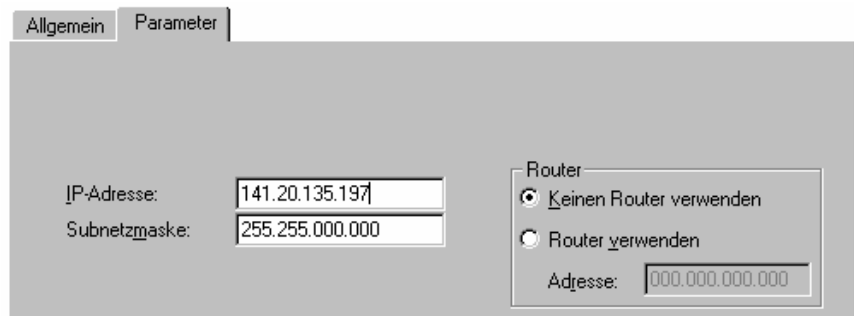


Bild 3-20: Netzwerkeinstellungen der CP5100

### 3.6.3.2 Projektierung mit CFC

Im folgenden werden nur die wichtigsten Anschlüsse behandelt. Abweichungen, die sich bei der Projektierung einer UDP-Kopplung ergeben, werden an den entsprechenden Stellen beschrieben.

#### Benötigte FBs

Für die Musterkonfiguration werden folgende Funktionsbausteine benötigt:

- Zentralbaustein **@TCPIP** zur Initialisierung und Überwachung der Baugruppe CP5100 (**wird immer benötigt**)
- Empfangsbaustein **CRV** (hier optional für Prozessdatenkopplung)
- Sendebaustein **CTV** (hier optional für Prozessdatenkopplung)

#### 3.6.3.2.1 Zentralbaustein @TCPIP

##### Projektierung

Anschluss	Anschlussbelegung (Bedeutung)
CTS	<b>D1800C</b> (Baugruppenname der projektierten CP5100)

##### CFC-Plan

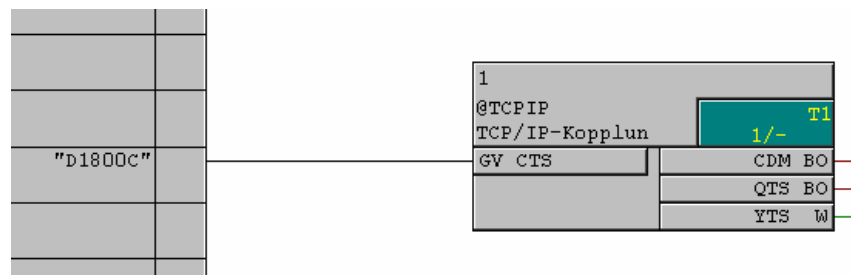


Bild 3-21 : Anschlussbelegung des @TCPIP

#### 3.6.3.2.2 Empfangsbaustein CRV

##### Anschluss „AR“

Initialisierungsanschluss für Adressangaben. Hier ist neben dem Kanalnamen die **Adressstufe 1** anzugeben.

**Regeln für Adressstufe 1:**

- das erste Zeichen (Buchstabe) legt das gewünschte Protokoll („T“ = TCP/IP, „U“ = UDP) fest (hier „T“).
- das zweite Zeichen muss ein Bindestrich „-“ sein.
- durch die nächsten **5 Ziffern** wird die Portnummer des Kanals festgelegt wobei führende Nullen angegeben werden müssen (hier: „01024“ für Portnummer 1024).

Als Portnummer sollten in Abstimmung mit dem Systemadministrator nur Ports von 1024 bis 65535 verwendet werden. Die Portnummern bis einschließlich 1023 sind normalerweise für „well known services“ und „Unix-specific services“ reserviert.

**Projektierung**

Anschluss	Anschlussbelegung (Bedeutung)
CTS	<b>D1800C</b> (Baugruppenname der projektierten CP5100)
AR	<b>RXKAN1.T-01024</b> (Adressparameter Empfangen)
MOD	<b>H</b> (Modus Empfangen)
EN	<b>1</b> (Freigabe)

**CFC-Plan**

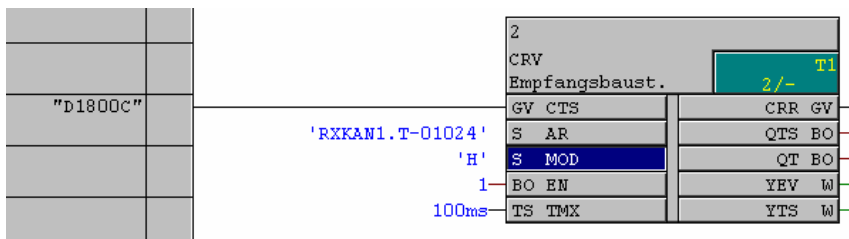


Bild 3-22: Anschlussbelegung des CRV

**3.6.3.2.3 Sendebaustein CTV**

**Anschluss „AR“**

Initialisierungsanschluss für Adressangaben. Hier ist neben dem Kanalnamen die **Adressstufe 1** und die **Adressstufe 2** anzugeben.

Die beiden Adressstufen werden in der Notation durch einen „.“ getrennt.

**Regeln für Adressstufe 1:**

- das erste Zeichen (Buchstabe) legt das gewünschte Protokoll („T“ = TCP/IP, „U“ = UDP) fest (hier „T“).
- das zweite Zeichen muss ein Bindestrich „-“ sein.
- durch die nächsten **5 Ziffern** wird die Portnummer des Kanals festgelegt wobei führende Nullen angegeben werden müssen (hier: „01024“ für Portnummer 1024).

Als Portnummer sollten nur Ports von 1024 bis 65535 verwendet werden. Die Portnummern bis einschließlich 1023 sind normalerweise für „well known services“ und „Unix-specific services“ reserviert.

**Regeln für Adressstufe 2:**

- die ersten **12 Ziffern** legen die IP-Adresse des remoten Koppelpartners fest. Die Angabe erfolgt in der sogenannten „Dot-Notation“ allerdings ohne Angabe des trennenden Punktes. Führenden Nullen sind anzugeben (hier: „**141020135198**“ für die IP-Adresse 141.20.135.198).
- das 13-te Zeichen muss ein Bindestrich „-“ sein
- durch die nächsten **5 Ziffern** wird die Portnummer des remoten Koppelpartners festgelegt; führende Nullen sind anzugeben (hier: „**01024**“ für die Portnummer 1024).

**Projektierung**

Anschluss	Anschlussbelegung (Bedeutung)
CTS	<b>D1800C</b> (Baugruppenname der projektierten CP5100)
AR	<b>TXKAN1.T-01024.141020135198-01024</b> (Adressparameter Senden)
MOD	<b>H</b> (Modus Senden)
EN	<b>1</b> (Freigabe)

**CFC-Plan**

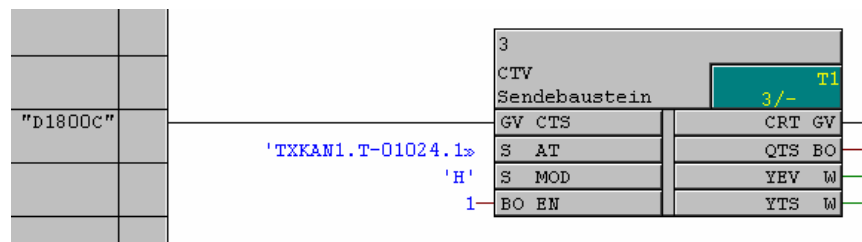


Bild 3-23 Anschlussbelegung des CTV

**3.6.4 Anwendungs-Hinweise**

**3.6.4.1 Kanalanzahl**

Es können maximal **256 Kanäle** auf der TCP/IP-Baugruppe durch Sende- und Empfangsbausteine (z.B. CTV und CRV) eingerichtet werden.

Die tatsächlich mögliche Anzahl der Kanäle hängt einerseits von der Größe andererseits von der Anzahl der Nutzdaten und vom Zugriffsmechanismus (Handshake, Refresh) ab. Für die Datenschnittstelle stehen 254 KByte RAM auf der CP5100 zur Verfügung.

**Berechnung**

Die Anzahl der Kanäle kann grob nach folgender Formel berechnet werden:

- **pro Refresh/Multiple-Kanal:** 150 Byte + 2 \* Anzahl der Nutzdatenbytes (Größe der virtuellen Verbindung)
- **pro Handshake/Select-Kanal:** 150 Byte + 1 \* Anzahl der Nutzdatenbytes

### 3.6.4.2 Telegrammlänge

Die Telegrammlänge ist folgendermaßen beschränkt:

- für den Empfangsbetrieb auf 55759 Bytes
- für den Sendebetrieb auf 65535 Bytes

### 3.6.4.3 „Ping“ auf CP5100

Die Kommunikation mit der CP5100 kann mit einem „Ping“ geprüft werden. Die Baugruppe antwortet auf einen „Ping“ nur dann, wenn der Kommunikationspartner sich innerhalb des eigenen Netzes befindet oder ein Default-Router (innerhalb des eigenen Netzes) die Verbindung herstellen kann.

### 3.6.4.4 Performance

Die Performance der TCP/IP-Kopplung ist abhängig von der verwendeten Konfiguration. Die folgenden Angaben zeigen die Performancewerte für eine typische Konfiguration.

#### Voraussetzung

- Bei den Telegrammlängen 200 und 1000 Bytes wurden je CPU 32 Sende- und 32 Empfangs-UDP-Verbindungen (Refresh) projektiert.
- Bei der Telegrammlänge 4000 Bytes wurden 8 Sende- und 8 Empfangsverbindungen projektiert.

#### Ergebnisse

Die Anzahl der Sende-/Empfangsaufträge pro CP5100 beträgt ca. **1600**.

Der maximale Datendurchsatz beträgt ca. 1,2 MByte/s (Telegrammlänge von 1000 Bytes), der auch trotz größerer Kanallängen nicht mehr gesteigert werden kann.

Datendurchsatz [kBytes/s]	Aufträge/s	Sende-/Empfangszyklus [ms] / Verbindungen	Telegrammlänge [Bytes]
320,000	1600	64 / 40	200
1142,857	1143	64 / 56	1000
1000,000	250	64 / 16	4000

## 3.7 Kopplung PROFIBUS DP (CP50M1)

### 3.7.1 Allgemeine Grundlagen

#### Eigenschaften

Die CP50M1 hat am PROFIBUS DP folgende Eigenschaften:

- **Master / Slave**  
Jede der beiden Schnittstellen der CP50M1 kann sowohl als Master (alleine oder mit anderen Mastern im Multi-Master-Betrieb) als auch als Slave am PROFIBUS DP betrieben werden. Dies kann für jede Schnittstelle unabhängig voneinander erfolgen.
- **Shared Input**  
Jeder Slave am PROFIBUS DP ist genau einem Master zugeordnet (dem parametrierenden Master) und kann zunächst nur mit diesem kommunizieren. Mit dem "Shared Input" ist es weiteren Mastern möglich, die Eingangsdaten des Slaves zu lesen. Die Schnittstellen der CP50M1 unterstützen diese Funktionalität als Master.
- **Routing**  
Die Funktion Routing wird von der CP50M1 ab der FW-Version V1.1 und mit D7-SYS ab V7.0 unterstützt.
- **SYNC und FREEZE**  
Mit den Diensten SYNC und FREEZE ist ein synchrones Schreiben/Lesen von Ausgängen/Eingängen mehrerer Slaves möglich. Die CP50M1 unterstützt diese Dienste als Master.
- **Äquidistanz**  
Äquidistanz ist die Eigenschaft des PROFIBUS DP, die exakt gleich lange Buszyklen gewährleistet.
- **Querverkehr**  
Die konfigurierten Slaves können ohne Projektierung in der CP50M1 "direkt" miteinander Daten austauschen.
- **Datenlängen**  
Es sind maximal 244 Byte je Richtung und Slave übertragbar.
- **Konsistenz**  
Die Daten innerhalb eines Telegramms sind immer konsistent.

---

#### HINWEIS

In einem Baugruppenträger können maximal sechs CP50M1 parallel betrieben werden.

---

## 3.7.2 Projektierung

### 3.7.2.1 Konfigurieren des DP-Systems auf CP50M1

Die Konfiguration des DP-Systems bei der CP50M1 erfolgt wie bei der SIMATIC mit HWKonfig und der Netzprojektierung. Es gibt hier keine Unterschiede zur Konfiguration anderer DP-Master (z.B. CPU 315-2DP).

Das Vorgehen wird im Handbuch "Hardware konfigurieren und Verbindungen projektieren mit STEP 7 in den Kapiteln 3, "Konfigurieren der Dezentralen Peripherie (DP)" und 11, "Vernetzen von Stationen" genau beschrieben.

Aus diesem Grunde wird im folgenden nur auf Besonderheiten der CP50M1 eingegangen.

### 3.7.2.2 Projektierung der Kommunikation in CFC

<b>Funktionsbausteine</b>	<p>Für eine Kopplung PROFIBUS DP müssen folgende Funktionsbausteine projektiert werden:</p> <ul style="list-style-type: none"><li>• ein Kopplungs-Zentralbaustein @PRODP</li><li>• maximal ein Sender- und Empfänger-Funktionsbaustein pro Slave-Station</li><li>• maximal ein Synchronisier-Funktionsbaustein SYNPRO kann projektiert werden</li><li>• maximal ein Diagnose-Funktionsbaustein DPDIAG und ein Slave-Diagnosebaustein DPSLDG pro Slave dürfen projektiert werden</li></ul>
<b>Kommunikations-Dienst</b>	<p>Erlaubte Kommunikations-Dienste sind:</p> <ul style="list-style-type: none"><li>• Prozessdaten</li><li>• Parameterbearbeitung von drehzahlveränderbaren Antrieben</li></ul>
<b>Übertragungsmodus</b>	<p>Erlaubter Übertragungsmodus:</p> <ul style="list-style-type: none"><li>• Refresh bei Empfängern wahlweise auch Multiple</li></ul>
<b>Kopplungs-Zentralbaustein</b>	<p>Der Kopplungszentralbaustein @PRODP initialisiert und überwacht die PROFIBUS DP Kopplung über die Stecker X1 und X2 der CP50M1.</p>



## Angaben am Adressanschluss AT, AR

Besonderheiten der Angaben am Adressanschluss AT, AR der Sende- / Empfangsbausteine bei Verwendung von PROFIBUS DP:

Eingabereihenfolge:

**“Kanalname.Adressstufe 1.Adressstufe 2“**

- **Kanalname:**
  - Maximal 8 Zeichen
  - ASCII-Zeichen außer „Punkt“ und @
  - Kanalnamen aller Sende- und Empfangsbausteine, die auf die Schnittstellen X1 und X2 der CP50M1 zugreifen, müssen unterschiedlich sein (Ausnahme bei Übertragungsmodus „Multiple“)
  - Der Kanalname hat keine spezifische Bedeutung für PROFIBUS DP
- Eingabe von „.“ nach Kanalname
- **Adressstufe 1:**
  - Als Adressstufe 1 wird die Slave-PROFIBUS-Adresse angegeben
  - Die Slave-PROFIBUS-Adresse darf je Sende- und Empfangskanal nur einmal vergeben werden
  - Wertebereich: 0, 3 - 123
  - 3...123: adressieren externe Slaves
- Eingabe von „.“ nach Adressstufe 1
- **Adressstufe 2:**
  - Besteht aus maximal 2 Zeichen, wobei das 2. Zeichen bei der CP50M1 ohne Bedeutung ist.
  - **1. Zeichen:** Byte-Ordering
    - “1“: PROFIBUS-StandardEinstellung  
Die Daten werden im "Motorola-Format" (höherwertiges Byte vor niederwertigem Byte) übertragen
    - “0“: Ausnahmeeinstellung  
Die Daten werden im "Intel-Format" (niederwertiges Byte vor höherwertigem Byte) übertragen. Diese Einstellung kann verwendet werden bei Kommunikationspartnern, deren Datenhaltung intern im Intel-Format erfolgt (z.B. SIMATIC TDC)

## Beispiele für Angaben am Adressanschluss

- AT- ‚Sollwert.25.1‘
  - Der Kanal mit Namen **Sollwert** sendet an einen **Slave** mit der PROFIBUS-Adresse **25**.
- AR- ‚RECEIVE.117.0‘

- Der Kanal mit Namen **RECEIVE** empfängt von einem **Slave** mit der PROFIBUS-Adresse **117**. Daten werden ausnahmsweise im **Intel-Format** übertragen.

### 3.7.2.3 Projektierung als Slave

Wie bei der Projektierung als Slave vorzugehen ist, ist in [„SIMATIC TDC CP50M1 as PROFIBUS DP slave“](#) ausführlich anhand eines Beispiels beschrieben.

### 3.7.2.4 Shared Input

Das Vorgehen bei der Projektierung ist in [„SIMATIC TDC CP50M1, Direct Data Exchange DX“](#) ausführlich anhand eines Beispiels beschrieben.

#### Konfigurations- hinweis

Im Gegensatz zum alten CP50M0 ist es mit dem CP50M1 nicht mehr möglich alle Slaves mitzulesen. Mit dem CP50M1 können nur Slaves mitgelesen werden, die die Funktion „direkter Datenaustausch“ unterstützen. Aus der Hardware Konfiguration kann entnommen werden, welche der Slaves dazu geeignet sind. Falls ein CP50M0 als Parametriermaster verwendet wird, ist es nicht möglich einen CP50M1 als mitlesenden Slave zu parametrieren.

### 3.7.3 Äquidistanz

#### Einführung

Die Projektierung von Äquidistanz am PROFIBUS DP erfolgt für die CP50M1 genauso wie bei einer SIMATIC CPU (siehe dazu das Handbuch „SIMATIC Hardware konfigurieren und Verbindungen projektieren mit STEP 7“, Kap. 3.12 „Einstellen von gleichlangen Buszyklen bei PROFIBUS-Subnetzen“).

### 3.7.4 Kommandos SYNC/FREEZE

#### Allgemeines

Die Kommandos SYNC und FREEZE synchronisieren die Ein- und Ausgänge einer Gruppe von Slaves. Der Funktionsbaustein SYNPRO löst diese Kommandos aus und unterstützt die Konsistenzsicherung.

#### Konsistenz

Für die Sicherstellung der Konsistenz ist der Projektierer verantwortlich. Beim SYNC/FREEZE-Kommando geht es um Konsistenz der Daten über alle beteiligten Slaves. Die Konsistenz der Ein- oder Ausgangsdaten eines Slaves ist selbstverständlich immer gewährleistet.

#### SYNC

Nach Auslösen eines SYNC-Kommandos wartet der DP-Master (CP50M1) eine DP-Busumlaufzeit ab, damit alle Slaves die neuen Ausgangswerte erhalten. Anschließend sendet der DP-Master ein SYNC-Broadcast-Telegramm an die projektierte Slave-Gruppe. Alle Slaves dieser Gruppe aktualisieren daraufhin gleichzeitig ihre zwischengespeicherten Ausgänge.

Die Ausgänge werden erst dann wieder zyklisch aktualisiert, wenn der DP-Master das Steuerkommando UNSYNC sendet (EN=0 am Baustein SYNPRO).

Konsistenzsicherung:

Durch Projektierung ist sicherzustellen, dass während einer DP-Busumlaufzeit nach Auslösen des SYNC-Kommandos die Ausgangsdaten nicht durch SIMATIC TDC-CPU's verändert werden.

#### **FREEZE**

Unmittelbar nach Auslösen eines FREEZE-Kommandos sendet der DP-Master ein FREEZE-Broadcast-Telegramm an die projektierte Slave-Gruppe. Alle Slaves dieser Gruppe lesen daraufhin gleichzeitig ihre Eingänge ein und speichern sie zwischen. Nach Ablauf einer DP-Busumlaufzeit liegen diese Eingangsdaten bei den SIMATIC TDC-CPU's vor.

Die Eingangsdaten werden erst dann wieder zyklisch vom DP-Slave an den DP-Master übertragen, wenn der DP-Master das Steuerkommando UNFREEZE sendet (EN=0 am Baustein SYNPRO).

Konsistenzsicherung:

Durch Projektierung ist sicherzustellen, dass während einer DP-Busumlaufzeit nach Auslösen des FREEZE-Kommandos die Eingangsdaten nicht durch den DP-Master ausgewertet werden.

### **3.7.4.1 SYNC/FREEZE-Projektierungsvarianten**

#### **Allgemeines**

Im folgenden werden Begriffe zur Konsistenzsicherung erklärt und verschiedene SYNC/FREEZE-Projektierungsvarianten dargestellt.

#### **Begriffe**

- **Busumlaufzeit**  
Zyklus, in dem der DP-Master (CP50M1) einmal alle Slaves anspricht. In Multimastersystemen pollen alle Master alle ihre Slaves. Die Busumlaufzeit wird durch Baudrate, Anzahl und Typ der Slaves in STEP 7 projektiert und berechnet.
- **Abtastzeit**  
Zyklus, in dem der Funktionsbaustein SYNPRO und die Sende- und Empfangs-Funktionsbausteine (aufSIMATIC TDC-CPU's) gerechnet werden. Die Abtastzeit wird mit CFC projektiert.

#### **HINWEIS**

Busumlaufzeit und Abtastzeit sind unabhängig voneinander.

- **Synczyklus**  
Synczyklus ist ein ganzzahliges Vielfaches der Abtastzeit, projektiertbar am Eingang CNX des Funktionsbausteins SYNPRO (Synczyklus=CNX x Abtastzeit).  
Ein Synczyklus beginnt immer mit einer Abtastzeit. Ein Sync-Kommando wird durch den Funktionsbaustein SYNPRO immer im Systemmode zu Beginn einer Abtastzeit ausgelöst.

#### **Projektierungs- variante 1**

Projektierungsvariante 1 entspricht den meisten Anwendungsfällen:

- Erzeugen von SYNC-Kommandos
- Die Konsistenz über alle Slaves ist gewährleistet
- Der Synczyklus ist mindestens doppelt so groß wie die Abtastzeit (CNX>1)

- Die Länge der Sendetelegramme (Ausgänge) je Slave darf nicht größer als 32 Byte sein
- Alle Sendebausteine und der Funktionsbaustein SYNPRO sind in derselben Abtastzeit zu projektieren
- Der Funktionsbaustein SYNPRO ist in der zeitlichen Bearbeitungsreihenfolge vor allen Sendebausteinen zu projektieren
- Der Ausgang SOK des Funktionsbausteins SYNPRO ist mit den Enable-Eingängen aller (zur Slave-Gruppe gehörenden) Sendebausteine zu verbinden
- Die Busumlaufzeit muss kürzer sein als der Synczyklus minus 1 x Abtastzeit. Es ist zur Laufzeit zu überprüfen, ob der Ausgang SOK in jedem Synczyklus einmal auf "1" geht, ansonsten ist der Synczyklus zu erhöhen

Beispiel:

- Synczyklus=3 x Abtastzeit
- Busumlaufzeit=2 x Abtastzeit
- Annahme: Funktionsbaustein SYNPRO rechnet in der Mitte der Abtastzeit (vor allen Sendebausteinen)

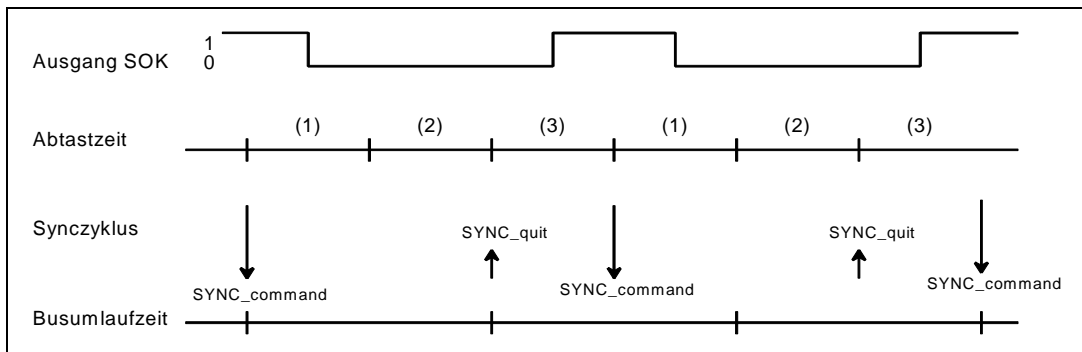


Bild 3-24 Zeitdiagramm SYNC-Variante 1

Nach Auslösen des SYNC-Kommandos sind die Sendebausteine zwei Abtastzeiten (eine Busumlaufzeit) lang gesperrt (SOK=0). In der dritten Abtastzeit nach Auslösen des SYNC-Kommandos werden die Sendebausteine freigegeben (SOK=1).

### Projektierungs- variante 2

Projektierungsvariante 2 hat die höchste SYNC-Performance:

- Erzeugen von SYNC-Kommandos
- Die Konsistenz über alle Slaves ist gewährleistet
- Synczyklus=Abtastzeit (CNX=1)
  - Die Länge der Sendetelegramme (Ausgänge) je Slave darf nicht größer als 32 Byte sein

- Alle Sendebausteine und der Funktionsbaustein SYNPRO sind in derselben Abtastzeit zu projektieren
- Hohe Baudrate (>1,5 Mbaud); mit niedrigeren Baudraten sind die Zeitbedingungen kaum einzuhalten
- Die Busumlaufzeit darf höchstens die Hälfte der Abtastzeit betragen
- Die Busumlaufzeit muss darüber hinaus so klein sein, dass sie von Beginn einer Abtastzeit bis zur Berechnung des Funktionsbausteins SYNPRO verstrichen ist. Diese Voraussetzung kann nicht garantiert werden, sondern diese muss zur Laufzeit überprüft werden

Beispiel:

- Synczyklus=Abtastzeit
- Busumlaufzeit=0,3 x Abtastzeit
- Annahme: Funktionsbaustein SYNPRO rechnet in der Mitte der Abtastzeit (vor allen Sendebausteinen)

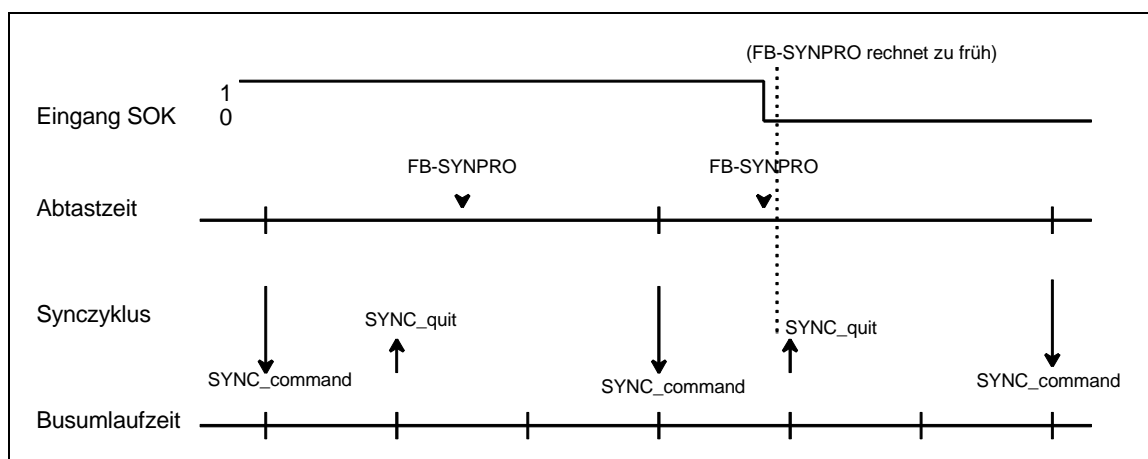


Bild 3-25 Zeitdiagramm SYNC-Variante 2

Im Normalfall sind die Sendebausteine immer freigegeben (SOK=1). Falls aufgrund von Zeitschwankungen der Funktionsbaustein SYNPRO vor Ablauf des SYNC gerechnet wird (rechts im Bild), so werden die Sendedaten nicht aktualisiert, sondern die Werte aus der vorherigen Abtastzeit übertragen. Der Synczyklus und die Konsistenz werden dadurch nicht beeinflusst.

#### Hinweise für gute SYNC-Funktionalität:

Neben einem kleinen Synczyklus ist ein möglichst geringer Jitter (zeitliche Schwankung) im Synczyklus erforderlich. Folgende Vorkehrungen unterstützen dies:

- Unregelmäßiger Datenverkehr auf dem DP-Bus ist zu vermeiden; Single-Master-Betrieb; kein temporäres Zuschalten von Stationen

- Keine Alarm-Tasks auf der gleichen SIMATIC TDC-CPU projektieren. Die Abtastzeit darf keinesfalls überlaufen; dies würde zu einem Ausfall eines SYNC-Kommandos oder zur Verschiebung um eine ganze Abtastzeit führen
- Eine hohe Baudrate und kurze Telegrammlängen projektieren. (In den Jitter geht die Zeit für das Pollen eines Slaves ein.)
- Den Funktionsbaustein SYNPRO und alle zugehörigen Sendebausteine in  $T1=T0$  (Grundabtastzeit) projektieren. Das SYNC-Kommando wird dann mit dem Grundtakt-Interrupt ausgelöst. Der Grundtakt-Interrupt kommt zeitlich genauer als ein im System-Mode ausgelöster Interrupt

### Projektierungs- variante 3

Projektierungsvariante 3 entspricht weniger üblichen Anwendungsfällen des FREEZE:

- Erzeugen von SYNC- und FREEZE- oder nur FREEZE-Kommandos
- Die Konsistenz über alle Slaves ist gewährleistet
- Der Synczyklus ist mindestens dreimal so groß wie die Abtastzeit ( $CNX>1$ ).
  - Die Länge der Sende- oder Empfangstelegramme (Ein- oder Ausgänge) je Slave darf nicht größer als 32 Byte sein
  - Alle Sende- und Empfangsbausteine und der Funktionsbaustein SYNPRO sind in derselben Abtastzeit (auf der selben CPU) zu projektieren
  - Der Funktionsbaustein SYNPRO ist als letzter Funktionsbaustein in der zeitlichen Bearbeitungsreihenfolge zu projektieren
  - Der Ausgang SOK des Funktionsbausteins SYNPRO ist mit den Enable-Eingängen aller (zur Slave-Gruppe gehörenden) Sende- und Empfangsbausteine zu verbinden
- Die Busumlaufzeit muss kürzer sein als der Synczyklus minus  $2 \times$  Abtastzeit. Es ist zur Laufzeit zu überprüfen, ob der Ausgang SOK in jedem Synczyklus einmal auf "1" geht, ansonsten ist der Synczyklus zu erhöhen

Beispiel:

- Synczyklus= $4 \times$  Abtastzeit
- Busumlaufzeit= $2 \times$  Abtastzeit
- Annahme:  
Funktionsbaustein SYNPRO rechnet in der Mitte der Abtastzeit (nach allen Empfangs- und Sendebausteinen)

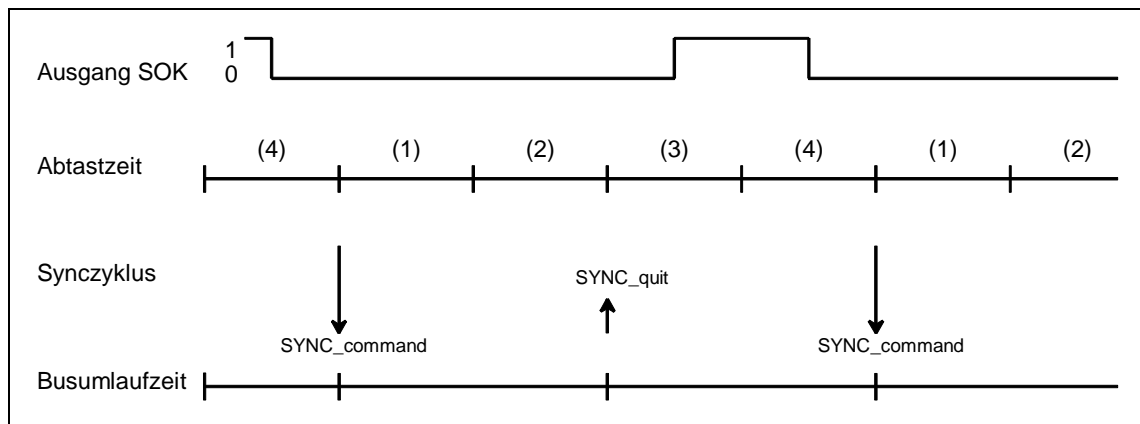


Bild 3-26 Zeitdiagramm SYNC-Variante 3

Nach Auslösen des SYNC-Kommandos sind die Sende- und Empfangsbausteine drei Abtastzeiten (eine Busumlaufzeit und eine Abtastzeit) lang gesperrt (SOK=0). In der vierten Abtastzeit nach Auslösen des SYNC-Kommandos sind die Sende- und Empfangsbausteine freigegeben (SOK=1).

### 3.7.5 Inbetriebnahme/Diagnose

#### 3.7.5.1 Funktionsbausteine Diagnose

**Allgemeines** Mit Hilfe der Funktionsbausteine DPDIAG, DPSLDG und DIAPRO können Master- oder Slave-spezifische Diagnosen vom PROFIBUS DP ausgegeben werden.

**Weitere Informationen**  
zu Diagnosedaten siehe Benutzerdokumentation zu den einzelnen Slaves.

**Übersicht Diagnosedaten** Baustein **DPDIAG**: Diagnoseübersicht

- Die System-Diagnose gibt eine Übersicht darüber, welcher Slave Diagnose gemeldet hat.
- Die 4 Doppelworte sind bitcodiert.
- Jedes Bit ist entsprechend folgender Tabelle einem Slave mit seiner PROFIBUS-Adresse zugeordnet.
- Ist das Bit für den zugeordneten Slave gesetzt, so hat der Slave Diagnose gemeldet.

Ausgang	Bit 16	Bit 15	Bit 14	...	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
DG1	15	14	13	...	4	3	(2)	(1)	(0)
	31	30	29		20	19	18	17	16
...	...								...
DG4	111	110	109		100	99	98	97	96
	-	-	(125)	...	116	115	114	113	112

Tabelle 3-13 Zuordnung System-Diagnose/Datentransfer-Liste zu Slave-PROFIBUS-Adresse

#### Datentransfer-Liste:

- Die Datentransfer-Liste gibt eine Übersicht darüber, mit welchem Slave innerhalb einer projektierten Zeit Datentransfer stattgefunden hat.
- Die 4 Doppelworte (DL1 – DL4) sind bitcodiert wie bei der System-Diagnose.
- Ist das Bit für den zugeordneten Slave gesetzt, so findet mit ihm Datentransfer statt.



**Master-Status:**

- Ausgabe von Master-spezifischen Informationen:

Ausgang	Bedeutung
MST	Status des DP-Masters: Stop (40h), Clear (80h), Operate (C0h)
ID	Ident-Nr. : 815Eh für CP50M1

Tabelle 3-14 Master-spezifische Informationen

**Baustein DPSLDG: Slave-Diagnose**

- Ausgabe von Slave-Diagnose.
- Die Angabe SEL entspricht der Slave-PROFIBUS-Adresse.
- Die Diagnosedaten sind abhängig vom Typ des Slaves.
- Es werden die ersten 16 Byte Slave-Diagnose ausgegeben.
- Weitere Slave-Diagnosedaten können mit SEL>1000 ausgegeben werden.

**Weitere Informationen**

zu Slave-spezifischen Diagnosedaten siehe Benutzerdokumentation zu den jeweiligen PROFIBUS-Slaves.

**Diagnosedaten  
von DP-Slaves**

Anschluss		
ST1	Status 1	Diagnose nach Norm 6 Byte
ST2	Status 2	
ST3	Status 3	
MPA	Master-PROFIBUS-Adresse	
ID	Identnummer	
D01 – D59	gerätespezifische Diagnosedaten (siehe Benutzerdokumentation zu den jeweiligen PROFIBUS-Slaves)	

Tabelle 3-15 Überblick über den Aufbau der Diagnosedaten von DP-Slaves

**Bits von Status 1, 2 und 3**

	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
<b>Status 1 (ST1)</b>	S: Slave wurde von einem anderen Master parametrier	S: letztes Parame-tertele-gramm war fehlerhaft	M: Slave antwortet fehlerhaft	S: Angefor-derte Funktion wird nicht unter-stützt	S: Dia-gnose-eintrag im spezi-fischen Diagnose-bereich	S: Konfigu-rations-daten stimmen nicht überein	S: Slave noch nicht für Datenaus-tausch bereit	M: Slave nicht am Bus erreichbar
<b>Status 2 (ST2)</b>	M: Slave als "nicht aktiv" eingetra-gen	(nicht ver-wendet)	S: Slave hat Sync-Kom-mando erhalten	S: Slave hat Freeze-Kom-mando erhalten	S: Ansprech-über-wachung aktiviert	S: 1 (fest)	S: Diagnose-daten müssen abgeholt werden	S: Para-metrie-rung und Konfigu-rierung erforder-lich
<b>Status 3 (ST3)</b>	S/M: Diagnose-daten nicht alle übertrag-bar	-	-	-	-	-	-	-

Tabelle 3-16 Bedeutung der einzelnen Bits von Status 1, 2 und 3

- **M:** Master erkennt Diagnose
- **S:** Slave erkennt Diagnose

**Master-PROFIBUS-Adresse (MPA)**

- PROFIBUS-Adresse des Masters, der diesen Slave parametrier hat.  
Falls der Slave nicht parametrier ist, dann FFh.

**Identnummer (ID)**

- Kennung zur Identifizierung des Slave-Typs.

Alle weiteren Diagnosedaten sind Slave-spezifisch.

Im allgemeinen (DP-Norm-Slave) folgen die Diagnoseblöcke: gerätebezogene, kennungsbezogene und kanalbezogene Diagnose. Nicht alle Slave-spezifischen Diagnoseblöcke müssen vorhanden sein.

Jedem Block ist ein Headerbyte vorangestellt. Bit 7 und Bit 8 identifizieren den Diagnoseblock:

Bit 7, 8 des Headerbytes	Bedeutung
Bit 7, 8= 00	gerätebezogene Diagnose
Bit 7, 8= 01	kennungsbezogene Diagnose
Bit 7, 8= 10	kanalbezogene Diagnose

Tabelle 3-17 Bedeutung von Bit 7 und Bit 8 des Headerbytes

Bit 1 bis Bit 6 bestimmen:

- Bei geräte- und kennungsbezogener Diagnose die Länge des Diagnoseblocks einschließlich Headerbyte, Wertebereich 2...63
- Bei kanalbezogener Diagnose die Kennungsnummer, Wertebereich 0...63

Baustein DIAPRO (siehe Kap. 3.7.1.5)

### 3.7.5.2 Error-Class (ECL) und Error-Code (ECO)

#### Ausgänge ECL, ECO

Bedeutung der Ausgänge ECL, ECO am Funktionsbaustein @PRODP:

- **Error-Class>0:** Ein Fehler liegt vor. Der Funktionsbaustein @PRODP setzt einen Kommunikationsfehler ab (LED auf der CP50M1 ist an).

Für den Anwender sind diese Anschlüsse kaum von Bedeutung, da die entsprechenden Kommunikationsfehler über den Diagnosepuffer ausgelesen werden können. Sie werden bei Bedarf nach den Werten dieser Anschlüsse gefragt, wenn Sie sich mit schwerwiegenden Fehlern an die Hotline gewendet haben.

### 3.8 Kopplung PROFIBUS DP (CP50M0)

#### Zusätzlich notwendige Hard- und Software

Für Projektierung und Betrieb der Kopplung PROFIBUS DP ist folgende Hardware und Software zusätzlich notwendig:

- **COM PROFIBUS**  
Bestellnummer 6ES5 895-6SE03
- **SS52load**  
SS52load ist in COM PROFIBUS ab V3.1 enthalten.
- **DP-fähige PC-Karte für Download der COM-Datenbasis über COM PROFIBUS:**

#### Eigenschaften

SIMATIC TDC hat am PROFIBUS DP folgende Eigenschaften:

- **Master**  
Das Kommunikationsmodul CP50M0 kann als Master sowohl alleine als auch mit anderen Mastern im Multi-Master-Betrieb am PROFIBUS DP betrieben werden.
- **Slave**  
Neben der Master-Funktionalität gibt es die Slave-Funktionalität. Diese beiden Funktionalitäten können gleichzeitig oder separat verwendet werden.
- **Shared Input**  
Jeder Slave am PROFIBUS DP ist genau einem Master zugeordnet (dem parametrierenden Master) und kann zunächst nur mit diesem kommunizieren. Mit dem "Shared Input" ist es weiteren Mastern möglich, die Eingangsdaten des Slaves zu lesen. SIMATIC TDC unterstützt diese Funktionalität als Master und Slave.
- **SYNC und FREEZE**  
Mit den Diensten SYNC und FREEZE ist ein synchrones Schreiben/Lesen von Ausgängen/Eingängen mehrerer Slaves möglich. SIMATIC TDC unterstützt diese Dienste als Master.
- **Datenlängen**  
Es sind maximal 244 Byte je Richtung und Slave übertragbar.
- **Übertragungszeiten**  
Bei kurzen Telegrammen (bis 32 Byte) geht ausschließlich die SIMATIC TDC-Abtastzeit und die DP-Busumlaufzeit in die Übertragungszeit ein. Bei längeren Telegrammen sind Software-Bearbeitungszeiten des Kommunikationsmoduls CP50M0 einzukalkulieren (max. 5 ms).
- **Konsistenz**  
Die Daten innerhalb eines Telegramms sind immer konsistent.

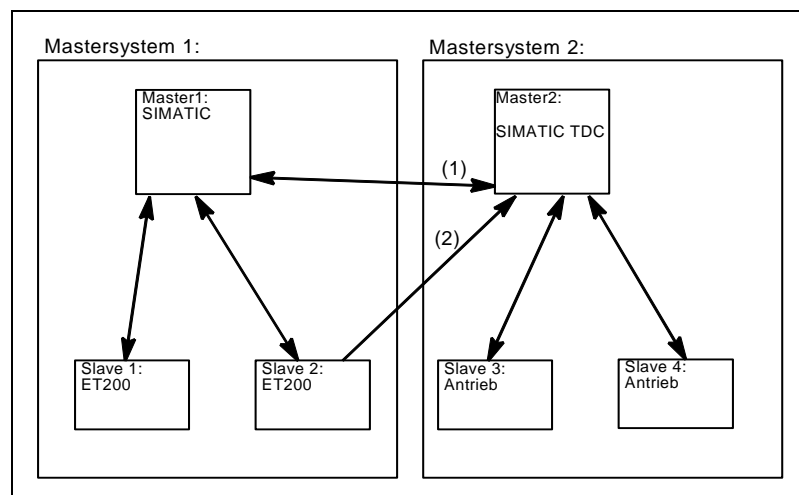


Bild 3-27 Multi-Master-System mit Slave-Funktionalität (1) und Shared Input (2)

### 3.8.1 Projektierung mit D7-SYS

#### Funktionsbausteine

Für eine Kopplung PROFIBUS DP müssen folgende Funktionsbausteine projektiert werden:

- ein Kopplungs-Zentralbaustein @PRODP
- maximal ein Sender- und Empfänger-Funktionsbaustein pro Slave-Station
- maximal ein Synchronisier-Funktionsbaustein SYNPRO kann projektiert werden
- maximal ein Diagnose-Funktionsbaustein DIAL2A kann projektiert werden

#### Kommunikations-Dienst

Erlaubte Kommunikations-Dienste sind:

- Prozessdaten
- Parameterbearbeitung von drehzahlveränderbaren Antrieben

#### Übertragungsmodus

Erlaubter Übertragungsmodus:

- Refresh
- bei Empfängern wahlweise auch Multiple

#### 3.8.1.1 Kopplungs-Zentralbaustein

##### Baudrate und PROFIBUS-Adresse

Baudrate und PROFIBUS-Adresse werden einerseits durch CFC (Funktionsbaustein @PRODP) und andererseits durch COM PROFIBUS vorgegeben.

Zur Gültigkeit dieser beiden Parameter ist folgendes zu beachten:

- Wenn noch keine COM-Datenbasis geladen ist, dann
  - gelten die durch CFC vorgegebenen Parameter,
  - wartet das Kommunikationsmodul CP50M0 auf Download einer COM-Datenbasis.
- Wenn eine COM-Datenbasis geladen ist, Baudrate und PROFIBUS-Adresse mit der CFC-Projektierung übereinstimmen, dann
  - ist die COM-Datenbasis aktiviert,
  - beginnt das Kommunikationsmodul CP50M0 mit dem Nutzdaten-Austausch.
- Wenn eine COM-Datenbasis geladen ist, aber Baudrate oder PROFIBUS-Adresse nicht mit der CFC-Projektierung übereinstimmen, dann
  - gelten die durch CFC vorgegebenen Parameter,
  - wartet das Modul auf Download. (Die vorhandene COM-Datenbasis kann aber auch aktiviert werden, indem Baudrate und PROFIBUS-Adresse am Zentralbaustein der COM-Projektierung angepasst werden.)

### 3.8.1.2 Adressanschlüsse AT, AR

#### Angaben am Adressanschluss AT, AR

Besonderheiten der Angaben am Adressanschluss AT, AR bei Verwendung von PROFIBUS DP:

Eingabereihenfolge:

**“Kanalname.Adressstufe 1.Adressstufe 2“**

- **Kanalname:**
  - Maximal 8 Zeichen
  - ASCII-Zeichen außer „Punkt“ und @
  - Kanalnamen aller Sende- und Empfangsbausteine, die auf dasselbe Kommunikationsmodul SS52 zugreifen, müssen unterschiedlich sein (Ausnahme bei Übertragungsmodus „Multiple“)
  - Der Kanalname hat keine spezifische Bedeutung für PROFIBUS DP
- Eingabe von „.“ nach Kanalname
- **Adressstufe 1:**
  - Als Adressstufe 1 wird die Slave-PROFIBUS-Adresse angegeben
  - Die Slave-PROFIBUS-Adresse darf je Sende- und Empfangskanal nur einmal vergeben werden
  - Wertebereich: 0, 3 - 123

- 0: bedeutet, dass dieser Kanal selbst als Slave-Kanal dient und von einem anderen Master angesprochen werden kann
- 3...123: adressieren externe Slaves
- Eingabe von „.“ nach Adressstufe 1
- **Adressstufe 2:**
  - Besteht aus maximal 2 Zeichen
  - **1. Zeichen:** Byte-Ordering
    - “1“: PROFIBUS-StandardEinstellung  
Die Daten werden im "Motorola-Format" (höherwertiges Byte vor niederwertigem Byte) übertragen
    - “0“: Ausnahmeeinstellung  
Die Daten werden im "Intel-Format" (niederwertiges Byte vor höherwertigem Byte) übertragen. Diese Einstellung kann verwendet werden bei Kommunikationspartnern, deren Datenhaltung intern im Intel-Format erfolgt (z.B. SIMATIC TDC)
  - **2. Zeichen:** optional, nur Empfänger
    - “R“:  
Der Zugriff erfolgt als mitlesender zweiter Master. Die Angabe "R" ist nur bei Empfangskanälen möglich. ("Shared Input")
    - Wird kein 2. Zeichen angegeben, dann erfolgt der Zugriff auf den Slave als parametrierender Master

#### Beispiele für Angaben am Adressanschluss

- AT- ‚Sollwert.25.1‘
  - Der Kanal mit Namen **Sollwert** sendet an einen **Slave** mit der PROFIBUS-Adresse **25**.
- AR- ‚RECEIVE.117.0‘
  - Der Kanal mit Namen **RECEIVE** empfängt von einem **Slave** mit der PROFIBUS-Adresse **117**. Daten werden ausnahmsweise im **Intel-Format** übertragen.
- AR- ‚Eingang.33.1R‘
  - Der Kanal mit Namen **Eingang** empfängt von einem **Slave** mit der PROFIBUS-Adresse **33** als **mitlesender (zweiter) Master**.
- AT- ‚Slavelst.0.1‘
  - Der Kanal mit Namen **Slavelst** sendet als **Slave** an einen **DP-Master**.

### 3.8.1.3 Kommandos SYNC/FREEZE

#### Allgemeines

Die Kommandos SYNC und FREEZE synchronisieren die Ein- und Ausgänge einer Gruppe von Slaves. Der Funktionsbaustein SYNPRO löst diese Kommandos aus und unterstützt die Konsistenzsicherung.

**Konsistenz** Für die Sicherstellung der Konsistenz ist der Projektteur verantwortlich. Beim SYNC/FREEZE-Kommando geht es um Konsistenz der Daten über alle beteiligten Slaves. Die Konsistenz der Ein- oder Ausgangsdaten eines Slaves ist selbstverständlich immer gewährleistet.

**SYNC** Nach Auslösen eines SYNC-Kommandos wartet der DP-Master (CP50M0) eine DP-Busumlaufzeit ab, damit alle Slaves die neuen Ausgangswerte erhalten. Anschließend sendet der DP-Master ein SYNC-Broadcast-Telegramm an die projektierte Slave-Gruppe. Alle Slaves dieser Gruppe aktualisieren daraufhin gleichzeitig ihre zwischengespeicherten Ausgänge.

Konsistenzsicherung:  
Durch Projektierung ist sicherzustellen, dass während einer DP-Busumlaufzeit nach Auslösen des SYNC-Kommandos die Ausgangsdaten nicht durch SIMATIC TDC-CPU's verändert werden.

**FREEZE** Unmittelbar nach Auslösen eines FREEZE-Kommandos sendet der DP-Master ein FREEZE-Broadcast-Telegramm an die projektierte Slave-Gruppe. Alle Slaves dieser Gruppe lesen daraufhin gleichzeitig ihre Eingänge ein und speichern sie zwischen. Nach Ablauf einer DP-Busumlaufzeit liegen diese Eingangsdaten bei den SIMATIC TDC-CPU's vor.

Konsistenzsicherung:  
Durch Projektierung ist sicherzustellen, dass während einer DP-Busumlaufzeit nach Auslösen des FREEZE-Kommandos die Eingangsdaten nicht durch den DP-Master ausgewertet werden.

### 3.8.1.4 SYNC/FREEZE-Projektierungsvarianten

**Allgemeines** Im folgenden werden Begriffe zur Konsistenzsicherung erklärt und verschiedene SYNC/FREEZE-Projektierungsvarianten dargestellt.

- Begriffe**
- **Busumlaufzeit**  
Zyklus, in dem der DP-Master (CP50M0) einmal alle Slaves anspricht. In Multimastersystemen pollen alle Master alle ihre Slaves. Die Busumlaufzeit wird durch Baudrate, Anzahl und Typ der Slaves mit COM PROFIBUS projektiert und vom COM PROFIBUS berechnet. Sie kann dort mit dem Menübefehl **Busparameter** als "Typische Datenzykluszeit" ausgelesen werden.
  - **Abtastzeit**  
Zyklus, in dem der Funktionsbaustein SYNPRO und die Sende- und Empfangs-Funktionsbausteine (auf SIMATIC TDC-CPU's) gerechnet werden. Die Abtastzeit wird mit CFC projektiert.

---

**HINWEIS** Busumlaufzeit und Abtastzeit sind unabhängig voneinander.

---



- **Synczyklus**

Synczyklus ist ein ganzzahliges Vielfaches der Abtastzeit, projektierbar am Eingang CNX des Funktionsbausteins SYNPRO (Synczyklus=CNX x Abtastzeit).

Ein Synczyklus beginnt immer mit einer Abtastzeit. Ein Sync-Kommando wird durch den Funktionsbaustein SYNPRO immer im Systemmode zu Beginn einer Abtastzeit ausgelöst.

### Projektierungs- variante 1

Projektierungsvariante 1 entspricht den meisten Anwendungsfällen:

- Erzeugen von SYNC-Kommandos
- Die Konsistenz über alle Slaves ist gewährleistet
- Der Synczyklus ist mindestens doppelt so groß wie die Abtastzeit (CNX>1)
  - Die Länge der Sendetelegramme (Ausgänge) je Slave darf nicht grösser als 32 Byte sein
  - Alle Sendebausteine und der Funktionsbaustein SYNPRO sind in derselben Abtastzeit zu projektieren
  - Der Funktionsbaustein SYNPRO ist in der zeitlichen Bearbeitungsreihenfolge vor allen Sendebausteinen zu projektieren
  - Der Ausgang SOK des Funktionsbausteins SYNPRO ist mit den Enable-Eingängen aller (zur Slave-Gruppe gehörenden) Sendebausteine zu verbinden
  - Die Busumlaufzeit muss kürzer sein als der Synczyklus minus 1 x Abtastzeit. Es ist zur Laufzeit zu überprüfen, ob der Ausgang SOK in jedem Synczyklus einmal auf "1" geht, ansonsten ist der Synczyklus zu erhöhen

Beispiel:

- Synczyklus=3 x Abtastzeit
- Busumlaufzeit=2 x Abtastzeit
- Annahme: Funktionsbaustein SYNPRO rechnet in der Mitte der Abtastzeit (vor allen Sendebausteinen)

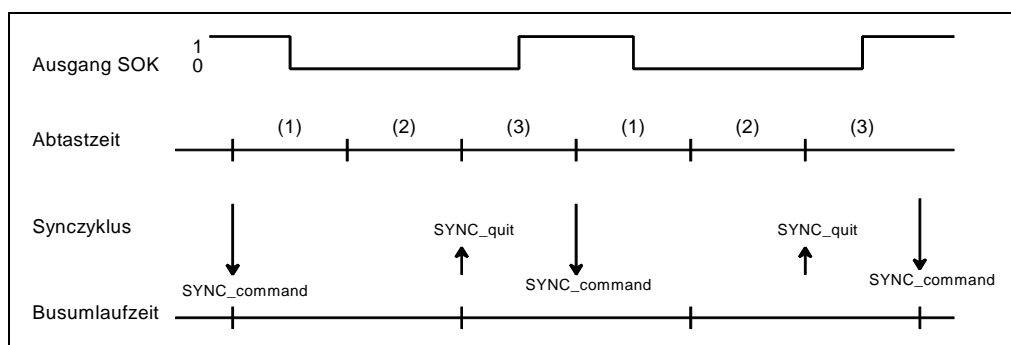


Bild 3-28 Zeitdiagramm SYNC-Variante 1

Nach Auslösen des SYNC-Kommandos sind die Sendebausteine zwei Abtastzeiten (eine Busumlaufzeit) lang gesperrt (SOK=0). In der dritten Abtastzeit nach Auslösen des SYNC-Kommandos werden die Sendebausteine freigegeben (SOK=1).

**Projektierungs-  
variante 2**

Projektierungsvariante 2 hat die höchste SYNC-Performance:

- Erzeugen von SYNC-Kommandos
- Die Konsistenz über alle Slaves ist gewährleistet
- Synczyklus=Abtastzeit (CNX=1)
  - Die Länge der Sendetelegramme (Ausgänge) je Slave darf nicht größer als 32 Byte sein
  - Alle Sendebausteine und der Funktionsbaustein SYNPRO sind in derselben Abtastzeit zu projektieren
  - Hohe Baudrate (>1,5 MBaud); mit niedrigeren Baudraten sind die Zeitbedingungen kaum einzuhalten
  - Die Busumlaufzeit darf höchstens die Hälfte der Abtastzeit betragen
  - Die Busumlaufzeit muss darüber hinaus so klein sein, dass sie von Beginn einer Abtastzeit bis zur Berechnung des Funktionsbausteins SYNPRO verstrichen ist. Diese Voraussetzung kann nicht garantiert werden, sondern diese muss zur Laufzeit überprüft werden

Beispiel:

- Synczyklus=Abtastzeit
- Busumlaufzeit=0,3 x Abtastzeit
- Annahme: Funktionsbaustein SYNPRO rechnet in der Mitte der Abtastzeit (vor allen Sendebausteinen)

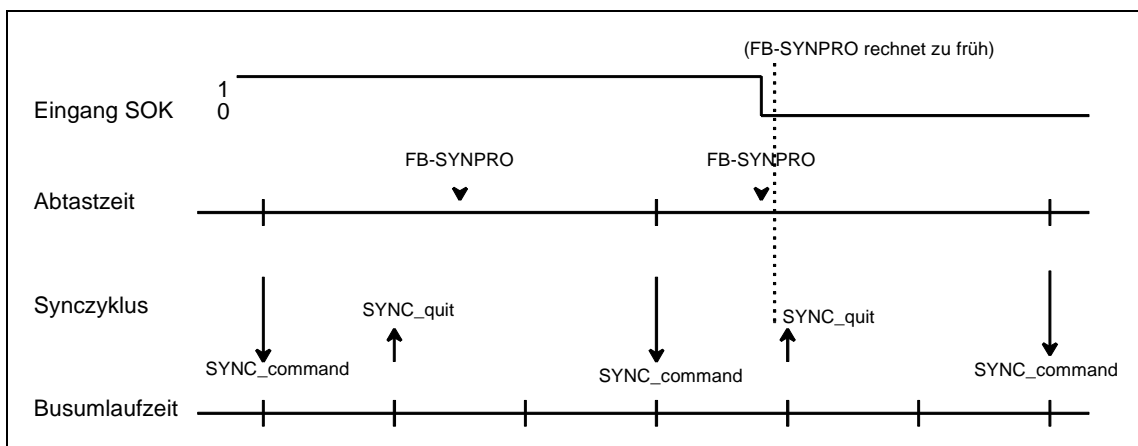


Bild 3-29 Zeitdiagramm SYNC-Variante 2

Im Normalfall sind die Sendebausteine immer freigegeben (SOK=1). Falls aufgrund von Zeitschwankungen der Funktionsbaustein SYNPRO vor Ablauf des SYNC gerechnet wird (rechts im Bild), so werden die Sendedaten nicht aktualisiert, sondern die Werte aus der vorherigen Abtastzeit übertragen. Der Synczyklus und die Konsistenz werden dadurch nicht beeinflusst.

#### **Hinweise für gute SYNC-Funktionalität:**

Neben einem kleinen Synczyklus ist ein möglichst geringer Jitter (zeitliche Schwankung) im Synczyklus erforderlich. Folgende Vorkehrungen unterstützen dies:

- Unregelmäßiger Datenverkehr auf dem DP-Bus ist zu vermeiden; Single-Master-Betrieb; kein temporäres Zuschalten von Stationen
- Keine Alarm-Tasks auf der gleichen SIMATIC TDC-CPU projektieren. Die Abtastzeit darf keinesfalls überlaufen; dies würde zu einem Ausfall eines SYNC-Kommandos oder zur Verschiebung um eine ganze Abtastzeit führen
- Eine hohe Baudrate und kurze Telegrammlängen projektieren. (In den Jitter geht die Zeit für das Pollen eines Slaves ein.)
- Den Funktionsbaustein SYNPRO und alle zugehörigen Sendebausteine in  $T1=T0$  (Grundabtastzeit) projektieren. Das SYNC-Kommando wird dann mit dem Grundtakt-Interrupt ausgelöst. Der Grundtakt-Interrupt kommt zeitlich genauer als ein im System-Mode ausgelöster Interrupt

#### **Projektierungs- variante 3**

Projektierungsvariante 3 entspricht weniger üblichen Anwendungsfällen des FREEZE:

- Erzeugen von SYNC- und FREEZE- oder nur FREEZE-Kommandos
- Die Konsistenz über alle Slaves ist gewährleistet
- Der Synczyklus ist mindestens dreimal so groß wie die Abtastzeit ( $CNX > 1$ ).
  - Die Länge der Sende- oder Empfangstelegramme (Ein- oder Ausgänge) je Slave darf nicht größer als 32 Byte sein
  - Alle Sende- und Empfangsbausteine und der Funktionsbaustein SYNPRO sind in derselben Abtastzeit (auf derselben CPU) zu projektieren
  - Der Funktionsbaustein SYNPRO ist als letzter Funktionsbaustein in der zeitlichen Bearbeitungsreihenfolge zu projektieren
  - Der Ausgang SOK des Funktionsbausteins SYNPRO ist mit den Enable-Eingängen aller (zur Slave-Gruppe gehörenden) Sende- und Empfangsbausteine zu verbinden
- Die Busumlaufzeit muss kürzer sein als der Synczyklus minus 2 x Abtastzeit. Es ist zur Laufzeit zu überprüfen, ob der Ausgang SOK in jedem Synczyklus einmal auf "1" geht, ansonsten ist der Synczyklus zu erhöhen

Beispiel:

- Synczyklus=4 x Abtastzeit
- Busumlaufzeit=2 x Abtastzeit
- Annahme:  
Funktionsbaustein SYNPRO rechnet in der Mitte der Abtastzeit (nach allen Empfangs- und Sendebausteinen)

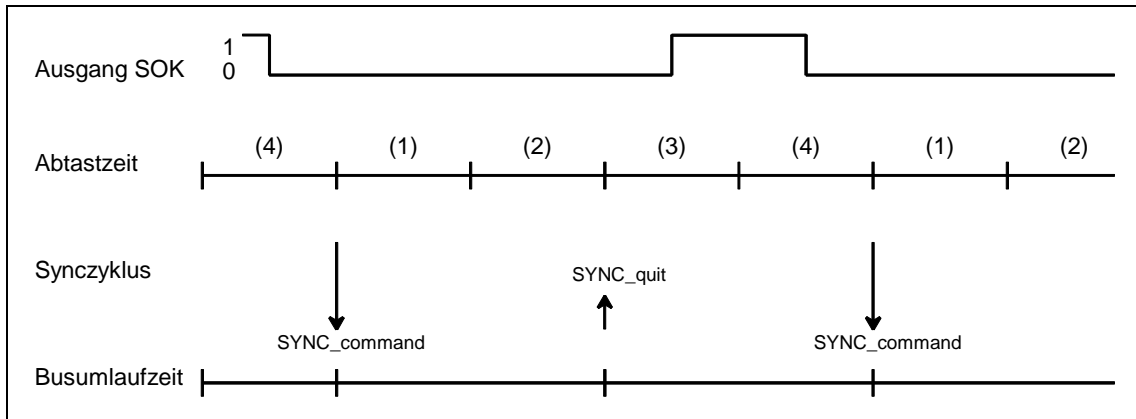


Bild 3-30 Zeitdiagramm SYNC-Variante 3

Nach Auslösen des SYNC-Kommandos sind die Sende- und Empfangsbausteine drei Abtastzeiten (eine Busumlaufzeit und eine Abtastzeit) lang gesperrt (SOK=0). In der vierten Abtastzeit nach Auslösen des SYNC-Kommandos sind die Sende- und Empfangsbausteine freigegeben (SOK=1).

### 3.8.1.5 Funktionsbaustein Diagnose

#### Allgemeines

Mit Hilfe des Funktionsbausteins DIAPRO können Master- oder Slave-spezifische Diagnosen vom PROFIBUS DP ausgegeben werden.

Mit dem Eingang SEL werden die auszugebenden Diagnosedaten ausgewählt. An den Ausgängen D01 bis D08 werden sie ausgegeben.

#### Weitere Informationen

zu Diagnosedaten siehe Benutzerdokumentation „COM PROFIBUS“ oder in der Benutzerdokumentation zu den einzelnen Slaves.

#### Übersicht Diagnosedaten

##### SEL=0: keine Diagnose

- Der Baustein gibt keine gültigen Diagnosedaten aus.

##### SEL=126: System-Diagnose

- Die System-Diagnose gibt eine Übersicht darüber, welcher Slave Diagnose gemeldet hat.
- Die 8 Worte sind bitcodiert.

- Jedes Bit ist entsprechend folgender Tabelle einem Slave mit seiner PROFIBUS-Adresse zugeordnet.
- Ist das Bit für den zugeordneten Slave gesetzt, so hat der Slave Diagnose gemeldet.

Ausgang	Bit 16	Bit 15	Bit 14	...	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
D01	15	14	13	...	4	3	(2)	(1)	(0)
D02	31	30	29		20	19	18	17	16
...	...								...
D07	111	110	109		100	99	98	97	96
D08	-	-	(125)	...	116	115	114	113	112

Tabelle 3-18 Zuordnung System-Diagnose/Datentransfer-Liste zu Slave-PROFIBUS-Adresse

### SEL=127: Datentransfer-Liste

- Die Datentransfer-Liste gibt eine Übersicht darüber, mit welchem Slave innerhalb einer projektierten Zeit (COM PROFIBUS) Datentransfer stattgefunden hat.
- Die 8 Worte sind bitcodiert wie bei der System-Diagnose.
- Ist das Bit für den zugeordneten Slave gesetzt, so findet mit ihm Datentransfer statt.

### SEL=128: Master-Status

- Ausgabe von Master-spezifischen Informationen. (Für den Anwender ist hier das low Byte von D01 relevant; die Bedeutung der anderen Ausgänge ist zwar dokumentiert, aber nicht näher erläutert):

Ausgang		Bedeutung
D01	low Byte	Status des DP-Masters: Stop (40h), Clear (80h), Operate (C0h)
	high Byte	Ident-Nr. SS52 (high Byte)=80h
D02	low	Ident-Nr. SS52 (low Byte)=37h
	high	(don't care)
D03...D08		

Tabelle 3-19 Master-spezifische Informationen

### SEL=3 ... 123: Slave-Diagnose

- Ausgabe von Slave-Diagnose.
- Die Angabe SEL entspricht der Slave-PROFIBUS-Adresse.
- Die Diagnosedaten sind abhängig vom Typ des Slaves.
- Es werden die ersten 16 Byte Slave-Diagnose ausgegeben.

- Weitere Slave-Diagnosedaten können mit SEL>1000 ausgegeben werden.

**Weitere Informationen**

zu Slave-spezifischen Diagnosedaten siehe Benutzerdokumentation „COM PROFIBUS“ und Benutzerdokumentation zu den jeweiligen PROFIBUS-Slaves.

**Diagnosedaten von**

**SIEMENS DP-Slaves**

Slave-Typ		SPC-Slave all-gemein	ET 200U	ET 200B	ET 200K	SPM-Slave	ET 200C 8DE/8DA	DP-Norm-slaves	
Anschluß									
D01	low	Status 1							Diagnose nach Norm 6 Byte
	high	Status 2							
D02	low	Status 3							
	high	Master-PROFIBUS-Adresse							
D03	low	Identnummer high Byte							
	high	Identnummer low Byte							
D04	low	Header gerätebezogene Diagnose							geräte-spezifische Diagnose
	high	Gerätediagnose U	Geräte-diag B	0	0	0			
D05	low	Header kennungsbezogene Diagnose	0	0	0	0			
	high	BG 7-0	0	Kanal 7-0					
D06	low	BG 15-8	0	Kanal 15-8		0			
	high	BG 23-16	0	Kanal 23-16		0			
D07	low	BG 31-24	0	Kanal 31-24		0			
	high	weitere gerätespe-	don't care						
D08	low	zifische	don't care						
	high	Diagnose	don't care						

Tabelle 3-20 Überblick über den Aufbau der Diagnosedaten von Siemens DP-Slaves

### Bits von Status 1, 2 und 3

	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
<b>Status 1</b> (D01 low Byte)	S: Slave wurde von einem anderen Master parametrier	S: letztes Parame- tertele- gramm war fehlerhaft	M: Slave antwortet fehlerhaft	S: Angefor- derte Funktion wird nicht unter- stützt	S: Dia- gnose- eintrag im spezi- fischen Diagnose- bereich	S: Konfigu- rations- daten stimmen nicht überein	S: Slave noch nicht für Datenaus- tausch bereit	M: Slave nicht am Bus erreichbar
<b>Status 2</b> (D01 high Byte)	M: Slave als "nicht aktiv" eingetra- gen	(nicht ver- wendet)	S: Slave hat Sync- Kom- mando erhalten	S: Slave hat Freeze- Kom- mando erhalten	S: Ansprech- über- wachung aktiviert	S: 1 (fest)	S: Diagnose daten müssen abgeholt werden	S: Para- metrie- rung und Konfigu- rierung erforder- lich
<b>Status 3</b> (D02 low Byte)	S/M: Diagnose- daten nicht alle übertrag- bar	-	-	-	-	-	-	-

Tabelle: 3-21 Bedeutung der einzelnen Bits von Status 1, 2 und 3

- **M:** Master erkennt Diagnose
- **S:** Slave erkennt Diagnose

#### Master-PROFIBUS-Adresse

- PROFIBUS-Adresse des Masters, der diesen Slave parametrier hat.

Falls der Slave nicht parametrier ist, dann FFh.

#### Identnummer

- high/low Byte:  
Kennung zur Identifizierung des Slave-Typs.

Alle weiteren Diagnosedaten sind Slave-spezifisch.

Im allgemeinen (DP-Norm-Slave) folgen die Diagnoseblöcke: gerätebezogene, kennungbezogene und kanalbezogene Diagnose. Nicht alle Slave-spezifischen Diagnoseblöcke müssen vorhanden sein.

Jedem Block ist ein Headerbyte vorangestellt. Bit 7 und Bit 8 identifizieren den Diagnoseblock:

Bit 7, 8 des Headerbytes	Bedeutung
Bit 7, 8= 00	gerätebezogene Diagnose
Bit 7, 8= 01	kennungsbezogene Diagnose
Bit 7, 8= 10	kanalbezogene Diagnose

Tabelle 3-22 Bedeutung von Bit 7 und Bit 8 des Headerbytes

Bit 1 bis Bit 6 bestimmen:

- Bei geräte- und kennungsbezogener Diagnose die Länge des Diagnoseblocks einschließlich Headerbyte, Wertebereich 2...63
- Bei kanalbezogener Diagnose die Kennungsnummer, Wertebereich 0...63

#### Ausgabe weiterer Slave-Diagnose

- Mit SEL=1002 bis SEL=1123 werden die Diagnosebytes 17 bis 32 eines Slaves ausgegeben

### 3.8.2 Projektierung mit COM PROFIBUS

#### Allgemeines

Für die Projektierung ist der COM PROFIBUS (Windows) einzusetzen (Projektierung mit dem Vorgänger COM ET200 Version 2.1 ist auch möglich). Mit Hilfe von COM PROFIBUS bestimmen Sie:

- Anzahl und Konfiguration der am Bussystem PROFIBUS DP angeschlossenen Teilnehmer
- die Baudrate

wichtige Parameter für den Betrieb des Bussystems PROFIBUS DP  
SIMATIC TDC-spezifische Hinweise zum COM PROFIBUS:

- Projektieren Sie die Kommunikationsbaugruppe CP50M0 als SIMADYN D SS52-Stationstyp
- Ein und Ausgangsadressen sind nicht anzugeben
- Nach Fertigstellung der Projektierung wird mit Menübefehl **Datei > Export > DP-Master** die Datenbasis über den DP-Bus zur CP50M0 heruntergeladen
- Alternativ ist auch ein Download über RS232 möglich. Hierzu wird mit dem Menübefehl **Datei > Export > SIMADYN-Master** das SS52load gestartet.

#### 3.8.2.1 Abgleich mit CFC-Projektierung

#### Regeln

Die Projektierungen sind wie folgt aufeinander abzustimmen:

- Baudrate und eigene PROFIBUS-Adresse müssen übereinstimmen.



	<ul style="list-style-type: none"> <li>• Die im COM projektierten Slaves müssen in Empfangs- und Senderichtung durch je einen CRV/CTV-Funktionsbaustein in der CFC-Projektierung vertreten sein. Die Zuordnung geschieht über die PROFIBUS-Adresse (Adressstufe 1 am Adressanschluss).</li> <li>• Die Länge der Eingangs- (Empfangs-) und Ausgangs- (Sende-) Daten pro Slave muss übereinstimmen.</li> </ul>
<b>Fehler- und Warnhinweise</b>	<p>Die Regeln werden überprüft. Bei Nichtbeachtung gibt es Fehler- oder Warnhinweise:</p> <ul style="list-style-type: none"> <li>• Kommunikationsfehlerfeld (blinkendes „C“ auf der CPU-Baugruppe), bzw. Ausgang YTS an Funktionsbaustein CRV/CTV</li> <li>• Ausgang ECO am Funktionsbaustein @PRODP</li> </ul>
<b>HINWEIS</b>	<p>Folgende Regel wird nicht überprüft: Die Nutzdatenstruktur muss bei den Kommunikationspartnern übereinstimmen.</p> <p>Bei Nichtbeachtung kann es zu falscher Interpretation der Daten (z.B. vertauschte Bytes innerhalb eines Datenworts) zwischen den Kommunikationspartnern kommen.</p>
<b>Nutzdatenstruktur</b>	<p>Bei SIMATIC TDC wird die Nutzdatenstruktur mit CFC durch die Projektierung der virtuellen Verbindungen (siehe Kapitel Kommunikations-Dienst Prozessdaten) vorgegeben.</p> <ul style="list-style-type: none"> <li>• Bei den meisten PROFIBUS-Slaves wird die Nutzdatenstruktur mit COM PROFIBUS durch die Angabe von Kennungen im Fenster „Konfigurieren“ vorgegeben.</li> </ul>
<b>3.8.2.2 CP50M0 als PROFIBUS-Slave</b>	
<b>Projektierung</b>	<p>Das Kommunikationsmodul CP50M0 kann als reiner Slave projektiert werden oder kombiniert als Master und Slave:</p> <ul style="list-style-type: none"> <li>• CP50M0 als reiner Slave kommt ohne COM-Projektierung aus: Am Funktionsbaustein @PRODP ist der Eingang SLA auf 1 oder 2 zu setzen. Außerdem ist ein Funktionsbaustein CRV und/oder CTV zu projektieren. Die Adressstufe 1 am AR-/At-Anschluss ist auf „0“ zu setzen.</li> <li>• CP50M0 kombiniert als Master und Slave Der Eingang SLA am Funktionsbaustein @PRODP ist auf „0“ zu setzen (Vorbelegung).</li> </ul>

- Die Busprojektierung erfolgt mit COM PROFIBUS. Je PROFIBUS-Master entsteht eine Datenbasis („Mastersystem“). Diese ist zum jeweiligen Master runter zu laden.
- Wird der Master mit einem anderen Werkzeug projektiert, so muss bei der Projektierung des CP50M0-Slaves ein fiktiver Master in COM PROFIBUS projektiert werden. Zu beachten ist dabei die korrekte Einstellung der Busparameter: Zu empfehlen ist die Erhöhung der Anzahl aktiver Stationen bzw. die Token-Rotation-Time in beiden Projektierungswerkzeugen.

### 3.8.2.3 Laden der Datenbasis

#### Varianten

Für das Laden der Datenbasis gibt es zwei Varianten:

#### Laden über PROFIBUS DP

- Das Laden über PROFIBUS DP ist die komfortablere Variante. Allerdings müssen Einschränkungen beachtet werden.
- Benötigt werden eine DP-fähige PC-Karte (aktuell lieferbare Karten können vom Produkt Support erfragt werden)
- Die zur PC-Karte passenden Treiber werden zusammen mit COM PROFIBUS installiert. Das Laden erfolgt im COM PROFIBUS mit dem Menübefehl **Datei > Export > DP-Master**.

#### Laden über RS232

- Mit Hilfe des „SS52LOAD“-Programms kann eine vom COM PROFIBUS erzeugte Datenbasis als Binärdatei über RS232-Schnittstelle in die CP50M0 geladen werden.
- SS52LOAD ist im COM PROFIBUS (ab Version 3.1) integriert.
- Einschränkung:  
Wenn der Sync-Funktionsbaustein SYNPRO projektiert ist, dann muss der Sync-Betrieb abgeschaltet sein (Enable-Eingang EN=0), damit der Download funktioniert.
- Das Erzeugen der Binärdatei (\*.2bf) erfolgt im COM PROFIBUS mit dem Menübefehl **Datei > Export > Binärdatei**.
- Das Laden erfolgt mit SS52LOAD mit Menübefehl **File > Download**.
- Die RS232-Schnittstelle befindet sich zusammen mit der PROFIBUS-Schnittstelle auf dem 9-poligen Stecker der entsprechenden CP50M0-Schnittstelle. Ein Kabel für die Verbindung zum COM-Port des PCs ist selbst anzufertigen.  
RS232-Belegung am Stecker (kein Standard):
  - 2 - TxD
  - 7 - RxD



**VORSICHT** Verwechslungsgefahr bei der RS232-Belegung.

### 3.8.3 Inbetriebnahme/Diagnose

#### 3.8.3.1 Leuchtdioden

**Grüne Leuchtdiode** Die grüne Leuchtdiode gibt generell Auskunft über das Kommunikationsmodul CP50M0 und über die Synchronisation mit dem Funktionsbaustein @PRODP von SIMATIC TDC.

**Gelbe Leuchtdiode** Im Gegensatz zu Kommunikationsbaugruppen zeigt beim Kommunikationsmodul CP50M0 die gelbe Leuchtdiode nicht direkt die Busaktivität an. Die gelbe Leuchtdiode gibt Auskunft über den DP-Bus und die COM-Datenbasis.

LED	Grün	Gelb
aus	CPU läuft nicht	Kein Busbetrieb (während Hochlauf).
blinkt schnell (alle 0,2 s)	Schwerer Fehler <ul style="list-style-type: none"> <li>Abhilfe: Error-Class und -Code am Funktionsbaustein @PRODP auslesen und Siemens AG benachrichtigen.</li> </ul>	Fehler am Bus (z.B. Kurzschluß) <ul style="list-style-type: none"> <li>Abhilfe: Kabel und andere Busteilnehmer überprüfen.</li> </ul>
Blinkt (alle 1 s)	Warten auf Synchronisation mit SIMATIC TDC-CPU <ul style="list-style-type: none"> <li>Abhilfe: Projektierung des Funktionsbausteins @PRODP überprüfen.</li> </ul>	COM-Datenbasis ist nicht vorhanden oder nicht aktiviert (auch während Download) <ul style="list-style-type: none"> <li>Abhilfe: Datenbasis laden.</li> </ul>
blinkt langsam (alle 2 s)	-	CFC- und COM-Projektierung passen nicht (100%-ig) zusammen. Busbetrieb eingeschränkt möglich <ul style="list-style-type: none"> <li>Abhilfe: CFC- und COM-Projektierung anpassen.</li> </ul>
an	Kommunikationsmodul CP50M0 und Synchronisation mit SIMATIC TDC-CPU's ok.	Busbetrieb mit aktivierter COM-Datenbasis ok.

Tabelle 3-23 Bedeutung der LED's des Kommunikationsmoduls CP50M0

#### Verhalten bei Anlauf

- Nach Spannung-Ein werden zunächst beide Dioden kurz ein- und ausgeschaltet.
- Während der Hochlaufzeit (ca. 5 Sek.) leuchtet nur die grüne LED.
- Nach Ablauf der Hochlaufzeit geht im OK-Fall auch die gelbe LED an.
- Nach Reset bleiben beide LED's zunächst im letzten Zustand, bis die Software erneut die LED's ansteuert.

**Verhalten bei Download**

- Während Download blinkt die gelbe LED. (Bei hohen Baudraten ist der Vorgang sehr kurz.)
- Danach ist das Verhalten wie beim Anlauf.

Die LED's geben keine Auskunft, ob alle Slaves am Bus vorhanden und richtig parametrier sind. Wenn der Datenaustausch mit einem Slave nicht ok ist, dann wird dies durch eine "Break"-Kennung am zugehörigen Funktionsbaustein (YEV=0x0002 oder YTS=0x6014) angezeigt. Auskunft über den aktuellen Zustand einzelner Slaves erhält man mit dem Diagnose-Funktionsbaustein DIAPRO.

### 3.8.3.2 Error-Class (ECL) und Error-Code (ECO)

**Ausgänge ECL, ECO**

Bedeutung der Ausgänge ECL, ECO am Funktionsbaustein @PRODP:

- **Error-Class=0:** Eine Warnung liegt vor. Die Warnung kann teilweise ohne Reset durch SIMATIC TDC beseitigt werden. Bei mehreren anstehenden Warnungen wird die mit der niedrigsten Nummer angezeigt.
- **Error-Class>0:** Ein Fehler liegt vor. Der Funktionsbaustein @PRODP setzt einen Kommunikationsfehler ab (blinkendes "C" auf der CPU-Baugruppe). Nach Behebung des Fehlers muß der SIMATIC TDC-Baugruppenträger zurückgesetzt werden.

Error-Class	Error-Code	Bedeutung
0 (Warnung)	0	o.k.
	1	COM-Datenbasis vorhanden, aber nicht aktiviert, da Baudrate und PROFIBUS-Adresse nicht mit den Anschlüssen BDR und MAA übereinstimmen. <ul style="list-style-type: none"> <li>Abhilfe: Baudrate und PROFIBUS-Adresse in CFC und COM-Projektierung angleichen.</li> </ul>
	2	Keine COM-Datenbasis vorhanden. <ul style="list-style-type: none"> <li>Abhilfe: Datenbasis laden.</li> </ul>
	3	Download der COM-Datenbasis mit nachfolgendem Anlauf ist im Gang.
	4	Es fehlen mit CFC zu projektierende Kanäle zu DP-Teilnehmern, die in der COM-Datenbasis projiziert sind. Dieser Zustand kann auch temporär nach SIMATIC TDC-Anlauf auftreten. Die DP-Teilnehmer werden nicht angesprochen. <ul style="list-style-type: none"> <li>Abhilfe: CFC- und COM-Projektierung angleichen.</li> </ul>
	5	(nicht verwendet)
	6	Es gibt mindestens einen mit CFC projizierten Kanal, der nicht zur COM-Datenbasis passt. Der zugehörige SIMATIC TDC-FB hat einen Kommunikationsfehler abgesetzt (blinkendes "C"). <ul style="list-style-type: none"> <li>Abhilfe: CFC- und COM-Projektierung angleichen.</li> </ul>
	7	Es gibt mindestens einen mit CFC projizierten Kanal, der prinzipiell nicht zur COM-Datenbasis passt. Der zugehörige SIMATIC TDC-FB hat einen Kommunikationsfehler abgesetzt (blinkendes "C"). <ul style="list-style-type: none"> <li>CFC-Projektierung korrigieren.</li> </ul>
	8	Ressourcen-Engpass. Nicht alle CFC-Kanäle werden bearbeitet. <ul style="list-style-type: none"> <li>Abhilfe: CFC-Projektierung (Kommunikations-Kanäle) verringern.</li> </ul>
	9	Es gibt zwei Kanäle, die zum gleichen Slave senden bzw. von ihm empfangen wollen. Der zum später angemeldeten Kanal gehörige SIMATIC TDC-FB hat einen Kommunikationsfehler abgesetzt (blinkendes "C"). <ul style="list-style-type: none"> <li>Abhilfe: CFC-Projektierung korrigieren.</li> </ul>
	10	Busbetrieb temporär gestört. <ul style="list-style-type: none"> <li>Abhilfe: Kabel und Busteilnehmer überprüfen.</li> </ul>
>1 (interne Fehler)	(beliebig)	<ul style="list-style-type: none"> <li>Abhilfe: Error-Class und Error-Code notieren und Siemens AG benachrichtigen.</li> </ul>

Tabelle 3-24 Bedeutung Error-Class und Error-Code

### 3.8.3.3 Applikationsbeispiel Kopplung PROFIBUS DP

**Allgemeines** Das Applikationsbeispiel beschreibt eine Musterkonfiguration bestehend aus:

- SIMOVERT 6SE70
- ET200U
- ET200B
- SIMATIC S5

Die Projektierung von SIMATIC TDC sowie die Projektierungssprache CFC werden als bekannt vorausgesetzt.

---

**HINWEIS** Es werden nur Tätigkeiten ausführlich erläutert, die für die ausgewählte Musterkonfiguration von Bedeutung sind. Varianten oder zusätzliche Komponenten werden angerissen, aber nicht detailliert erklärt. Im Text sind diese Stellen mit dem rechts abgebildeten Symbol gekennzeichnet.

---

Folgende Themengebiete werden in diesem Applikationsbeispiel behandelt:

- **Musterkonfiguration**  
Beschreibung einer typischen Konfiguration für SIMATIC TDC am PROFIBUS DP mit den dazugehörigen Systemvoraussetzungen
- **Projektierung unter CFC**  
Zusammenstellung der PROFIBUS DP-spezifischen Bausteine und deren Projektierung in der Musterkonfiguration
- **Konfigurieren des Kommunikationsmoduls auf CP50M0...**  
Konfigurieren des Kommunikationsmoduls auf CP50M0 anhand der Parametriersoftware COM PROFIBUS 3.0 und dem Downloadtool „SS52load“

### 3.8.3.4 Musterkonfiguration und Systemvoraussetzungen

**Allgemeines** Als Musterkonfiguration wurden folgende Systeme und Geräte ausgewählt, wobei die angegebenen PROFIBUS-Adressen willkürlich festgelegt wurden:

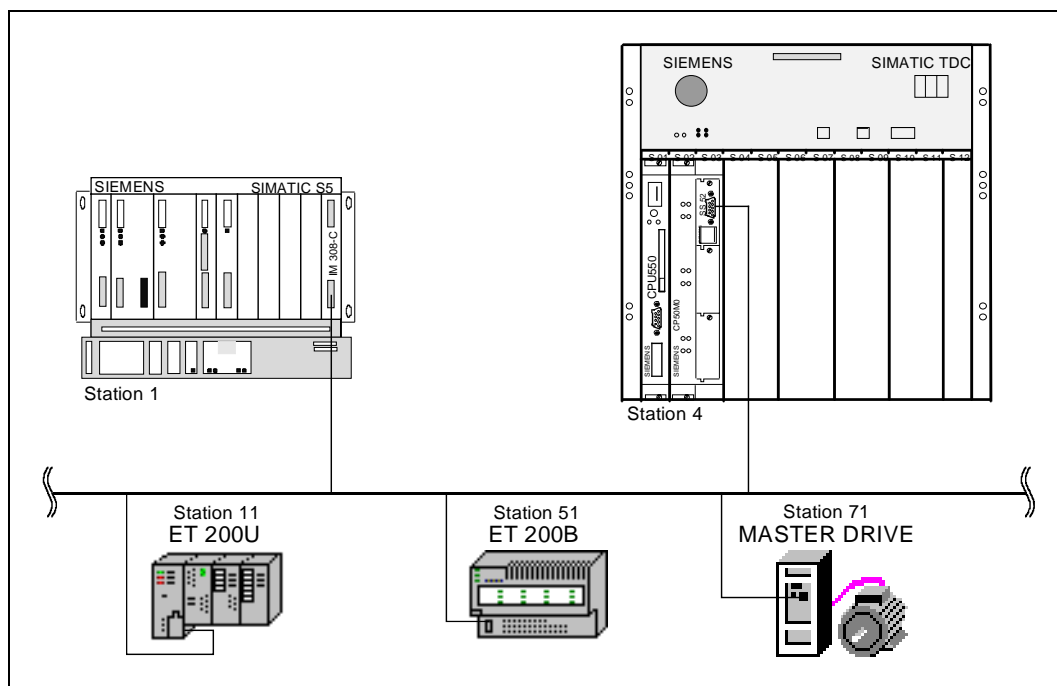


Bild 3-31 Musterkonfiguration

### Kommunikationspartner

Die Kommunikationspartner von SIMATIC TDC (Station 4) sind im einzelnen:

- **SIMATIC S5-105U (Station 1) als Master zu SIMATIC TDC:**  
Der CP50M0 wird ein Master (S5) übergeordnet, von dem aus SIMATIC TDC gepollt wird. Der Datentransfer (Anzahl und Menge von Prozessdaten) zwischen den beiden Steuerungen kann frei gestaltet werden. Festgelegt wurden:
  - S5 ⇒ SIMATIC TDC: drei Worte (Ein-/Ausgabe), ein Wort (Eingabe), ein Byte (Eingabe), ein Byte (Eingabe)
  - SIMATIC TDC ⇒ S5: drei Worte (Ein-/Ausgabe), ein Wort (Ausgabe), ein Byte (Ausgabe)
- **SIMOVERT MASTER DRIVE mit CB1 (Station 71) als Slave:**  
Für den Datenaustausch mit diesem Teilnehmer stehen fünf definierte PPO-Typen zur Verfügung.  
PPO: Parameter-Prozessdaten-Objekte Struktur der Nutzdaten bei drehzahlveränderbaren Antrieben. Es gibt Nutzdaten, die entweder aus Parameter-Kennungs-Werten (PKW) und Prozessdaten (PZD) bestehen (PPO-Typen 1,2,5) oder nur aus Prozessdaten (PPO-Typen 3,4).  
In der Beispielprojektierung ist der PPO-Typ 3 projektiert. Hierbei werden zwei Worte (Steuerwort und Hauptsollwert) gesendet und zwei Worte (Statuswort und Hauptistwert) empfangen.

- **ET 200 B (Station 51) als Slave:**  
Bei Verwendung dieser Slaveart ist eine genaue Typenauswahl vorzunehmen, durch die der Datentransfer automatisch festgelegt wird. Beim 8DI/8DO-Typ wird ein Byte ausgegeben und ein Byte eingelesen.
- **ET 200 U (Station 11) als Slave:**  
Bei dieser ET 200 U-Konfiguration (drei digitale Ausgabemodule und ein digitales Eingabemodul) werden drei Byte ausgegeben und ein Byte eingelesen.

### 3.8.3.5 Checkliste der benötigten Hard- und Softwarekomponenten für SIMATIC TDC

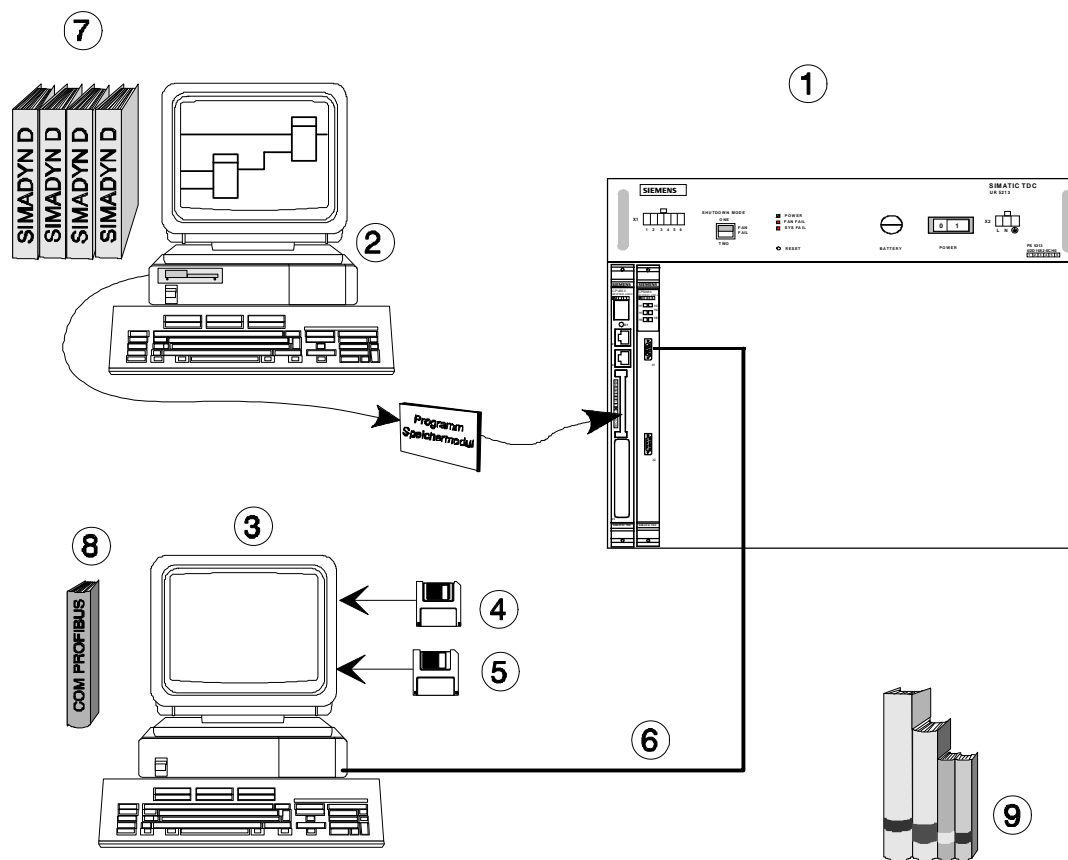


Bild 3-32 Hard- und Softwarekomponenten für SIMATIC TDC



**Legende**

1	SIMATIC TDC-Gerät, mindestens bestehend aus: Baugruppenträger, CPU, Programmspeichersubmodul, Kommunikationsbaugruppe CP50M0
2	CFC-Projektierungsgerät: PC mit Windows 95/NT als Betriebssystem, STEP 7-Software, Optionspaket D7-SYS und PCMCIA-Laufwerk
3	PC zum Betrieb von „COM PROFIBUS“ und „SS52load“ (kann derselbe PC sein, wie für CFC), mit: 3,5“-Diskettenlaufwerk, eine verfügbare serielle Schnittstelle, Betriebssystem Windows 3.1x oder Windows 95
4	Parametriersoftware „COM PROFIBUS 3.0“: Software zur Erstellung der PROFIBUS DP-Buskonfiguration
5	Download-Software „SS52load“: Software zur Übertragung der mit „COM PROFIBUS“ erstellten DP-Konfiguration auf die CP50M0 über den COM-Port (RS 232) eines PCs.
6	RS232-Leitung: Verbindung zwischen CP50M0 (in jedem 9-pol. Stecker der CP50M0 ist neben der RS485 des Profibus auch eine RS232 integriert: 2-TxD; 7-RxD) und einem PC COM-Port (RS232). Dieses Kabel muss nach Vorlage (siehe Kapitel Download der COM-Datenbasis auf CP50M0) angefertigt werden, da die RS232 der CP50M0 kein Standard ist !  Falls bei vorhandenem Kommunikationsprozessor CP 5411 (zusätzliche Steckkarte im PC) der Download über den Bus (RS 485) durchgeführt wird, entfällt das Tool „SS52load“ und die RS232 Leitung. Die CP 5411 ist jedoch nicht Bestandteil dieser Anleitung.
7	Benutzerdokumentation SIMATIC TDC
8	Handbuch zur Parametriersoftware COM PROFIBUS
9	Handbücher der übrigen Busteilnehmer: SIMATIC S5, ET 200U, ET 200B, SIMOVERT Master Drives

Tabelle 3-25 Legende Hard- und Softwarekomponenten für SIMATIC TDC

**3.8.3.6 Projektierung unter STEP 7 CFC****Allgemeines**

Um ein durchgehendes Projektieren einer „Kopplung PROFIBUS DP“ unter CFC zu erleichtern, wird eine Zusammenfassung der busspezifischen CFC-Bausteine und der zu verwendenden Syntax aufgeführt.

Bei der Projektierung eines Kommunikationsschnittstelle auf CP50M0 unter CFC ist zu beachten:

- Pro Kommunikationsschnittstelle CP50M0 genau ein Zentralbaustein @PRODP
- Pro Kommunikationspartner maximal ein Sender- und/oder ein Empfänger-Baustein

- Erlaubte Kommunikations-Dienste:
  - Prozessdaten
  - Parameterbearbeitung von drehzahlveränderbaren Antrieben
- Erlaubter Übertragungsmodus: Refresh (bei Empfängern auch Multiple)
- Pro Kommunikationsschnittstelle auf CP50M0 maximal ein Synchronisations-Funktionsbaustein SYNPRO
- Pro Kommunikationsschnittstelle auf CP50M0 maximal ein Diagnose-Funktionsbaustein DIAPRO

**Funktionsbausteine**

Zentralbaustein PROFIBUS DP-Kopplung @PRODP

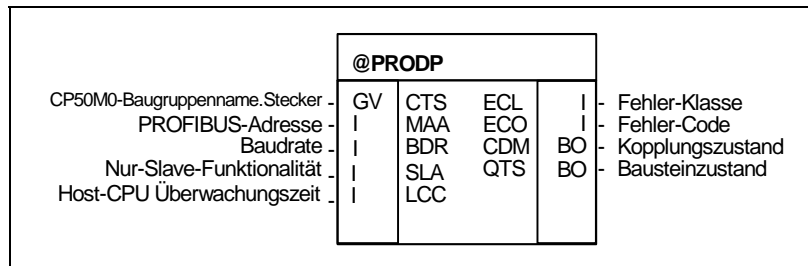


Bild 3-33 Zentralbaustein PROFIBUS DP-Kopplung @PRODP

- **Verwendung**  
Dieser Funktionsbaustein initialisiert und überwacht die PROFIBUS DP-Kopplung (CP50M0). Er darf nur in einer Abtastzeit von  $32\text{ ms} \leq TA \leq 255\text{ ms}$  projektiert werden.
- **Anschlüsse**  
Bei Anschlüssen wie ECL, ECO, CDM, QTS und YTS handelt es sich um Service- bzw. Diagnoseanschlüsse, wie sie für die Inbetriebnahme von SIMATIC TDC üblich sind. Sie werden für unsere Projektierung nicht gebraucht.

**Weitere Informationen**

zu den Anschlüssen des Zentralbausteins PROFIBUS DP-Kopplung @PRODP siehe Benutzerdokumentation Referenzhandbuch „Regelsystem SIMATIC TDC, Funktionsbaustein-Bibliothek“.

<b>CTS</b>	An diesem Initialisierungskonnektor wird der projektierte Name des CP50M0-Moduls (identisch mit dem Eintrag im Masterprogramm, aktuell: D04) und die Bezeichnung des CS7-Steckplatzes, auf der sich die SS52 befindet (X01, X02 oder X03, aktuell: X02), angegeben.
<b>MAA</b>	Die CP50M0-Schnittstelle besitzt, wie alle Busteilnehmer, eine Stationsadresse. Diese muss an diesem Konnektor angegeben werden (Zahl zwischen 1 und 123, aktuell: 4).
<b>BDR</b>	Mit diesem Konnektor wird die Baudrate eingestellt, mit der die CP50M0-Schnittstelle am Bus betrieben wird. Diese Werte müssen in codierter Form angegeben werden: 0=9,6 kBaud ; 1=19,2 kBaud ; 2=93,75 kBaud ; 3=187,5 kBaud ; 4=500 kBaud ; 5=1,5 MBaud ; 6=3 MBaud ; 7=6 MBaud ; 8=12 MBaud ; (aktuell: 5).

<b>SLA</b>	Initialisierungsanschluss für Nur-Slave-Funktionalität: 0: CP50M0 arbeitet als PROFIBUS-Master und/oder Slave. Eine COM PROFIBUS - Datenbasis muss geladen werden. 1 oder 2: CP50M0 arbeitet als reiner PROFIBUS-Slave ohne COM PROFIBUS - Datenbasis 1: Slave mit entweder Eingängen oder Ausgängen, 2: Slave mit Ein- und Ausgängen (aktuell: 0)
<b>LCC</b>	Initialisierungsanschluss für die Zeit, in der die CP50M0 die SIMATIC TDC-Host-CPU überwacht: <0: keine Überwachung 0...10: Überwachungszeit=1s (Default) >10: Überwachungszeit in 1/10 s (aktuell: 0)

Tabelle 3-26 Anschlüsse des Zentralbausteins PROFIBUS DP-Kopplung

### 3.8.3.7 Verwendung von Sende- und Empfangsbausteinen

**Allgemeines** Für PROFIBUS DP sind die Funktionsbausteine des Kommunikations-Dienstes Prozessdaten zu projektieren.

Die Adressanschlüsse AT und AR derjenigen Bausteine, die auf die Datenschnittstelle der CP50M0 zugreifen, müssen folgenden Regeln gerecht werden:

#### **AT/AR- ,Kanalname. Adressstufe 1.Adreßstufe 2'**

##### **Kanalname**

- muss entsprechend der allgemeinen Regeln der Kommunikation eindeutig sein (Kanalnamen aller Sende- und Empfangsbausteine, die auf dieselbe Kommunikationsschnittstelle auf CP50M0 zugreifen, müssen unterschiedlich sein)
- darf höchstens aus 8 Zeichen bestehen
- hat für PROFIBUS DP keine spezielle Bedeutung

##### **Adressstufe 1**

- In dieser Adressstufe wird die PROFIBUS-Adresse des Kommunikationspartners angegeben
- Mit Adresse 0 wird dieser Kanal zum Slave und wird durch andere Busmaster abgerufen
- Mit den Adressen 3..123 können externe Slaves angesprochen werden
- Eine PROFIBUS-Adresse darf je Sende-/Empfangskanal nur einmal vorkommen

**Adressstufe 2**

Diese Adressstufe ist mit einem oder zwei Zeichen zu projektieren:

- **1. Zeichen:** Festlegung des Byte-Orderings zur Übertragung von Wortgrößen für verschiedene Kommunikationspartner.
  - **1=Motorola-Format** (high Byte vor low Byte)  
damit entspricht der Telegrammaufbau der PROFIBUS-Norm, sollte standardmäßig verwendet werden, vor allem bei der Übertragung von Wortgrößen zu Norm-Busteilnehmern (Analog Ein-/Ausgabe, SIMOVERT, SIMATIC usw.)
  - **0=Intel-Format** (low Byte vor high Byte)  
kann eingesetzt werden für Datenübertragung zu Geräten deren Datenverarbeitung wie in SIMATIC TDC nach dem Intel-Format erfolgt (z.B. zweites CP50M0)

Koppelpartner	1. Zeichen
SIMOVERT Master Drives mit CB 1 (genormter Busteilnehmer)	1
Dezentrale Peripherie ET200 (genormter Busteilnehmer)	1
SIMATIC (IM 308 C,...) (genormter Busteilnehmer)	1
SIMOREG 6RA24	1
MICRO / MIDI Master (genormter Busteilnehmer)	1
SIMATIC TDC (CP50M0) (Koppelpartner muss dieselbe Einstellung besitzen)	0

Tabelle 3-27 Byte-Ordering für verschiedene Kommunikationspartner

- **2. Zeichen** (optional, nur für Empfänger):  
mit der Angabe „R“ an einem Empfangskanal erhält die CP50M0 eine lesende Zugriffsberechtigung auf andere Slaves (Shared Input).

**3.8.3.8 Projektierung der Musterkonfiguration in CFC**

**Allgemeines**

Es geht hierbei nicht nur um die Prozessdatenverarbeitung, sondern in erster Linie um die Realisierung der aufgeführten Kommunikationswege zu den übrigen Busteilnehmern.

Ein CFC-Plan mit Erläuterungen zeigt die Projektierung des PROFIBUS DP. Der CFC-Plan erhebt nicht den Anspruch, alle Details zu enthalten.

Zu projektieren sind:

- CPU CPU550 auf Steckplatz S01 unter dem Namen D01P01:
- Kommunikationsbaugruppe CP50M0 auf Steckplatz S02 unter der Bezeichnung D02
- Kommunikationsschnittstelle 1 auf CP50M0 Stecker X01

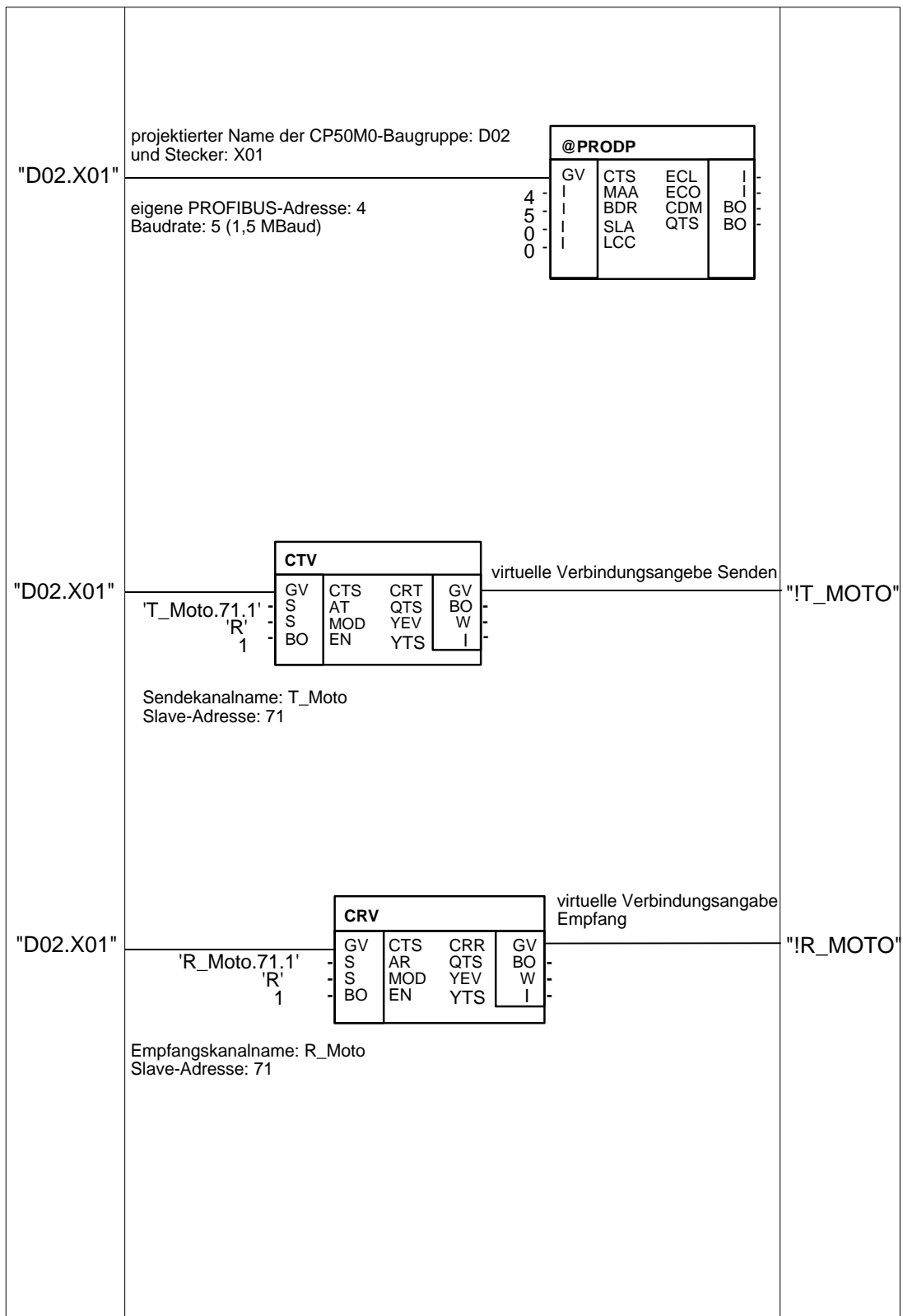


Bild 3-34 CFC-Plan (Teil 1) der Musterkonfiguration

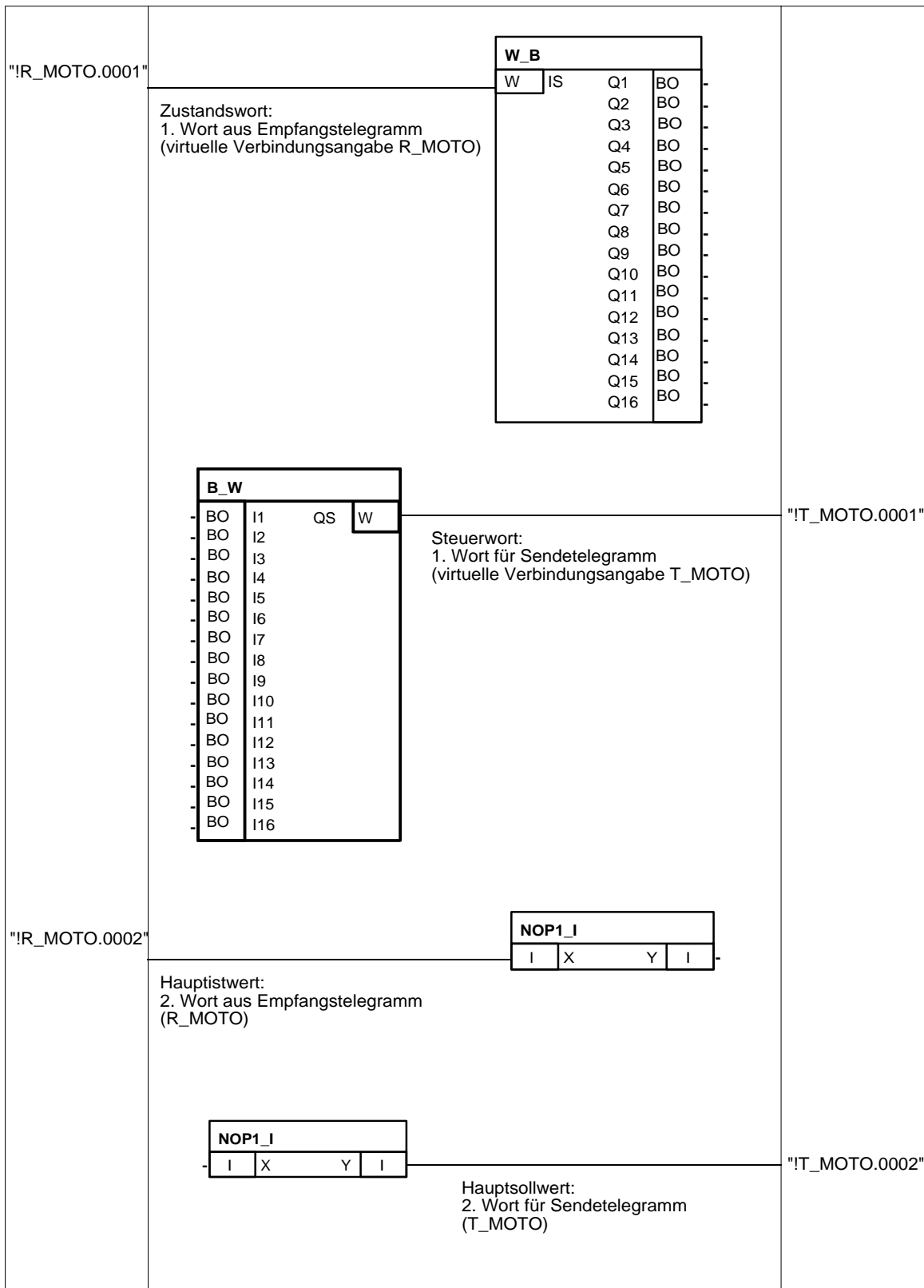


Bild 3-35 CFC-Plan (Teil 2) der Musterkonfiguration

### 3.8.3.9 Konfiguration des Kommunikationsmoduls SS52 mit COM PROFIBUS

#### Allgemeines

Wenn eine Kommunikationsschnittstelle auf der CP50M0 für PROFIBUS (aktuell: X01) projektiert wurde, dann werden Werte zwischen Sende- oder Empfangsbausteinen und dem Busstecker auf dem Kommunikationsmodul CP50M0 ausgetauscht. Da SIMATIC TDC ein frei projektierbares System ist, müssen folgende logische Kommunikationsstrukturen vorgegeben sein:

- Festlegung der Busparameter (Baudrate, ...)
- Beschreibung der Kommunikationsbeziehungen zwischen den Teilnehmern (wer kommuniziert mit wem in welcher Funktion?)
- Definition der Kommunikationsobjekte (Kommunikationsobjekte sind Nutzdaten. Bei SIMATIC TDC setzen sie sich aus Prozess- und Gerätedaten zusammen. Für die Musterkonfiguration läuft die Kommunikation jedoch nur mit Prozessdaten ab.)

Diese Daten (im folgenden als COM-Datenbasis bezeichnet) sind auf der CP50M0 in einem fest integrierten Speicher hinterlegt und werden mittels eines Downloads über den 9-poligen Sub-D-Stecker der Baugruppe verändert und angepasst.

### 3.8.3.10 Generierung der COM-Datenbasis mit COM PROFIBUS

#### Vorgehensweise

Master und Slaves eines Busaufbaus werden über eine graphische Benutzeroberfläche und eine Liste von unterstützten Kommunikationspartnern projektiert..

Zu Beginn werden alle Kommunikationsbeziehungen der Musterkonfiguration festgelegt, indem die beteiligten Teilnehmer ausgewählt werden.

#### Parametrierung des 1. Hostsystems

1. Nach dem Programmstart wird mit dem Menübefehl **Datei > Neu** das erste Mastersystem eingerichtet.

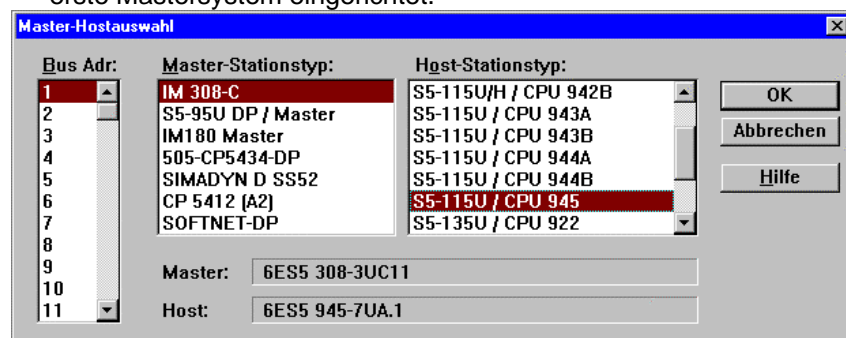


Bild 3-36 Dialogfeld „Master-Hostauswahl“

- Nach Zwischenspeichern (**Datei > Speichern unter...**) mit einem beliebigen Namen (aktuell: „Muster“), ist ein erstes Hostsystem mit der Bezeichnung „Mastersystem <1>“ erstellt. Die Kennzahl (aktuell: 1) ist dabei identisch mit der eingestellten PROFIBUS-Adresse. Mit diesem ersten Schritt wurde festgelegt, wer an diesem Hostsystem das „Sagen“ hat.

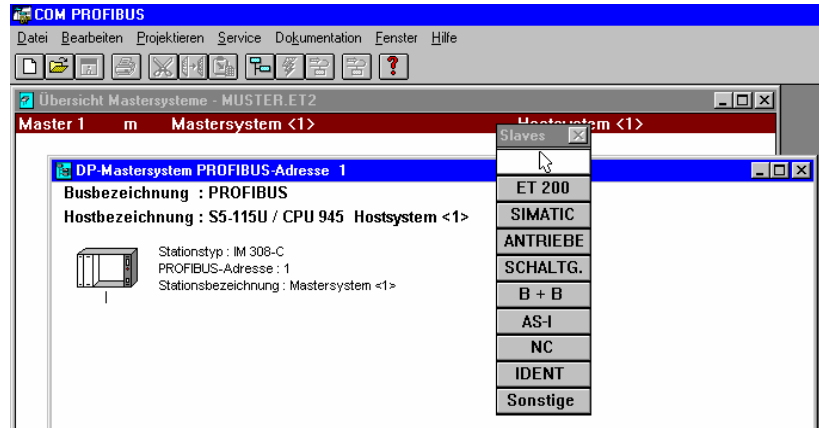


Bild 3-37 Fenster „DP-Mastersystem PROFIBUS-Adresse 1“

- Nach Anwählen der Schaltfläche „ET200“ im Menü „Slaves“ ändert sich der Mauszeiger zu einem leeren Kasten mit einem nach oben zeigenden Pfeil.



Damit können der S5-Station jetzt Slaves zugewiesen werden, indem der Mauszeiger unter dem Stationssymbol positioniert und ein Mausklick ausgeführt wird.

- Nach Abfrage der PROFIBUS-Adresse (aktuell: 4) kann in einem weiteren Auswahlfenster der Kommunikationspartner selektiert werden.



Bild 3-38 Fenster „PROFIBUS-Adresse“



5. Die meisten Einstellmöglichkeiten im Fenster „Slaveeigenschaften“ sind für die Musterkonfiguration nicht von Bedeutung. Es können die Standardeinstellungen übernommen werden. Lediglich die Familie (aktuell: SIMADYN), der Stationstyp (aktuell: „SS52 Master/Slave“) und die Schaltfläche „Konfigurieren...“ sind wichtig.

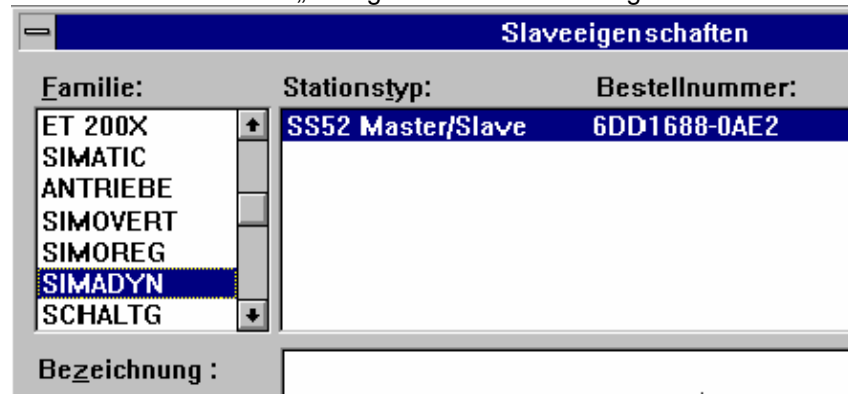


Bild 3-39 Dialogfeld „Slaveeigenschaften“

6. Bevor die Konfiguration jedoch beginnt, müssen in einem Dialogfeld „Master-Hostauswahl“ die angegebenen Einstellungen mit OK quittiert werden.

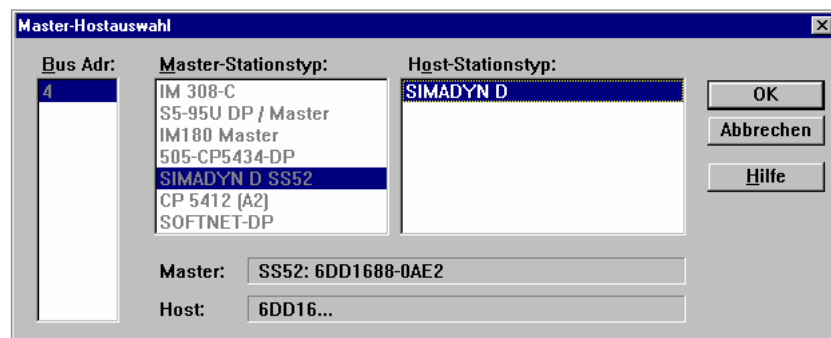


Bild 3-40 Dialogfeld „Master-Hostauswahl“

7. Danach erfolgt die eigentliche Konfiguration des Busteilnehmers. Für das Kommunikationsmodul CP50M0 ist dieses Konfigurationsfenster zunächst vollkommen leer. Es müssen jetzt die Nutzdatenstrukturen in die Liste im Dialogfeld „Konfigurieren: SIMADYN D Slave ...“ eingetragen werden.

#### HINWEIS

In diesem „Mastersystem <1>“ ist die S5 der Master, so dass Sende- und Empfangsbetrieb aus deren Sicht zu betrachten ist (I/O Adressen der S5).

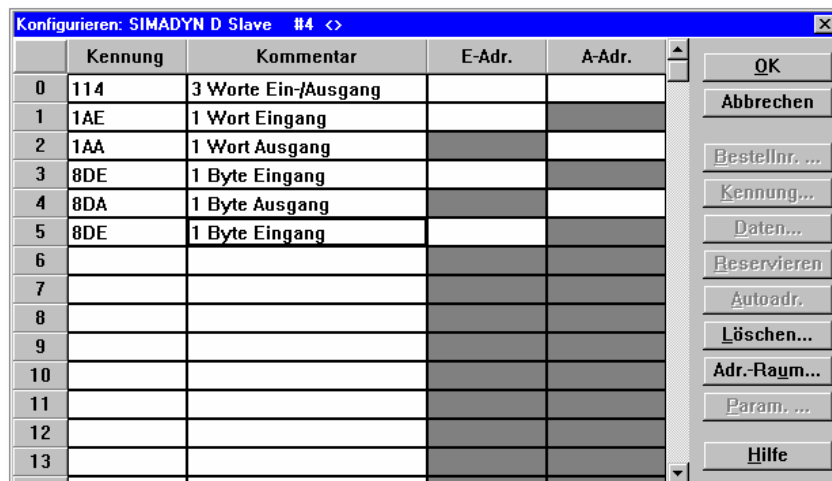


Bild 3-41 Dialogfeld „Konfigurieren: SIMADYN D Slave ...“

8. In der Spalte „Kennung“ werden alle Datenarten eingetragen. Dazu ist das zugehörige Dialogfenster zu aktivieren. Dies erreichen Sie entweder durch einen Doppelklick auf eine Zelle oder nach Markierung der Zelle und anschließender Auswahl der Schaltfläche „Kennung“. Folgende Parameter können angegeben werden:

- **Typ**  
Auswahl zwischen:
  - Eingang, Ausgang
  - Ein-/Ausgang
  - Leerplatz
  - Spezialformat
- **Länge**  
1 bis 16
- **Format**  
Auswahl zwischen Wort oder Byte



Bild 3-42 Dialogfeld „Kennung“

- Nach Beenden des Dialoges mit OK ist die entsprechende Kennung in der Liste eingetragen. Die Reihenfolge der Prozeßdaten im Telegramm wird durch die Position, an der die Kennung in den Ein- bzw. Ausgabeadressbereichen eingetragen ist, bestimmt (grau hinterlegte Felder bleiben unberücksichtigt). Einträge in die Spalte Kommentar sind optional und frei gestaltbar. Die Adresseinstellungen („E-Adr.“ und „A-Adr.“) werden für die SIMATIC TDC/SIMADYN D-Datenbasis nicht benötigt. Damit ist das erste Hostsystem erstellt, in dem SIMATIC TDC der S5 als Slave untergeordnet ist. Die Parametrierung dafür ist nun abgeschlossen. Zu beachten ist, dass es sich dabei um die Konfigurationsdaten für die IM308 (S5) handelt und diese damit nicht weiter bearbeitet werden müssen, weil sie für die CP50M0 nicht relevant sind.

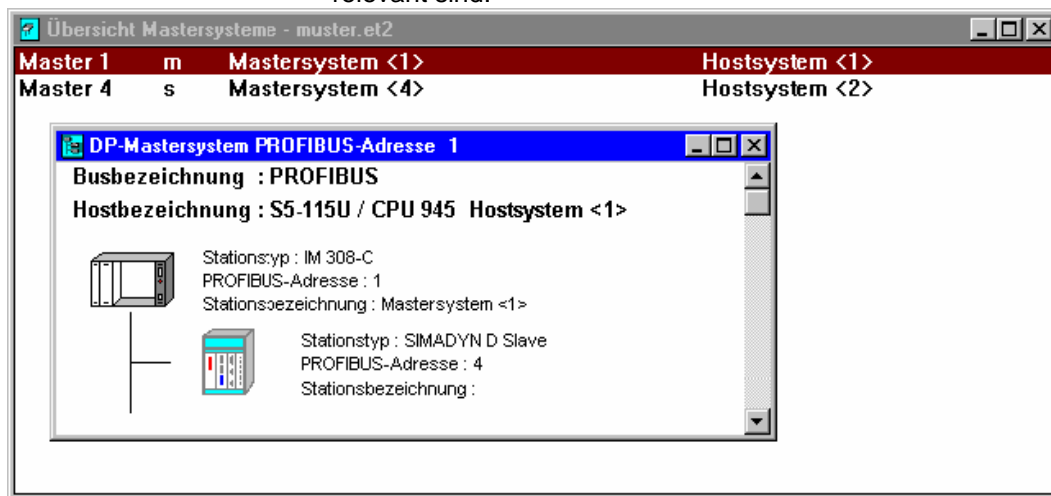


Bild 3-43 Fenster „DP -Mastersystem PROFIBUS-Adresse 1“

### Parametrierung des 2. Hostsystems

- Mit einem Doppelklick auf „Master 4“ wird das erste Hostsystem geschlossen und das zweite Hostsystem zur Parametrierung des CP50M0-Masters zugänglich gemacht.

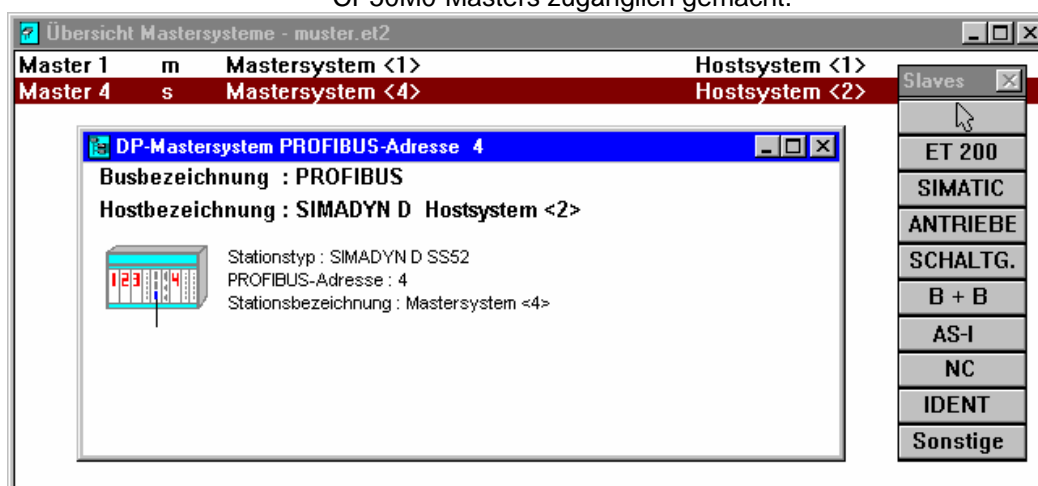


Bild 3-44 Dialogfeld „Übersicht Mastersysteme „

- Wie ein weiterer Doppelklick auf dem Symbol „SIMADYN D“ zeigt, ist es wichtig, zuerst das SS52 als Slave im „Hostsystem <1>“ einzurichten. Der komplette Telegrammaufbau wird automatisch in die Konfiguration der CP50M0 übernommen, mit dem Unterschied, dass die Telegramme vertauscht in den Adressbereichen liegen: der Ausgang der S5 wird zum Eingang für SIMATIC TDC und umgekehrt.

Die Datenkennung (über Konfigurieren...) ist jetzt grau hinterlegt und für diese Kommunikation aus dem jetzigen Hostsystem heraus nicht mehr veränderbar (Kennung und Kommentar gehören zur S5). Mit „OK“ werden die Angaben quittiert. Damit ist die Kommunikation zur S5 fertig eingerichtet.



Bild 3-45 Dialogfeld „Konfigurieren: SIMADYN D Slave“

- Die Masterfunktion des CP50M0-Moduls kann projektiert werden. Dazu muss zum Fenster DP-Mastersystem PROFIBUS-Adresse 4 zurückgekehrt werden. Nach einer erneuten Aktivierung des Slave-Menüs (der Mauszeiger ändert sich) wird mit einer durchgehenden Ankopplung von ET 200 U, ET 200 B und SIMOVERT Master Drive begonnen, wobei nach jedem Aufruf einer Komponente die PROFIBUS-Adresse abgefragt wird. Danach öffnet sich automatisch das Fenster „Slaveeigenschaften“, in dem, wie bereits beschrieben, über **Konfigurieren...** die nötigen Einstellungen vorgenommen werden können.
- Da es sich bei Feldgeräten um reine Slaves handelt, ist je nach Funktion, Bauweise und „Eigenintelligenz“ deren Parametrierung nur in eingeschränkter Form möglich.  
Die einzelnen Konfigurationen sehen wie folgt aus:
  - ET 200 U**  
Modularer Aufbau mit drei Ausgabemodulen (je 8 digitale Ausgänge) und einem Eingangsmodul (8 digitale Eingänge): Damit müssen drei Byte gesendet und ein Byte empfangen werden.

Konfigurieren: ET 200U #11 <>					
	Kennung	Bestellnummer	Kommentar	E-Adr.	A-Adr.
0	8DA		1 Byte Ausgang		
1	8DA		1 Byte Ausgang		
2	8DA		1 Byte Ausgang		
3	8DE		1 Byte Eingang		

Bild 3-46 Fenster „Konfigurieren“

– **ET 200 B**

Kompakte Bauweise mit acht digitalen Ausgängen und acht digitalen Eingängen: Je ein Byte im Sende- und Empfangstelegramm. Dabei sind die Kennungen durch die Baugruppenauswahl fest vorgegeben.

Konfigurieren: B-8DI/8DO DP #51 <>				
	Kennung	Kommentar	E-Adr.	A-Adr.
0	8DA	1 Byte Ausgang		
1	8DE	1 Byte Eingang		

Bild 3-47 Fenster „Konfigurieren“

– **SIMOVERT Master Drive**

Slave mit Eigenintelligenz: Je nach Einstellung am Umrichter sind fünf unterschiedliche Telegrammaufbauten (PPO-Typen) zugelassen. Diese müssen schon beim Projektieren bestimmt werden und lassen sich hinterher auch nicht mehr verändern. (Felder grau hinterlegt und damit inaktiv)

Konfigurieren: MASTER DRIVES CB1 #71 <>				
	Kennung	Kommentar	E-Adr.	A-Adr.
0	2AX	PZD 2 Words		

Bild 3-48 Fenster „Konfigurieren“

5. Nach Beenden des Konfigurierens sollte der Bildschirm wie folgt aussehen:

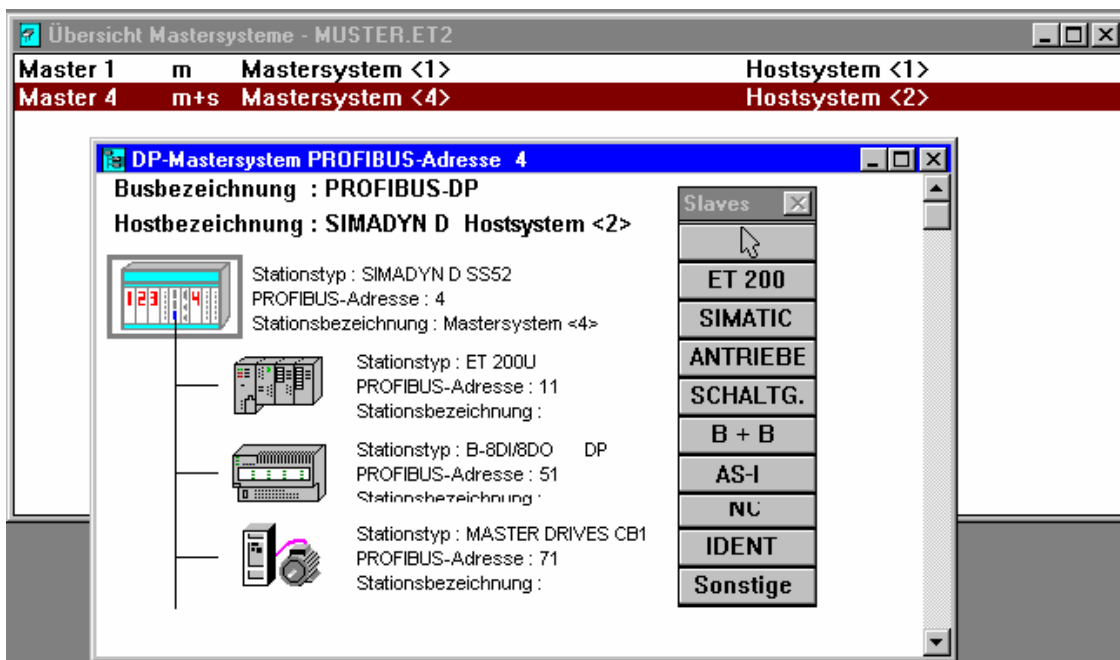


Bild 3-49 Fenster „DP-Mastersystem PROFIBUS-Adresse“

### Ändern der Konfiguration der Slaves

Die Konfigurationen der einzelnen Slaves können nachträglich verändert und angepasst werden.

1. Die jeweiligen Symbole in obiger Abbildung mit einem Doppelklick anwählen. Über das Fenster „Slaveeigenschaften“ gelangt man wieder zu den Konfigurationsdialogen.
2. Um die Parametrierung zu vervollständigen, müssen als letzter Schritt die Busparameter eingestellt werden. Unter **Projektieren > Busparameter...** wird ein Dialogfenster geöffnet, in dem für die Musterkonfiguration nur das Busprofil (PROFIBUS-DP) und die Baudrate von Bedeutung sind. Die Baudrate muss mit der in CFC angegebenen übereinstimmen und ist in unserem Beispiel durch die ET 200 U bzw. die Kommunikationsbaugruppe CB1 des SIMOVERT Master Drive begrenzt (aktuell: 1,5 Mbaud).

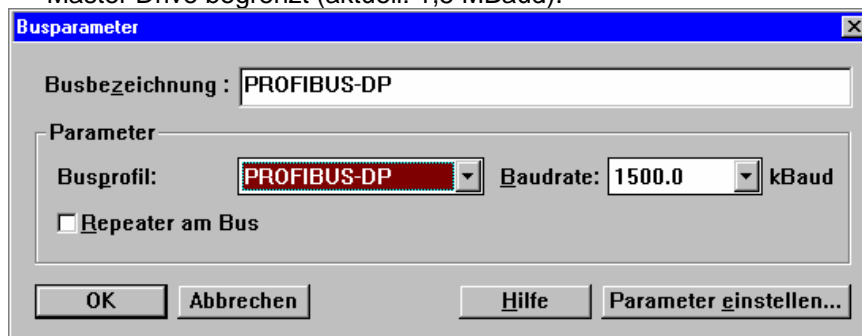


Bild 3-50 Dialogfeld „Busparameter“

3. Damit ist die Projektierung der CP50M0 für diese Musterkonfiguration beendet und kann abgespeichert werden.

### Übertragen der Projektierung in Speicher der CP50M0

Der nächste Schritt beim Einrichten der CP50M0-Konfiguration ist die Übertragung der Projektierung in den Speicher der entsprechenden Kommunikationsschnittstelle CP50M0. Hierfür stehen zwei Arten zur Verfügung:

- Transfer über eine zweite Baugruppensenchnittstelle (RS232), die sich auf dem gleichen 9-poligen Sub-D-Stecker befindet wie die RS485.
  - Der Transfer über RS232 kann über eine gewöhnliche PC-Schnittstelle (COM 1 oder COM 2) ausgeführt werden, wobei ein spezielles Transferprogramm namens „SS52load“ das Beschreiben der CP50M0-Speichers (Download) übernimmt.
  - Dieser Download erfordert das Dateiformat „2bf“, weshalb das markierte „Hostsystem <2>“ über den Menübefehl **Datei-Export > Binärdatei...** in das richtige Format konvertiert werden muss (die Hostsysteme sind bei diesem Vorgang getrennt zu behandeln). Die CP50M0-Konfigurationsdatei steht damit im Stammverzeichnis des COM PROFIBUS-Programms im Verzeichnis „\progdta“ für die Übertragung auf die Baugruppe bereit.
- Transfer über die PROFIBUS-Schnittstelle RS485 (wird direkt durch COM PROFIBUS unterstützt).
  - Der Transfer über RS485 wird nicht behandelt, weil hierfür eine spezielle PC-Schnittstellenkarte (z.B. CP 5411) benötigt wird.

#### 3.8.3.11 Download der COM-Datenbasis auf CP50M0

### Notwendige Hardware

Notwendige Hardware für Download der COM-Datenbasis auf CP50M0:

- RS232-Verbindung zwischen PC und SS52
  - Auf dem 9-pol. Sub-D-Stecker ist neben der RS485 eine zusätzliche Schnittstellenphysik RS232 integriert.  
**Weitere Informationen**  
zum Sub-D-Stecker siehe Handbuch „Regelsystem SIMATIC TDC, Hardware“.
  - Da es sich bei deren Pinbelegung um keinerlei Standard handelt, muss ein gekreuztes Kabel (TxD auf RxD) nach Plan selbst angefertigt werden.

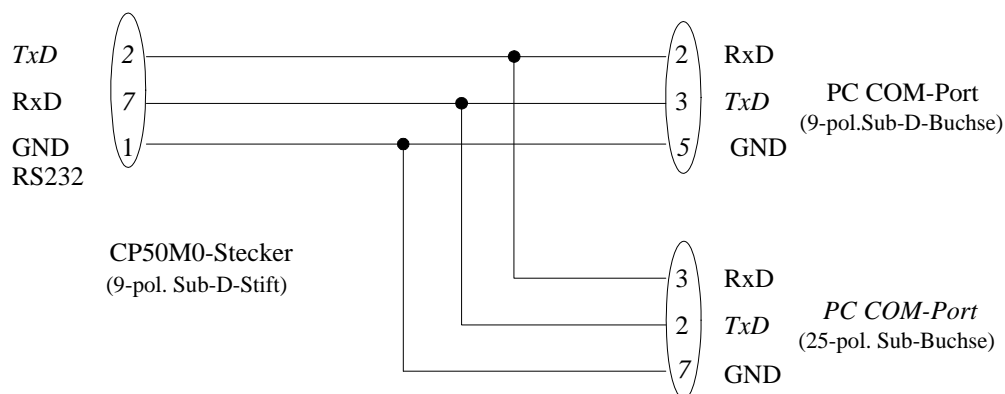


Bild 3-51 Schnittstellenbelegung RS232

### 3.8.3.12 Arbeiten mit dem Downloadtool „SS52load“

#### SS52load

SS52load ist in COM PROFIBUS (ab Version 3.1) integriert.

Die Benutzeroberfläche bietet folgende Funktionen:

- **Option-Comport:** Festlegung des zu verwendenden COM-Ports
- **Datei-Download:** Auswahl der gewünschten Datei und anschließender Download

### 3.8.3.13 Verhalten der CP50M0 während und nach dem Download

#### Allgemeines

Um den Download erfolgreich durchführen zu können, sollten die unterschiedlichen Verhaltensweisen von SIMATIC TDC bzw. des Kommunikationsmoduls CP50M0 vor, während und nach diesem Vorgang bekannt sein. Allgemeine Systemzustände werden über eine grüne und gelbe LED ausgegeben, die sich an jedem der zwei CP50M0-Schnittstelle befinden.

Diese LED's geben ausschließlich Auskunft darüber, ob SIMATIC TDC als abgeschlossenes System korrekt arbeitet bzw. wo eventuelle Fehler vorliegen. Busaktivitäten oder Kommunikationen mit anderen Busteilnehmern werden damit nicht ausgewertet.

#### Verhalten der LED's beim Anlauf von SIMATIC TDC

- Beim Einschalten der Spannung leuchten zunächst beide LED's kurz auf (ca. eine halbe Sekunde).
- Danach erlischt die gelbe LED wieder, so dass während der restlichen Hochlaufzeit die grüne (ca. fünf Sekunden) alleine leuchtet. Während dieser Zeit kann kein Download vorgenommen werden.
- Nach Beendigung der Anlaufphase wird der Betriebszustand der CP50M0 angezeigt.



## 3.9 Kopplung MPI

### 3.9.1 Eigenschaften und Hardware

**Eigenschaften** Bei SIMATIC S7/M7 ist MPI (Multi Point Interface) das Standard-Kommunikations-Protokoll. Der Datenaustausch erfolgt über einen Multi-Master-Bus mit maximal 126 Teilnehmern.

Bei SIMATIC TDC wird MPI außer zum Anschluss des CFC für Inbetriebnahme und Test einer Projektierung, auch zur Kommunikation mit WinCC und SIMATIC-OPs verwendet.

Mit der MPI-Kopplung werden die Kommunikations-Dienste Service (FB-SER) und S7-Kommunikation (FB S7OS) verwendet.

**Hardware** Für Kopplung MPI notwendige Hardware:

- Baugruppenträger
- CPU
- CP50M1- / CP50M0-Baugruppe entsprechende Schnittstelle in HWKonfig für MPI projiziert, bei CP50M1 nur X01)
- MPI-Leitung (ist im Lieferumfang des PG enthalten)

### 3.9.2 Projektierung

**HWKonfig** Im HWKonfig muß die Kommunikationsbaugruppe CP50M1 / CP50M0 und die entsprechende Schnittstelle als MPI projiziert werden. Bei ES ist die eigene MPI-Adresse anzugeben.

**Funktionsbaustein @MPI** Es muss genau ein Kopplungszentralbaustein @MPI pro Schnittstelle projiziert werden. Der Funktionsbaustein @MPI initialisiert und überwacht die MPI-Kopplung.

#### **Weitere Informationen**

zur Projektierung einer MPI-Kopplung siehe:

- Kap. „Kommunikations-Dienst Service“
- Kap. „Kommunikation mit SIMATIC Operator Panels“
- Kap. „Kommunikation mit WinCC“

## 3.10 Tabellenfunktion

### 3.10.1 Einleitung

Die Tabellenfunktion in SIMATIC TDC/SIMADYN D gibt dem Anwender die Möglichkeit, Tabellenwerte in eine Projektierung einzubinden und zu verwenden. Hierzu müssen auf SIMATIC TDC- bzw. SIMADYN D-Seite die Funktionsbausteine TAB und TAB\_D projektiert werden. Tabellenwerte des Datentyps REAL werden mit dem Funktionsbaustein TAB und des Datentyps DINT mit dem Funktionsbaustein TAB\_D verwaltet. Die Tabellenwerte werden dabei vom Anwender zur Verfügung gestellt.

Die Tabellenfunktion kann in drei Betriebsarten projektiert werden:

- **Handbetrieb:** d.h. die Tabellenwerte werden direkt über eine Online-Schnittstelle (z.B. CFC im Testmodus) am Baustein eingegeben oder mittels Teach-in aus dem Programm heraus an den Baustein weitergegeben. (Vgl. Bild 3-52)
- **Automatikbetrieb: Kommunikation**, d.h. die Tabellenwerte werden über eine Kommunikationsschnittstelle (TCP/IP, DUST1, S7 über P-Bus) übertragen. Um Tabellenwerte von einer S7-Steuerung über den P-Bus an eine SIMATIC FM 458 Applikationsbaugruppe zu übertragen, ist zusätzlich der Funktionsbaustein WR\_TAB auf S7-Steuerungsseite zu projektieren. (Vgl. Bild 3-53)
- **Automatikbetrieb: Speicherkarte**, d.h. die Tabellenwerte werden auf die Speicherkarte geladen und von dort ausgelesen.

---

#### HINWEIS

Es ist zu beachten, dass ein Wechsel der Betriebsarten nur zwischen „**Handbetrieb**“ und „**Automatikbetrieb: Kommunikation**“ sowie „**Handbetrieb**“ und „**Automatikbetrieb: Speicherkarte**“ möglich ist.

---

Sind die Tabellenwerte eingegeben bzw. übertragen, erfolgt eine Gültigkeitsprüfung. Die Adresse der Tabelle wird am Ausgang „TAB“ angezeigt.

Die Tabellenwerte werden doppelt, d.h. in zwei Tabellen verwaltet. Die als „gültig“ (=aktiv) definierte Tabelle wird für sämtliche Rechenoperationen der Projektierung verwendet. Die „ungültige“ (=inaktive) Tabelle dient zur Verwaltung von Wertänderungen. Sämtliche vom Anwender geänderten Tabellenwerte werden zunächst in die ungültige Tabelle übernommen. Wird die inaktive Tabelle aktiviert, werden die neuen Tabellenwerte in die zweite Tabelle gespiegelt. Die bis dahin aktive Tabelle verliert automatisch ihre Gültigkeit. Die neuen Tabellenwerte stehen damit in beiden Tabellen zur Verfügung.

Beide Tabellen können im batteriegepufferten SAVE-Bereich abgelegt werden, um einem Datenverlust vorzubeugen (Anschluss SAV=1 bei der Initialisierung).

**HINWEIS** Eine genaue Beschreibung der Funktionsbausteine TAB und TAB\_D findet sich in deren Online-Hilfe.  
Eine genaue Beschreibung des Funktionsbausteins WR\_TAB findet sich weiter unten im Kapitel „Funktionsbaustein WR\_TAB“.

### 3.10.1.1 Übersicht „Handbetrieb“

Die folgende Abbildung zeigt die prinzipielle Vorgehensweise im „Handbetrieb“:

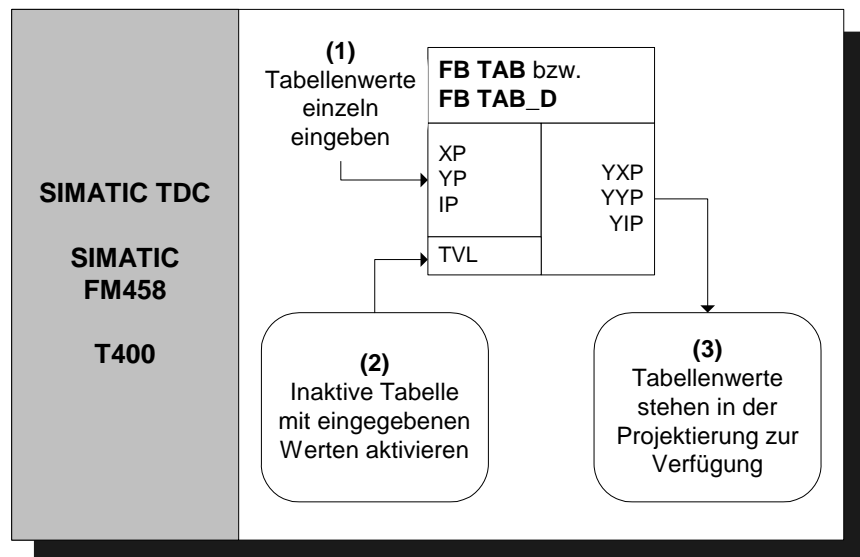


Bild 3-52 Prinzipielle Vorgehensweise im „Handbetrieb“

Eine ausführliche Beschreibung der Betriebsart „Handbetrieb“ befindet sich im Abschnitt „Handbetrieb“ (Seite 3-111)

### 3.10.1.2 Übersicht „Automatikbetrieb: Kommunikation“

Im „Automatikbetrieb: Kommunikation“ können die Tabellenwerte mit Hilfe der folgenden Kommunikationsvarianten übertragen werden:

- S7 über P-Bus für SIMATIC FM 458 (zusätzliche Projektierung des WR\_TAB auf Steuerungsseite nötig)
- TCP/IP (Tabellenwerte können mit Hilfe der Funktionsbausteine CTV und CRV auch von einer SIMATIC TDC Baugruppe zu anderen übertragen werden)
- DUST1 (Tabellenwerte können über eine DUST1-Schnittstelle übertragen werden)

Die Tabellenwerte werden durch Datentelegramme übertragen.  
 Die folgende Abbildung zeigt die prinzipielle Vorgehensweise im „Automatikbetrieb: Kommunikation“ für die Übertragung von Tabellen von einer S7-Steuerung zu einer SIMATIC FM 458 Applikationsbaugruppe via P-Bus:

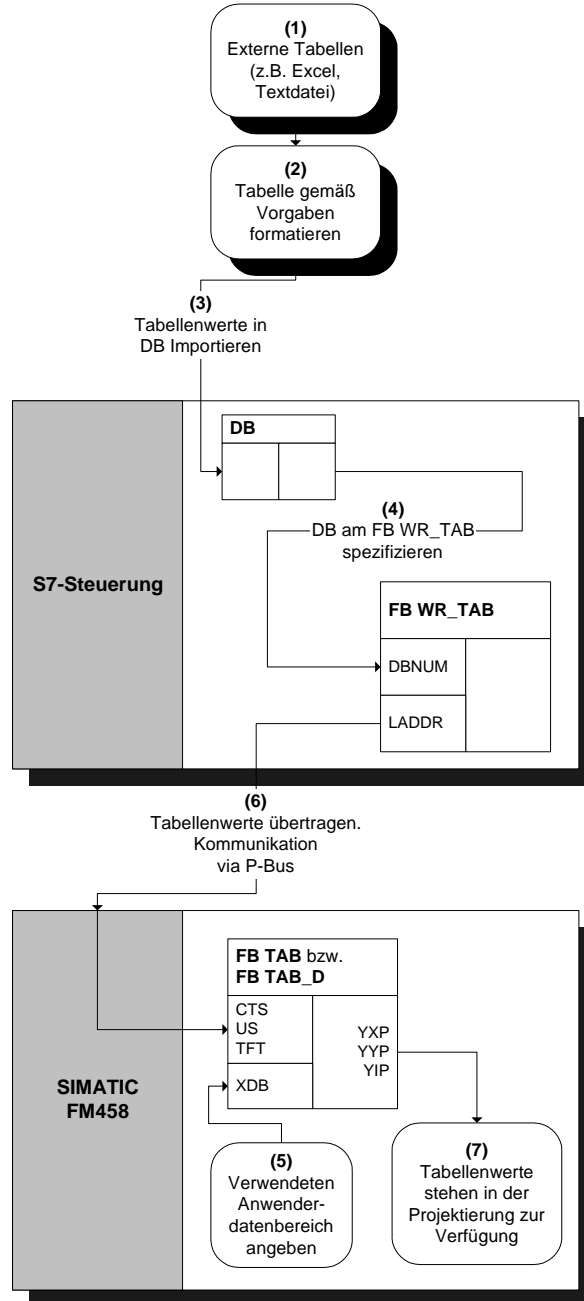


Bild 3-53 Prinzipielle Vorgehensweise beim „Automatikbetrieb: Kommunikation“ (via P-Bus)

Eine ausführliche Beschreibung der Betriebsart „Automatikbetrieb: Kommunikation“ für die Übertragung von Tabellen von einer S7-Steuerung zu einer SIMATIC FM 458 Applikationsbaugruppe befindet sich im Kapitel „Automatikbetrieb: Kommunikation“ (Seite 3-113).

### 3.10.1.3 Übersicht „Automatikbetrieb: Speicherkarte

Im „Automatikbetrieb: Speicherkarte“ können die Tabellenwerte mit Hilfe einer PC-Software (D7-SYS additionalComponentBuilder) auf die Speicherkarte geladen und bei der Initialisierung der Baugruppe übertragen werden.

Eine ausführliche Beschreibung der Betriebsart „Automatikbetrieb: Speicherkarte“ befindet sich im Kapitel „Automatikbetrieb: Speicherkarte“ (Seite 3-135).

### 3.10.1.4 Funktionsbaustein WR\_TAB

#### Symbol

		WR_TAB			
Bausteinaktivierung	BO	EN	TABEL	W	Anzahl der Datenblöcke, um kompletten DB-Inhalt zu übertragen
Anforderung zum Schreiben einer neuen Tabelle	BO	REQTAB	CNTTEL	W	Anzahl der bereits übertragenen Datenblöcke
Anforderung zum Schreiben der Tabellenwerte im Datenbaustein	BO	REQDB	STATUS	W	Aktueller Bearbeitungsstatus
Letzter Datenbaustein für Tabelle	BO	LASTDB	ERROR	W	ggf. Fehlermeldungen
Logische Adresse der Baugruppe	W	LADDR	DONE	B	Zustandsparameter DONE: Sendevorgang abgeschlossen
Datensatznummer für Schreib- und Lesedatensatz	BY	RECNUM			
Datenbausteinnummer	W	DBNUM			
TIMEOUT-Zeit für den Empfang der Quittungstelegramme vom FM-Modul	DW	TFT			

#### Kurzbeschreibung

Der Funktionsbaustein WR\_TAB dient der Übertragung von Tabellen von einer S7-Steuerung an eine SIMATIC FM 458 Applikationsbaugruppe. Die Tabellenwerte (zulässige Datentypen sind REAL und Double Integer) sind in einem Datenbaustein hinterlegt. Sie werden vom WR\_TAB an die Funktionsbausteine TAB bzw. TAB\_D auf der SIMATIC FM 458 Applikationsbaugruppe weitergeleitet, die die Tabellenwerte intern verwalten.

Der Funktionsbaustein WR\_TAB ist auf der Steuerungsseite zu projektieren. Die Übertragung der Tabellendaten erfolgt von einer S7-400 Steuerung über den P-Bus an eine SIMATIC FM 458 Applikationsbaugruppe. Es werden dabei immer alle Werte übertragen, die sich in dem am DBNUM-Eingang spezifizierten DB befinden.

**Anschlüsse**

In der folgenden Tabelle sind die einzelnen Anschlüsse, deren Datentypen und eine Anschlussbeschreibung aufgelistet:

Parameter	Deklaration	Datentyp	Beschreibung
REQTAB	INPUT	BOOL	REQTAB = 1: Anforderung zum Schreiben einer neuen Tabelle
REQDB	INPUT	BOOL	REQDB = 1: Anforderung zum Schreiben der Tabellenwerte, die im Datenbaustein abgelegt sind
LASTDB	INPUT	BOOL	Letzter DB für Tabelle
LADDR	INPUT	WORD	Logische Adresse der SIMATIC FM 458 Applikationsbaugruppe
RECNUM	INPUT	BYTE	Datensatznummer für Schreib- und Lesedatensatz
DBNUM	INPUT	WORD	Datenbausteinnummer, des DB, in dem die Tabellenwerte stehen.
TFT	INPUT	DWORD	TIMEOUT-Zeit für den Empfang von Quittungstelegrammen von der SIMATIC FM 458 Applikationsbaugruppe in ms.
TABTEL	OUTPUT	WORD	Anzahl der Datenblöcke, um kompletten DB-Inhalt zu übertragen
CNTTEL	OUTPUT	WORD	Anzahl der bereits zum FM-Modul übertragenen Datenblöcke
STATUS	OUTPUT	WORD	Zeigt den aktuellen Status der Bearbeitung / Übertragung an: 0: Tabellenübertragung ist inaktiv 1: Tabellenübertragung ist aktiv. Teilübertragung der Tabellenwerte aus einem DB erfolgt (Warten auf nächste Teilübertragung) 2: Übertragung der Tabellenwerte aus einem Datenbaustein ist noch nicht beendet.
ERROR	OUTPUT	WORD	Tritt während der Bearbeitung der Funktion ein Fehler auf, dann enthält der Rückgabewert einen Fehlercode
DONE	OUTPUT	BOOL	Zustandsparameter DONE=1: Sendevorgang abgeschlossen

Folgende Fehler können auftreten und werden am ERROR-Ausgang angezeigt:

Fehlercode	Erläuterung	Abhilfe
0xB210	OK	-
0xB211	Logische Adresse der Baugruppe ungültig	Angabe einer gültigen Baugruppenadresse am Eingang LADDR.
0xB212	Datensatznummer ungültig	Tabellenwerte in aufsteigender Reihenfolge im DB eintragen.
0xB213	Ungültiges Datenformat der Tabelle	Tabellenwerte müssen vom Datentyp REAL für den TAB und vom Datentyp DINT für den TAB_D sein.
0xB214	Datenformat des neuen Datensatzes passt nicht zu dem der bisher übertragenen Datensätze	Sicherstellen, dass alle Tabellenwerte dasselbe Datenformat besitzen.
0xB215	FM 458 antwortet nicht	Kommunikationsverbindung und Projektierung überprüfen.
0xB216	Tabelle ist zu groß	Übertragung der Tabelle in Teilübertragungen vornehmen, d.h. Tabellenwerte entweder auf mehrere DBs verteilen oder nach jeder Teilübertragung neue (weitere) Tabellenwerte in DB schreiben und übertragen.
0xB217	Tabelle ist nicht vollständig (X- / Y-Werte)	Tabelle vervollständigen, zu jedem X-Wert muss auch ein Y-Wert vorhanden sein.
0xB218	REQTAB während der Bearbeitung zurückgesetzt	Übertragung der Tabellenwerte neu durchführen.
0xB219	REQDB während der Bearbeitung zurückgesetzt	Übertragung der Tabellenwerte neu durchführen.
0xB21A	DB-Nummer ist ungültig	Angabe einer gültigen DB-Nummer.
0xB21B	TIMEOUT beim Empfang des Quittungstelegramms	Kommunikationsverbindung und Projektierung überprüfen. Übertragung der Tabellenwerte wiederholen.
0xB21C	Ungültiger Bearbeitungszustand	Projektierung des WR_TAB überprüfen.

Weiter werden gegebenenfalls Fehler des SFC58 bzw. SFC59 am ERROR-Ausgang angezeigt.

### 3.10.2 Handbetrieb

#### 3.10.2.1 Anwendung

Die Betriebsart „Handbetrieb“ stellt die einfachste Möglichkeit dar, Tabellenwerte in eine Projektierung einzufügen. Durch die manuelle Eingabe bzw. das Teach-in aus dem Programm heraus ist sie jedoch vergleichsweise zeitraubend.

#### Eingabe der Tabellenwerte

Nach korrekter Projektierung des TAB bzw. des TAB\_D können die Tabellenwerte nacheinander eingegeben werden. Zunächst ist die Tabellengröße, d.h. die Anzahl der Wertpaare (=Punkte) am Eingang NP anzugeben. Soll die Tabelle im SAVE-Bereich gespeichert werden, so muss der Eingang SAV des Funktionsbausteins auf 1 stehen.

Anschließend können die Tabellenwerte eingegeben werden. Hierzu ist als erstes der Index Punkt *i* am Eingang IP des einzugebenden Wertepaares anzugeben. Anschließend sind X- und Y-Wert des Punktes an den Eingängen XP und YP einzugeben. Um die eingegebenen Werte zu übernehmen, ist nach Eingabe jedes Wertepaares der Eingang WR von 0 auf 1 zu setzen. Vor der Eingabe des nächsten Punktes, ist der Index am Eingang IP heraufzusetzen. Dann sind die Werte für diesen Punkt einzugeben. Diese Prozedur wird solange wiederholt, bis alle Werte eingegeben sind.

Für die Eingabe der einzelnen Punkte muss keine bestimmte Reihenfolge beachtet werden.

Die Anzahl der eingegebenen Punkte muss mit der Angabe am Eingang NP übereinstimmen.

Sämtliche Eingaben während dieser Prozedur werden in die inaktive Tabelle des Funktionsbausteins übernommen und stehen erst nach Aktivierung in der Projektierung zur Verfügung. Um die inaktive Tabelle mit den eingegebenen Werten zu aktivieren, ist der Eingang TVL auf 1 zu setzen.

Weitere Änderungen können dann wieder in der inaktiven Tabelle vorgenommen werden und stehen erst nach erneuter Aktivierung zur Verfügung.

#### **Abfrage der Tabellenwerte**

Um die eingegebenen Tabellenwerte auszugeben, ist nach Beendigung der Eingabe am Eingang IP der Index des anzuzeigenden Punktes *i* anzugeben und der Eingang RD von 0 auf 1 zu setzen. Die Tabellenwerte des Punktes *i* werden dann an den Ausgängen YXP (X-Wert) und YYP (Y-Wert) angezeigt. Der Index des Punktes *i* wird am Ausgang YIP ausgegeben.

### **3.10.2.2 Projektierung**

Für die Betriebsart „Handbetrieb“ müssen nur der TAB und/oder TAB\_D projektiert werden, je nachdem ob Tabellenwerte des Datentyps REAL und/oder DINT verwaltet werden sollen. Jede Tabelle darf nur Werte eines Datentyps enthalten. Sollen mehrere Tabellen unterschiedlicher Datentypen verwaltet werden, so ist für jede Tabelle ein TAB bzw. TAB\_D zu projektieren.

Die Funktionsbausteine TAB und TAB\_D sollten in einer Abtastzeit größer gleich 32ms projektiert werden. Folgende Anschlusseinstellungen sind nötig:

- AUT** = 0 (Automatikbetrieb deaktiviert)
- NP** = [Angabe der Tabellengröße]
- XP** = [Eingabe der X-Werte]
- YP** = [Eingabe der Y-Werte]
- IP** = [Eingabe des zu ändernden Wertepaares]
- TVL** = 1 (für Aktivierung der Tabelle nach Eingabe aller Werte)
- WR** = 1 (für Übernahme des eingegebenen Wertepaares in Tabelle)
- RD** = 1 (für Anzeige des unter IP angegebenen Wertepaares an den Ausgängen YXP und YY)



---

<b>HINWEIS</b>	<p>Falls im „Handbetrieb“ der CTS-Anschluss bei der Initialisierung auf „0“ gesetzt ist (CTS=0; AUT=0), kann danach nicht mehr in den „Automatikbetrieb: Speicherkarte“ umgeschaltet werden (CTS=0; AUT=1).</p> <p>Wenn während der Initialisierung der CTS-Anschluss auf „0“ gesetzt ist und der „Automatikbetrieb: Speicherkarte“ aktiviert ist (AUT=1), kann anschließend auf „Handbetrieb“ (CTS=0; AUT=0) umgeschaltet werden. Die auf der Speicherkarte hinterlegte Tabelle kann dann im „Handbetrieb“ bearbeitet werden.</p> <p>Wird danach wieder auf „Automatikbetrieb: Speicherkarte“ umgeschaltet (CTS=0; AUT=1), hat dies keine Auswirkungen mehr, weil dieser nur während des Initialisierungsvorgangs aktiv ist.</p> <p>Ist am CTS-Anschluss eine Kommunikationsschnittstelle projektiert, kann beliebig zwischen „Handbetrieb“ und „Automatikbetrieb: Kommunikation“ gewechselt werden.</p>
----------------	---

---

### 3.10.3 Automatikbetrieb: Kommunikation

#### 3.10.3.1 Anwendung mit S7-Steuerung und SIMATIC FM 458 Applikationsbaugruppe

<b>Übertragung von Tabellenwerten</b>	<p>Folgende Voraussetzungen müssen für eine erfolgreiche Übertragung von Tabellen erfüllt sein:</p> <ul style="list-style-type: none"> <li>• In der FM 458 Applikationsbaugruppe müssen die Funktionsbausteine TAB und/oder TAB_D entsprechend den Projektierungsvorgaben für „Automatikbetrieb: Kommunikation“ projektiert sein. (Eine genaue Erläuterung hierfür findet sich weiter unten im Kapitel „Projektierung für S7-Steuerung und SIMATIC FM 458 Applikationsbaugruppe“.)</li> <li>• Die X- und Y-Werte einer in einem DB befindlichen Tabelle müssen immer abwechselnd vorliegen. Zu jedem X-Wert muss ein Y-Wert existieren, so dass die Anzahl der Werte in einem Datensatz immer geradzahlig ist.</li> </ul> <p>Um eine Übertragung zu starten, müssen die Eingänge REQTAB und REQDB am WR_TAB auf 1 gesetzt werden. Anschließend werden die Tabellenwerte des am Eingang DBNUM am WR_TAB spezifizierten DBs übertragen.</p> <p>Am CNTTEL-Ausgang des WR_TAB wird dabei immer die aktuelle Anzahl der übertragenen Datenblöcke angezeigt.</p> <p>Am TABTEL-Ausgang des WR_TAB wird die Anzahl der Datenblöcke angezeigt, die benötigt wird, bis der gesamte Inhalt des DBs zur SIMATIC FM 458 Applikationsbaugruppe übertragen ist.</p>
---------------------------------------	--

Sind in dem angegebenen DB die Tabellenwerte vollständig vorhanden oder handelt es sich um die letzte Teilübertragung einer Tabelle, die nicht vollständig in einen DB „passt“, ist vor Beginn der Übertragung der Eingang LASTDB des WR\_TAB auf 1 zu setzen. Damit wird der SIMATIC FM 458 Applikationsbaugruppe das Ende der Übertragung signalisiert. Der STATUS-Ausgang des WR\_TAB wechselt anschließend von 2 auf 0.

---

**HINWEIS**

Es werden immer alle Tabellenwerte, die sich in dem am DBNUM-Eingang des WR\_TAB angegebenen DB befinden übertragen.

---

**Tabelle zu groß für einen DB**

Falls die Tabelle für einen Datenbaustein zu groß ist, so ist die Übertragung der Tabellenwerte in einzelne Teilübertragungen zu zerlegen. Dabei ist folgendermaßen vorzugehen:

Zunächst wird der erste Tabellenteil in den DB geschrieben und wie oben beschrieben übertragen. Der Eingang LASTDB des WR\_TAB bleibt auf 0. Der STATUS-Ausgang des WR\_TAB steht während der Übertragung auf 2 und wechselt am Ende der Teilübertragung von 2 auf 1.

Anschließend sind die alten Tabellenwerte im DB mit den nachfolgenden Tabellenwerten zu überschreiben. Ist dies erledigt, so ist am WR\_TAB für die Aktivierung der nächsten Teilübertragung der REQDB-Eingang erneut von 0 auf 1 zu setzen.

Diese Prozedur ist so oft zu wiederholen, bis sämtliche Tabellenwerte übertragen sind.

Bei der letzten Teilübertragung ist der Eingang LASTDB des WR\_TAB von 0 auf 1 zu setzen. Damit wird der SIMATIC FM 458 Applikationsbaugruppe das Ende der Übertragung signalisiert. Der STATUS-Ausgang des WR\_TAB wechselt anschließend von 2 auf 0.

---

**HINWEIS**

Steht genug Anwendungsspeicher zur Verfügung, so kann die Tabelle auch in mehreren unterschiedlichen DBs hinterlegt werden. In diesem Fall ist bei jeder Teilübertragung nur die jeweils passende DB-Nummer am Eingang DBNUM des WR\_TAB anzugeben. Es ist jedoch zu beachten, dass die DBs in der richtigen Reihenfolge übertragen werden, so dass sämtliche Tabellenwerte in aufsteigender Folge übertragen werden.

---

**Übertragungsdauer**

Die Zeitdauer für die Übertragung der Tabellenwerte hängt von folgenden Faktoren ab:

- Anzahl der Tabellenwerte
- Größe der Datenblöcke
- Abtastzeit des TAB bzw. TAB\_D
- Bearbeitungszeit WR\_TAB

In jedem Zyklus wird ein Telegramm mit 56 Tabellenwerten von der Steuerung zur SIMATIC FM 458 Applikationsbaugruppe übertragen.

Die Übertragungsdauer einer Tabelle kann dann wie folgt berechnet werden:

$$\text{Zeitdauer für Übertragung} = \left[ \frac{\text{Anzahl der Tabellenwerte}}{56} \right] \cdot \text{Zykluszeit langsamer FB} \\ \text{(d.h. TAB, TAB\_D oder WR\_TAB)}$$

Die Übertragungsdauer der Daten über den P-Bus ist für diese Abschätzung nicht relevant, da diese Übertragungsdauer in der Regel kleiner 1ms ist und die Funktionsbausteine TAB bzw. TAB\_D in der Regel in Abtastzeiten größer 32ms projektiert werden.

Falls eine Tabelle auf mehrere Datenbausteine verteilt ist, erhöht sich der Zeitbedarf. Denn zusätzlich zu der Übertragungsdauer der Tabellenwerte, die gemäß der obigen Formel ermittelt werden kann, sind die oben beschriebenen manuellen Änderungen durch den Anwender nötig.

### 3.10.3.2 Projektierung für S7-Steuerung und Applikationsbaugruppe FM 458

Für die Verbindung zwischen einer S7-Steuerung und einer SIMATIC FM 458 Applikationsbaugruppe via P-Bus müssen folgende Funktionsbausteine projektiert werden:

- SIMATIC FM 458 Applikationsbaugruppe:
  - TAB (für Datentyp REAL) und/oder
  - TAB\_D (Datentyp DINT)
  - @CPB (P-Bus-Kopplung Zentralbaustein)
- S7-Steuerung:
  - WR\_TAB

Jede Tabelle darf nur Werte eines Datentyps enthalten. Sollen mehrere Tabellen unterschiedlicher Datentypen verwaltet werden, so ist für jede Tabelle ein TAB bzw. TAB\_D zu projektieren.

Der WR\_TAB dient der Übertragung der Tabellenwerte vom SIMATIC DB zu den Funktionsbausteinen TAB bzw. TAB\_D. Die Tabellenwerte werden durch Datentelegramme übertragen. Mit dem letzten übertragenen Datentelegramm erhält der TAB bzw. der TAB\_D automatisch die Information, dass alle Tabellenwerte übertragen wurden, und dass die Tabelle aktiviert werden soll. Der WR\_TAB erhält eine Rückmeldung, ob die Aktivierung erfolgreich war oder nicht. Nach erfolgreicher Aktivierung der Tabelle wird ihre Adresse am TAB-Ausgang des TAB bzw. des TAB\_D ausgegeben.

### TAB bzw. TAB\_D

Die TAB bzw. TAB\_D sind folgendermaßen zu projektieren:

Sie sollten in einer Abtastzeit größer gleich 32ms projektiert werden. Folgende Anschlusseinstellungen sind nötig:

<b>CTS =</b>	[Name der projektierten Kommunikationsschnittstelle]
<b>AUT =</b>	1 (Automatikbetrieb aktiviert)
<b>US =</b>	[Kanalname.Adressstufe1] (Adressangaben zum Empfang)
<b>MOD =</b>	[Übertragungsmodus] (H=Handshake; R=Refresh; S=Select; M=Multiple)
<b>TFT =</b>	[Überwachungszeit in Millisekunden] (Maximale Telegrammausfallzeit während des Empfangs von Tabellenwerten)
<b>NP =</b>	[Angabe der maximalen Tabellengröße]

---

### HINWEIS

Ist am CTS-Anschluss eine Kommunikationsschnittstelle projektiert, kann beliebig zwischen „Automatikbetrieb: Kommunikation“ und „Handbetrieb“ gewechselt werden.

---

### WR\_TAB

Am WR\_TAB sind folgende Anschlusseinstellungen zu projektieren:

<b>LADDR =</b>	[Angabe der logischen Adresse der SIMATIC FM 458 Applikationsbaugruppe]
<b>RECNUM =</b>	[Angabe der Datensatznummer für Schreib- und Lesekanal. Muss Identisch mit „Adressstufe1“ am US-Anschluss des TAB bzw. TAB_D sein.]
<b>DBNUM =</b>	[Angabe der Datenbausteinnummer]

### 3.10.3.3 Tabellenwerte in Datenbaustein einfügen

Um Tabellenwerte an eine SIMATIC FM 458 Applikationsbaugruppe übertragen zu können, müssen sie in einem Datenbaustein (DB) zur Verfügung stehen. Der DB ist auf Steuerungsseite zu programmieren.

Um einen DB mit den gewünschten Tabellenwerten zu erzeugen, stehen zwei Möglichkeiten zur Verfügung:

- Erzeugen eines neuen DBs in STEP7 und manuelle Eingabe der Tabellenwerte in der Applikation „KOP/AWL/FUP“
- Tabellenwerte aus bereits vorhandener Tabelle (z.B. MS Excel) als externe Quelle in STEP7 importieren

### 3.10.3.3.1 Tabellenwerte manuell eingeben

Hierbei handelt es sich um die einfachste Methode, Tabellenwerte in einem DB zur Verfügung zu stellen. Sie besteht darin, die Anfangs- und Aktualwerte der einzelnen Tabellenwerte manuell in einem neu erzeugten DB in der Applikation „KOP/AWL/FUP“ einzugeben. Die dafür nötigen Schritte werden im folgenden beschrieben.

#### HINWEIS

Der Anfangswert ist ein für jeden Tabellenwert beliebig festlegbarer Wert. Er wird nur verwendet, wenn für den betreffenden Tabellenwert kein Aktualwert angegeben ist.

Der Aktualwert ist derjenige Wert, der in der Projektierung als Tabellenwert zur Verfügung gestellt wird. Die gewünschten Tabellenwerte sind hier anzugeben.

#### (1) Erzeugen eines neuen DBs unter STEP7

Als erstes ist unter STEP7 ein neuer DB zu erzeugen. Dazu wird der Ordner „Bausteine“ im entsprechenden S7-Programm markiert und im kontextsensitiven Menü der Eintrag „Neues Objekt einfügen → Datenbaustein“ angewählt.

Das folgende Bild zeigt die Vorgehensweise:

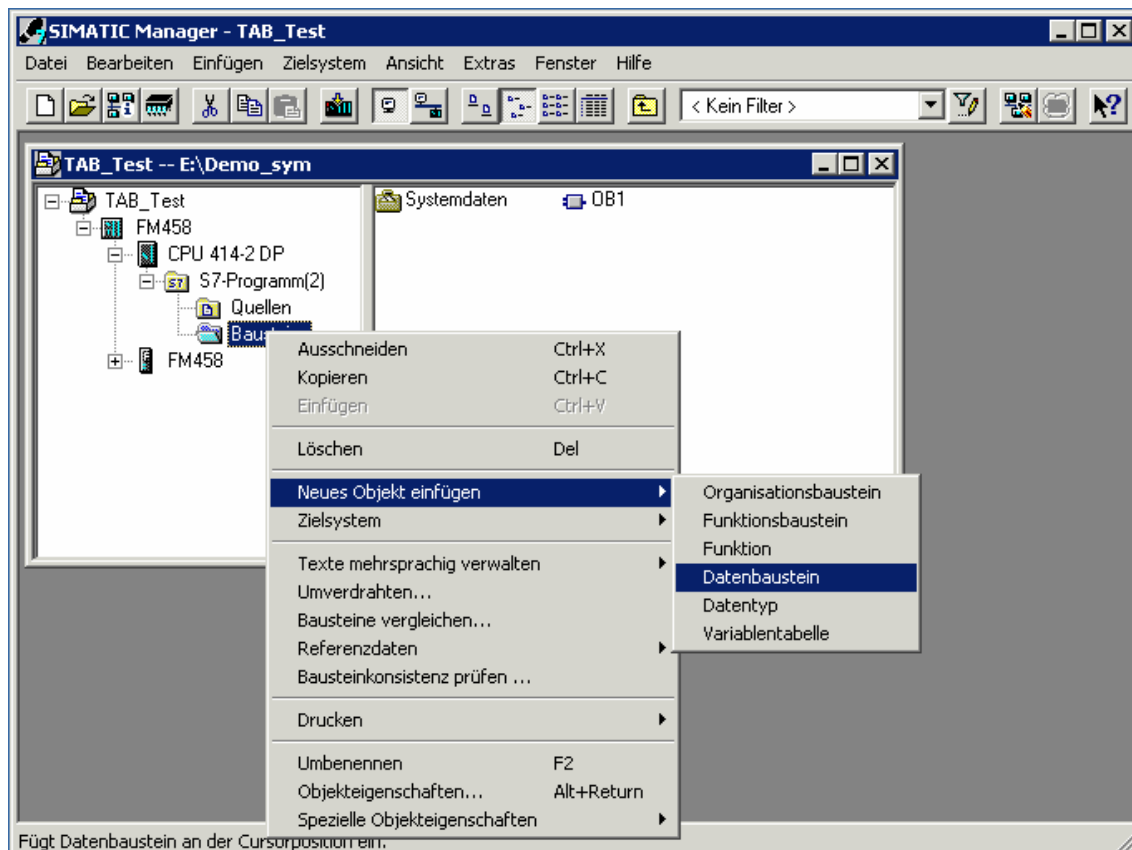


Bild 3-54 Neuen Datenbaustein unter STEP7 erzeugen

## (2) Öffnen des neuen DBs

Der nächste Schritt besteht darin, den neu erzeugten DB per Doppelklick mit der Applikation „KOP/AWL/FUP“ zu öffnen. Erstellungswerkzeug ist „DB-Editor“ und erstellt wird nur ein „Datenbaustein“.

Das folgende Bild zeigt die Auswahl beim Öffnen des neuen DBs:

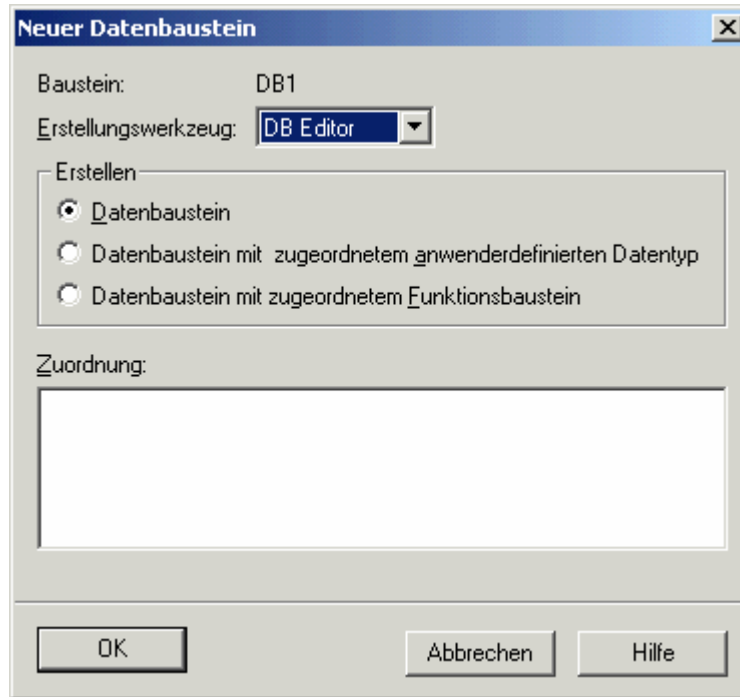


Bild 3-55 Auswahl bei der Erzeugung eines neuen DBs

Das folgende Bild zeigt den geöffneten neuen DB:

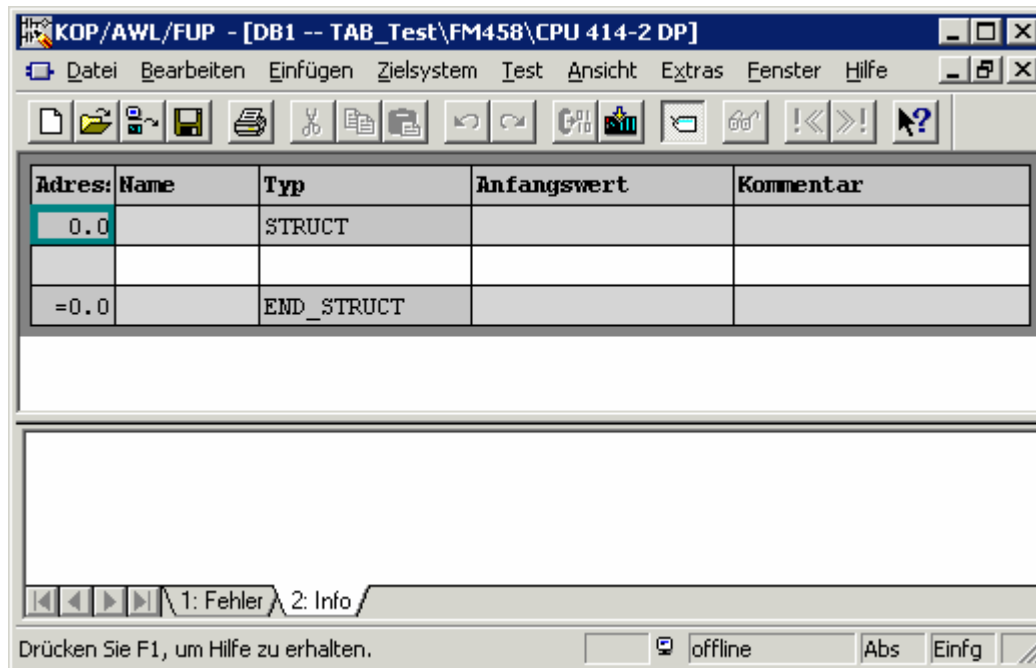


Bild 3-56 Neu erzeugter DB in Applikation „KOP/AWL/FUP“

### (3) Eingabe der Tabellenwerte

Nun können die gewünschten Tabellenwerte eingegeben werden. Dabei ist darauf zu achten, dass X- und Y-Werte jeweils im Wechsel eingegeben werden.

Als erstes ist der in der Tabelle verwendete Datentyp einzugeben (REAL oder DINT). Name ist dabei immer „Datatype“, Typ „WORD“ und Anfangswert für Datentyp REAL „W#16#1“, für Datentyp DINT „W#16#2“. Anschließend sind für jeden einzelnen Tabellenwert jeweils Name, Datentyp (Spalte „Typ“) und Wert (Spalte „Anfangswert“) einzugeben.

Das folgende Bild zeigt die Vorgehensweise bei der Eingabe von Tabellenwerten des Datentyps REAL:

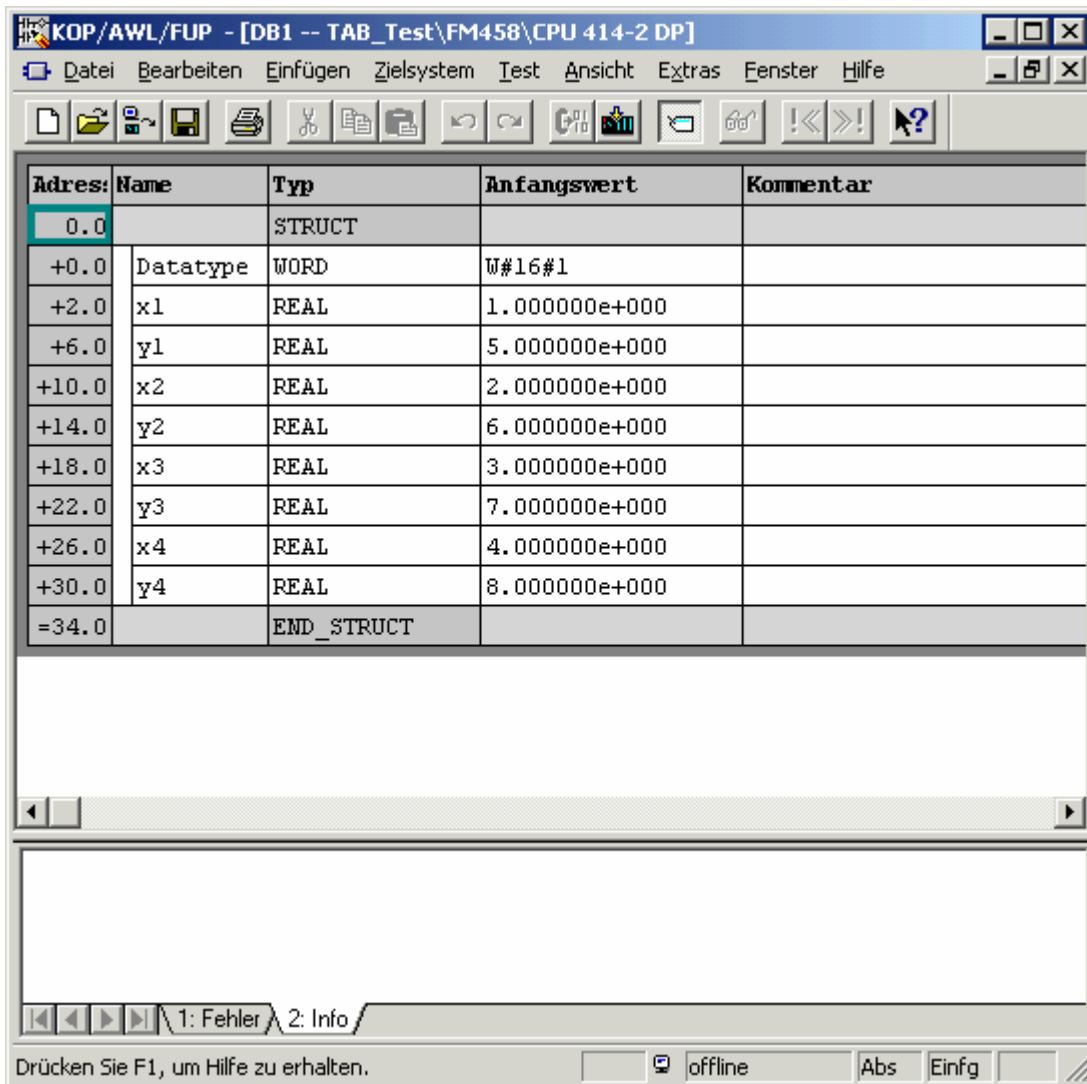


Bild 3-57 Manuell eingegebene Tabellenwerte in Applikation „KOP“AWL/FUP“

**HINWEIS**

Da in einer Tabelle nur Werte desselben Datentyps vorkommen dürfen, besteht eine effektive Art der Eingabe darin, eine ARRAY anzugeben. Auf diese Weise muss nicht jedes Mal der Datentyp angegeben werden.

Zur Vorgehensweise bei der Erzeugung von Eingaben des Typs ARRAY siehe die Online-Hilfe der Applikation „KOP/AWL/FUP“, insbesondere „Hilfe zu AWL“.

**(4) Speichern des DBs**

Nach der vollständigen Eingabe der Tabellenwerte kann der DB unter „Datei → Speichern“ gespeichert werden. Die Tabellenwerte stehen dann im DB für die Übertragung zur Verfügung.



### 3.10.3.3.2 Tabellenwerte importieren

Die im DB zur Verfügung zu stellenden Tabellenwerte können auch aus einer externen Quelle, z.B. einer MS Excel Tabelle, importiert werden. Für den fehlerfreien Import sind jedoch folgende Punkte zu beachten:

- Die Quelldatei der Tabelle muss eine bestimmte Formatierung aufweisen
- Die Quelldatei ist als externe Quelldatei unter STEP7 einzubinden
- Anhand der externen Quelldatei wird ein neuer DB erzeugt

Die für den Importvorgang nötigen Punkte bzw. Schritte werden im folgenden erläutert.

#### Tabellenformat

- Um eine bereits vorhandene (z.B. mit Excel erstellte) Tabelle in den DB importieren zu können, muss sie bestimmten Formatierungsvorschriften genügen:
- Die Tabelle muss einen Header enthalten, der Informationen über den Namen des DBs und die Version enthält.
- Als nächstes sind Informationen über die Struktur und den Datentyp der Tabellenwerte anzugeben.
- Anschließend erfolgt die Angabe der Tabellenwerte (als Anfangswerte).
- Es ist darauf zu achten, dass X- und Y-Werte immer abwechselnd anzugeben sind.
- Die Tabelle ist mit der Endung \*.AWL zu speichern.
- Anschließend kann die Tabelle als externe Quelldatei verwendet werden.

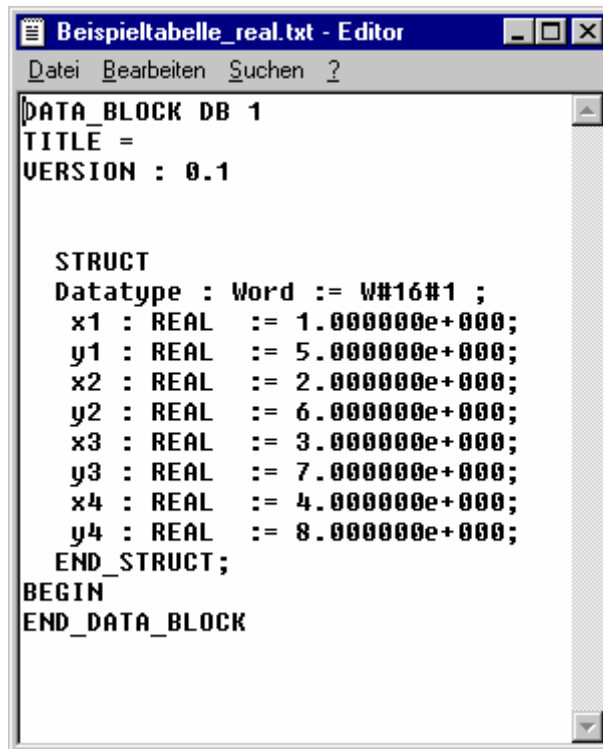
---

#### HINWEIS

Der *Anfangswert* ist ein für jeden Tabellenwert beliebig festlegbarer Wert. Er wird nur verwendet, wenn für den betreffenden Tabellenwert kein Aktualwert angegeben ist. Die Tabellenwerte werden ausschließlich als *Anfangswerte* definiert. *Aktualwerte* werden nicht verwendet. Dadurch werden die Dateigröße und damit der erforderliche Speicherbedarf erheblich vermindert.

---

Die folgende Abbildung zeigt ein Beispiel einer Tabelle mit je vier X- und Y-Werten des Datentyps REAL:



```
Beispieltable_real.txt - Editor
Datei Bearbeiten Suchen ?

DATA_BLOCK DB 1
TITLE =
VERSION : 0.1

STRUCT
Datatype : Word := W#16#1 ;
  x1 : REAL := 1.000000e+000;
  y1 : REAL := 5.000000e+000;
  x2 : REAL := 2.000000e+000;
  y2 : REAL := 6.000000e+000;
  x3 : REAL := 3.000000e+000;
  y3 : REAL := 7.000000e+000;
  x4 : REAL := 4.000000e+000;
  y4 : REAL := 8.000000e+000;
END_STRUCT;
BEGIN
END_DATA_BLOCK
```

Bild 3-58 Beispieltable mit Werten des Datentyps REAL

Die folgende Abbildung zeigt ein Beispiel einer Tabelle mit je zwei X- und Y-Werten des Datentyps DINT:

```

DATA_BLOCK DB 1
TITLE =
VERSION : 0.1

STRUCT
Datatype := W#16#2 ;
x1 : DINT := L#123456;
y1 : DINT := L#789012;
x2 : DINT := L#654321;
y2 : DINT := L#210987;
END_STRUCT ;
BEGIN
END_DATA_BLOCK

```

Bild 3-59 Beispieltable mit Werten des Datentyps DINT

### Von Excel zu AWL

Die folgenden Abschnitte erläutern beispielhaft die Vorgehensweise bei der Umformatierung einer Excel-Tabelle zum erforderlichen Tabellenformat.

Die in der folgenden Abbildung gezeigte Beispieldatei wird Schritt für Schritt entsprechend den Vorgaben des erforderlichen Tabellenformats formatiert.

	A	B	C
1	x-Wert	y-Wert	
2	1	5	
3	2	6	
4	3	7	
5	4	8	
6			
7			
8			
9			
10			
11			
12			
13			

Bild 3-60 Beispieltable in MS Excel

### (1) Header

Als erstes wird der erforderliche Header eingefügt. Hierzu werden am Anfang 5 Zeilen eingefügt und folgende Informationen eingegeben:

- DATA\_BLOCK DB 1 [Nummer des DBs]
- TITLE = [nach Bedarf eingeben]
- VERSION : 0.1 [Versionsangabe]

Die folgende Abbildung zeigt die Excel-Tabelle mit eingefügtem Header:

	A	B	C
1	DATA_BLOCK DB 1		
2	TITLE =		
3	VERSION : 0.1		
4			
5			
6	x-Wert	y-Wert	
7	1	5	
8	2	6	
9	3	7	
10	4	8	
11			
12			
13			

Bild 3-61 Beispieltable in MS Excel mit eingefügtem Header

### (2) Struktur und Tabellenwerte einfügen

Als nächstes werden die Struktur der Tabellenwerte und die Werte mit Angabe der Datentypen eingefügt. Hierzu werden für jedes Wertepaar zwei Zeilen plus eine Anfangs- und Endzeile eingefügt. Außerdem wird eine Zeile am Anfang für die Angabe des verwendeten Datentyps eingefügt.

Der Beginn der Strukturangaben wird in der Anfangszeile durch den Eintrag „STRUCT“ angezeigt. In der folgenden Zeile wird der in der Tabelle verwendete Datentyp angegeben („W#16#1“ für Datentyp REAL, „W#16#2“ für Datentyp DINT).

Anschließend erfolgen die Strukturangaben und Tabellenwerte für die einzelnen Wertepaare, X- und Y-Wert immer abwechselnd. Die Tabellenwerte sind entsprechend dem verwendeten Datentyp (hier REAL) anzugeben. Das Ende der Strukturangaben wird in der Schlusszeile mit dem Eintrag „END\_STRUCT;“ angezeigt.

Abschließend sind nur noch die Angaben für den Datenteil der Aktualwerte anzugeben („BEGIN“ und „END\_DATA\_BLOCK“). Da die Tabellenwerte bereits in den Anfangswerten der Strukturangaben enthalten sind, kann auf die Angabe von einzelnen Aktualwerten verzichtet werden.

Die folgende Abbildung zeigt die Excel-Tabelle mit eingefügten Strukturangaben und Tabellenwerten:

	A	B	C
1	DATA_BLOCK DB 1		
2	TITLE =		
3	VERSION : 0.1		
4			
5			
6	STRUCT		
7	Datatype := W#16#1 ;		
8	x1 : REAL := 1.000000e+000;		
9	y1 : REAL := 5.000000e+000;		
10	x2 : REAL := 2.000000e+000;		
11	y2 : REAL := 6.000000e+000;		
12	x3 : REAL := 3.000000e+000;		
13	y3 : REAL := 7.000000e+000;		
14	x4 : REAL := 4.000000e+000;		
15	y4 : REAL := 8.000000e+000;		
16	END_STRUCT;		
17	BEGIN		
18	END_DATA_BLOCK		
19			
20			
21			
22			

Bild 3-62 Beispieltable in MS Excel mit eingefügten Strukturangaben und Tabellenwerten

### (3) Als AWL-Datei speichern

Als letztes ist die korrekt formatierte Datei nur noch als Textdatei mit der Endung **.AWL** zu **speichern**. **Hierzu ist in MS Excel „Datei → Speichern unter...“ anzuwählen. Dort ist als Dateityp „Formatierter Text (Leerzeichen getrennt) (.prn)“ auszuwählen** und die Beispieltable unter frei zu wählenden Namen und Ort zu speichern.

Die folgende Abbildung zeigt das „Speichern unter“-Fenster in MS Excel mit der entsprechenden Auswahl:

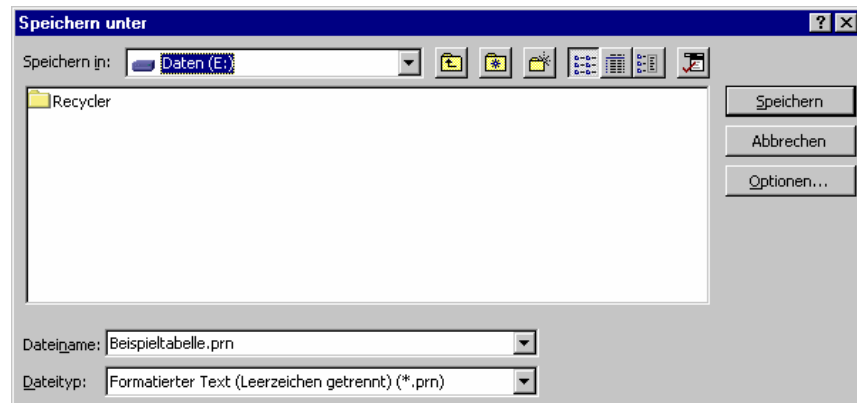


Bild 3-63 Beispieltable in MS Excel als Textdatei (\*.prn) speichern

Nach dem Speichern der Datei ist noch der Dateityp von \*.prn auf \*.awl abzuändern. Diese Datei kann dann mit einem beliebigen Texteditor geöffnet werden.

Die folgende Abbildung zeigt die Beispieltable als AWL-Datei, geöffnet im Windows-Standard-Texteditor:

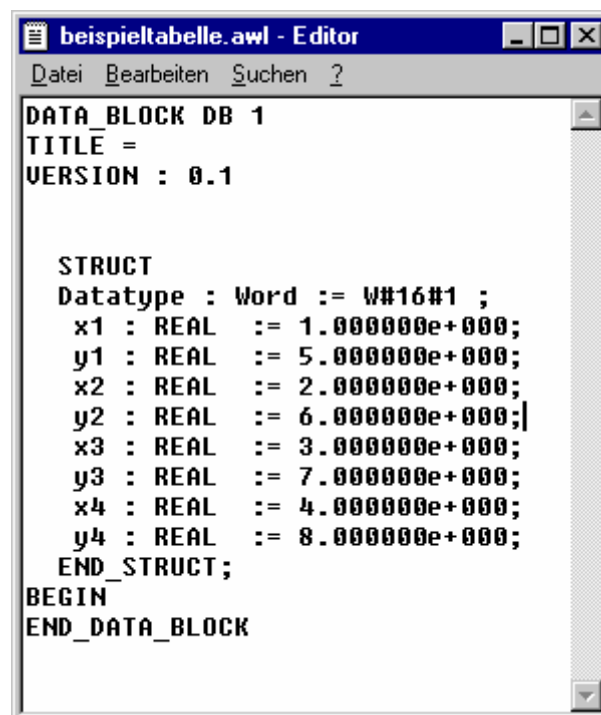


Bild 3-64 Gespeicherte Beispieltable als \*.awl-Datei im Texteditor geöffnet

Diese Datei kann nun als externe Quelldatei in STEP7 für einen DB dienen.

## Einbinden der Tabelle als Quelldatei

Anhand der oben erstellten Beispieldatei „BEISPIELTABELLE.AWL“ werden die einzelnen Schritte zur Einbindung einer extern erstellten Tabelle in einen DB erläutert.

### HINWEIS

Neben der Angabe der Tabellenwerte ist besonders die Angabe des Namens des DBs zu beachten. Anhand des in der Datei angegebenen Namens wird später ein DB erzeugt. In der obigen Beispieldatei ist als DB-Name in der ersten Zeile „DB1“ angegeben. (vergleiche Bild 3-61)

In der STEP7-Projektierung ist im S7-Programm unter „Quellen“ nun eine externe Quelle einzufügen. Nach Anwahl von „Quellen“ kann über einen rechten Mausklick im rechten Teilfenster das kontextsensitive Menü aufgerufen werden. Hier ist eine externe Quelle als neues Objekt einzufügen.

Das folgende Bild zeigt die Vorgehensweise:

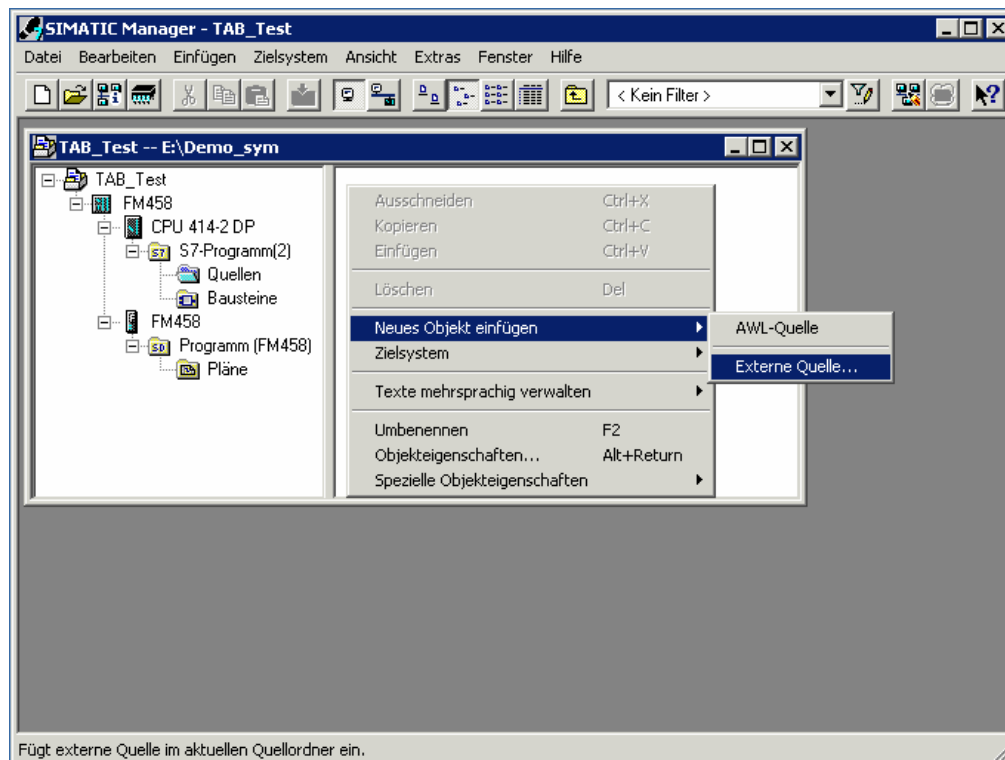


Bild 3-65 Externe Quelle in STEP7 einfügen

Als Quelldatei wird die oben erzeugte AWL-Datei gewählt. Das folgende Bild zeigt das Dateiauswahlfenster:

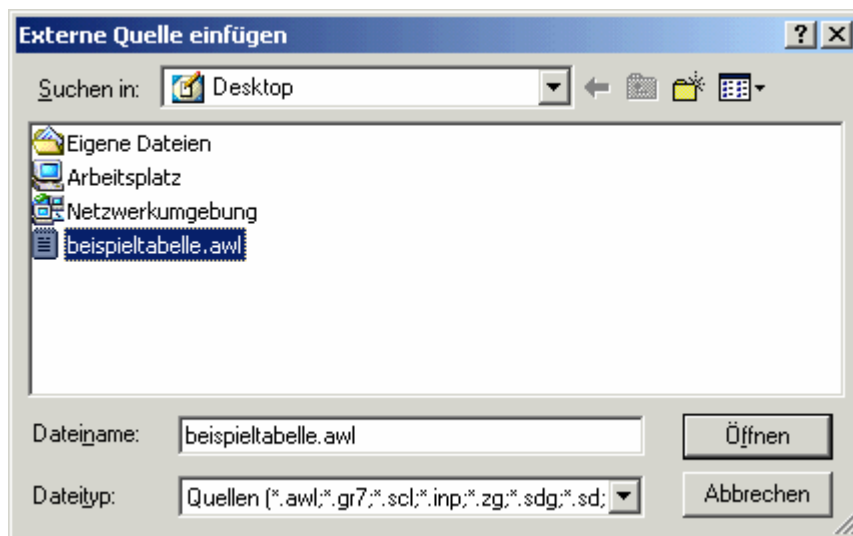


Bild 3-66 Datei für die Einfügung als externe Quelle in STEP7 wählen

Die gewählte Datei wird geöffnet (hier: BEISPIELTABELLE.AWL). Sie ist nun als Quelldatei in der Projektierung unter „Quellen“ vorhanden. Dort wird sie angewählt und wiederum geöffnet.



Das folgende Bild zeigt die unter „Quellen“ vorhandene Beispieldatei sowie deren kontextsensitives Menü:

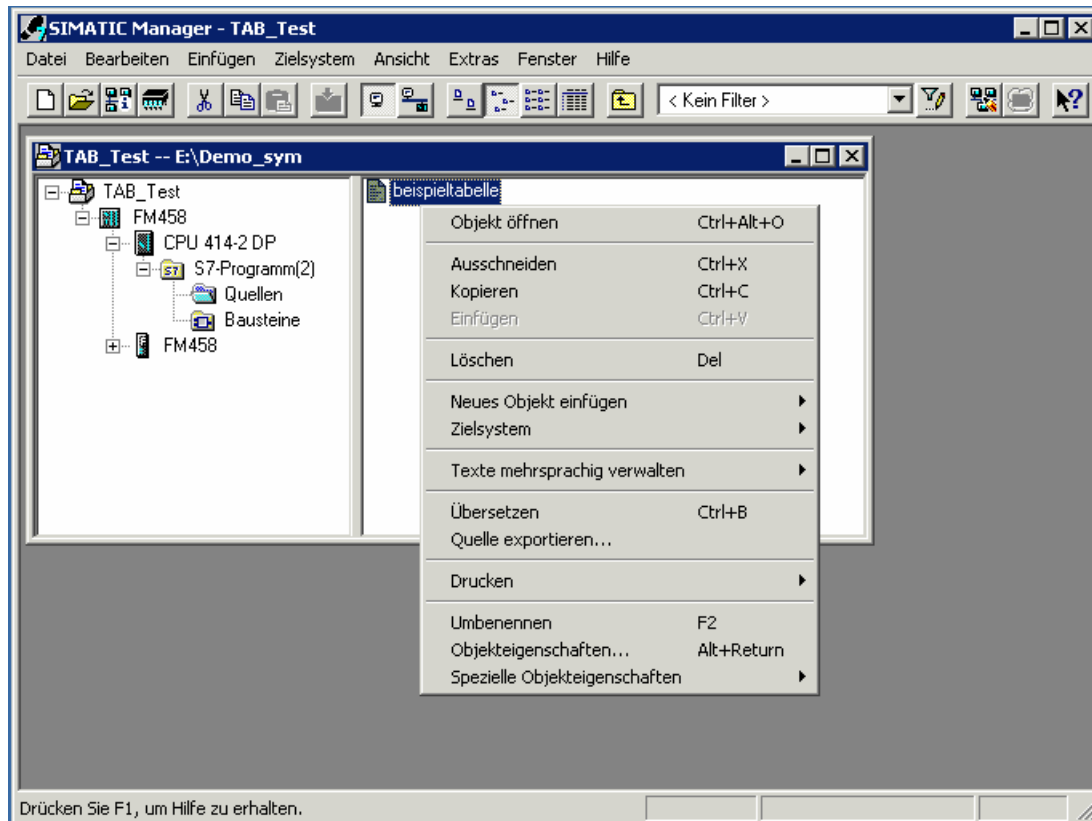


Bild 3-67 Erzeugte Quelldatei in STEP7

Nach dem Öffnen der Datei steht sie im Programm „KOP/AWL/FUP“ zur Bearbeitung zur Verfügung. Sie ist dort lediglich über „Datei / Übersetzen“ zu übersetzen.

Das folgende Bild zeigt die Vorgehensweise:

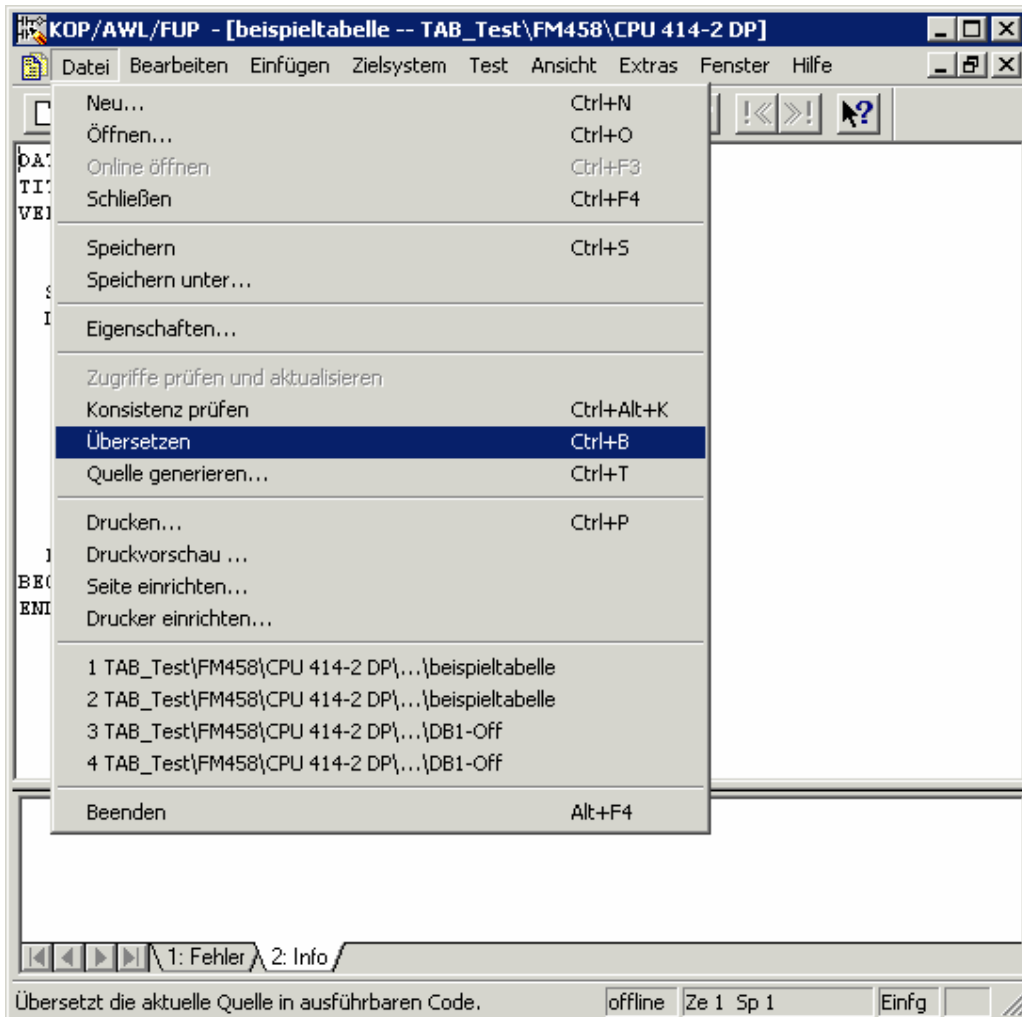


Bild 3-68 Übersetzen der Quelldatei in der Applikation „KOP/AWL/FUP“

Nach dem erfolgreichen Übersetzen der Datei steht in der Projektierung ein neuer DB zur Verfügung. Der Name des DBs entspricht dem in der Kopfzeile der Datei angegebenen Namen.

Das folgende Bild zeigt den in der STEP7-Projektierung unter „Bausteine“ neu generierten DB:

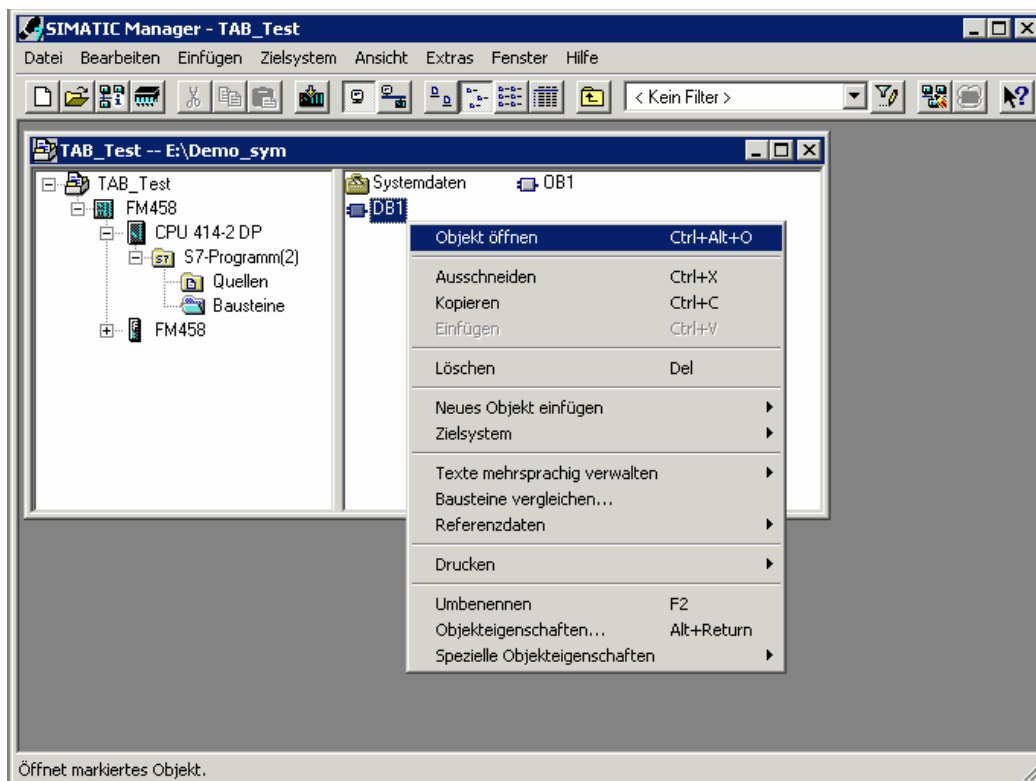


Bild 3-69 Durch Übersetzen der Quelldatei neu erzeugter DB

Um den Inhalt des DBs zu überprüfen, kann er im Programm „KOP/AWL/FUP“ geöffnet werden. Im Menü „Ansicht“ ist „Datenansicht“ anzuwählen, um sowohl die Anfangswerte als auch die Aktualwerte anzuzeigen.

Das folgende Bild zeigt den Inhalt des geöffneten DBs:

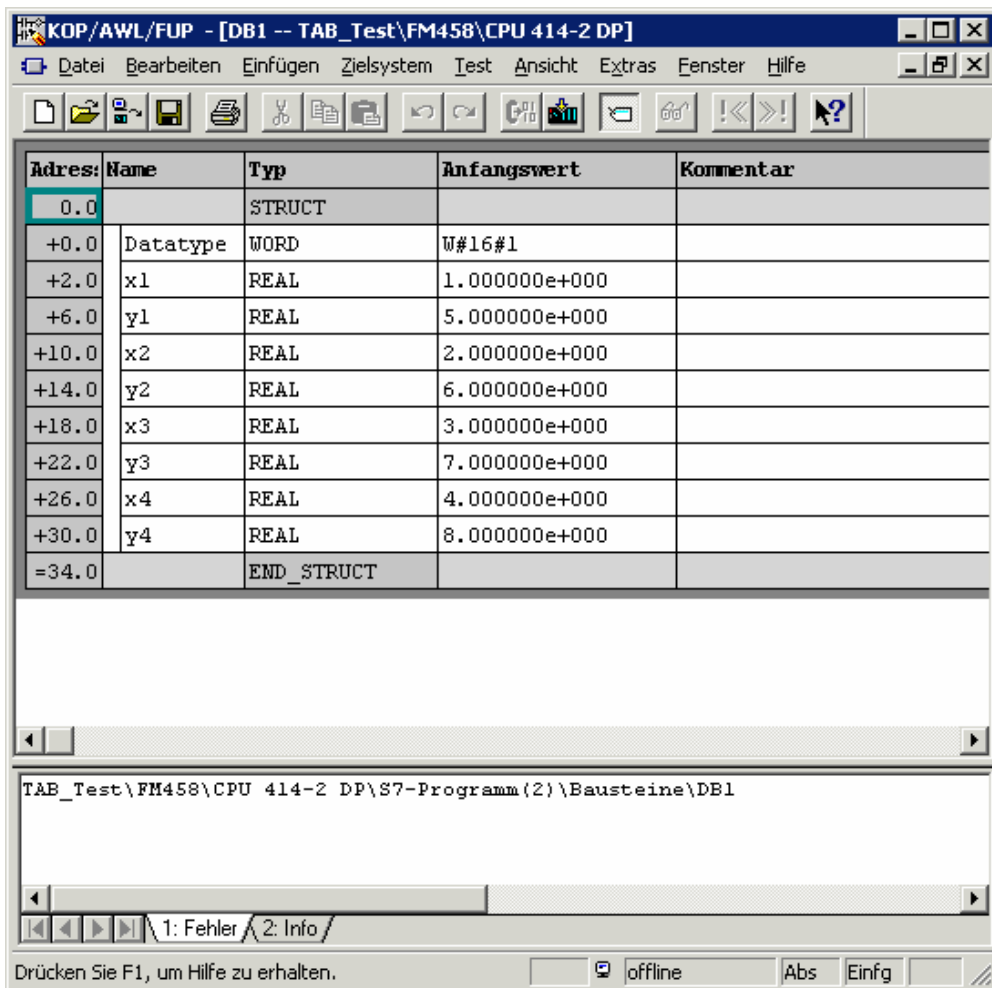


Bild 3-70 Inhalt des neu erzeugten DBs in der Applikation „KOP/AWL/FUP“

### 3.10.3.3 Nachladen von Tabellenwerten in einen DB

Sollen Tabellenwerte in den DB nachgeladen werden, weil die Tabelle zu groß ist und nicht genügend Anwendungsspeicher für mehrere DBs zur Verfügung steht, so ist die Tabelle in mehreren Teilübertragungen an die SIMATIC FM 458 Applikationsbaugruppe zu übertragen. Hierzu muss die Tabelle in einzelne Teiltabellen zerlegt werden. Die Größe der einzelnen Teiltabellen ist dabei so zu wählen, dass der zur Verfügung stehende Anwenderspeicher der S7-CPU nicht überschritten wird. Die einzelnen Teiltabellen werden dann nacheinander übertragen.

#### HINWEIS

Es ist unbedingt darauf zu achten, dass die einzelnen Teiltabellen in aufsteigender Reihenfolge der Wertepaare übertragen werden. Eine falsche Reihenfolge führt dazu, dass die Tabellenwerte in der Projektierung nicht korrekt zur Verfügung stehen.

Es stehen zwei Möglichkeiten zur Verfügung:

- Jeweils nacheinander erfolgende manuelle Eingabe der einzelnen Tabellenteile am DB in der Applikation „KOP/AWL/FUP“ und anschließende Übertragung dieses Tabellenteils
- Generieren einzelner Quelldateien für jeden Tabellenteil unterschiedlichen Namens und jeweils nacheinander erfolgende Einbindung im DB und anschließende Übertragung

### **Manuelle Eingabe**

Für das Nachladen von Tabellenwerten in einen DB per manueller Eingabe sind folgende Schritte durchzuführen:

- Der entsprechende DB ist per Doppelklick in der Applikation „KOP/AWL/FUP“ zu öffnen.
- Die vorhandenen Tabellenwerte sind durch die Eingabe der Werte des nachfolgenden Tabellenteils zu ersetzen.
- Der DB ist zu speichern.
- Die Werte des Tabellenteils stehen jetzt für die Übertragung zur Verfügung.

### **Generieren mehrerer Quelldateien**

Für das Nachladen von Tabellenwerten in einen DB durch die Generierung mehrerer Quelldateien sind folgende Schritte durchzuführen:

- Im Header der einzelnen Quelldateien (\*.AWL) ist stets derselbe DB-Name anzugeben.
- Die einzelnen Dateien dürfen nicht größer sein als die Speicherkapazität des DBs.
- Die Dateinamen sind am besten in aufsteigender Reihenfolge zu nummerieren.
- Die einzelnen Dateien sind nun wie oben beschrieben als Quelldateien einzubinden. Sie werden jedoch noch nicht übersetzt.
- Die erste Quelldatei wird übersetzt und die nun im DB vorhandenen Tabellenwerte übertragen.
- Die zweite Quelldatei wird übersetzt, so dass deren Tabellenwerte nun im DB zur Verfügung stehen. Diese werden jetzt zur S7-Steuerung übertragen.
- Analog werden nun nacheinander die weiteren Quelldateien übersetzt und übertragen.
- Bei der Übertragung des letzten Tabellenteils ist der Anschluss LASTDB von 0 auf 1 zu setzen. Damit wird das Ende der Übertragung signalisiert.

### 3.10.3.4 Aufbau des Datentelegramms bei TCP/IP- oder DUST1-Verbindung

Handelt es sich bei der Kommunikationsverbindung um eine TCP/IP- oder DUST1-Verbindung, dann ist der Aufbau der Datentelegramme zu beachten. Dieser wird im folgenden beschrieben. Die Datentelegramme werden mit den Funktionsbausteinen CTV und CRV „erstellt“.

Das Datentelegramm ist so definiert, dass sämtliche Tabellenwerte sowohl in einem Datenblock als auch in mehreren Datenblöcken übertragen werden können.

Die folgende Tabelle zeigt den Aufbau eines Datenblocks:

Datentyp	Beschreibung
char [4]	Telegrammkennung. Identifiziert jedes Tabellentelegramm mit der Kennung „TAB0“
u_int16	Telegrammkommandos (Bit-Kodiert) 1: Neue Tabelle (steigende Flanke von 0 -> 1) 2: Tabellenende
u_int16	Datenformat (REAL=1, DINT=2)
u_int32	Nr. des aktuellen Datenblockes
u_int32	Anzahl der Tabellenwerte (X- und Y-Werte) Die Anzahl der Werte muss immer geradzahlig sein. D.h., es werden immer gleich viele X- und Y-Werte übertragen.
u_int32 [56] / float [56]	Array mit Tabellenwerten. (Immer abwechselnd X- und Y-Werte)

Für jeden empfangenen Datenblock sendet der TAB bzw. der TAB\_D eine Quittierung an den Sender.

Die folgende Tabelle zeigt den Aufbau des Quittierungstelegramms:

Datentyp	Beschreibung
char [4]	Telegrammkennung. Identifiziert jedes Tabellentelegramm mit der Kennung „TAB0“
u_int32	Nr. des aktuellen Datenblockes
u_int32	Status / Fehlernummern 0xB210 OK (Datenblock ist in Ordnung) .....

#### HINWEIS

Neue Tabellendaten werden nur dann in die inaktive Tabelle übertragen, wenn das Kommando „Neue Tabelle“ gesetzt ist. Nach Empfang des Kommandos „Tabellenende“, werden alle weiteren Tabellendaten abgelehnt, bis wieder das Kommando „Neue Tabelle“ empfangen wird.

### 3.10.4 Automatikbetrieb: Speicherkarte

Tabellenwerte können mit Hilfe des D7-SYS additionalComponentBuilder (in D7-SYS V5.2 plus SP1 enthalten) zu Komponenten zusammengestellt werden, welche beim Laden als zusätzliche Objekte auf die Speicherkarte geladen werden können. Von dort werden sie mit Hilfe der Funktionsbausteine TAB bzw. TAB\_D ausgelesen.

Eine oder mehrere Tabellendateien werden in den D7-SYS additionalComponentBuilder importiert, der diese Dateien zu einer Komponentendatei (Ladedatei) zusammenfügt, welche auf die Speicherkarte geladen werden kann.

Grundsätzlich überprüft der D7-SYS additionalComponentBuilder (aCB) den Inhalt der Dateien nicht. Eine Ausnahme von dieser Regel stellen Tabellen dar. Diese Tabellendateien werden inhaltlich überprüft. Bei einem fehlerhaften Aufbau der Tabellendatei meldet dies der aCB sofort.

In den folgenden Abschnitten wird das Vorgehen von der Erstellung einer Tabellendatei bis zur Projektierung der Funktionsbausteine anhand eines Beispiels erläutert.

#### 3.10.4.1 Erstellung einer Tabellendatei im csv-Format

Die Tabellenwerte werden mit einer Tabellenkalkulation (z.B. Excel) beliebig erstellt.

The image shows two side-by-side screenshots of Microsoft Excel spreadsheets. The left spreadsheet, titled 'Table1.xls', shows a table with two columns, A and B, and 16 rows of numerical data. The right spreadsheet, titled 'Table2.xls', shows a table with two columns, A and B, and 16 rows of numerical data, including some scientific notation values.

	A	B
1	1,00	1,00
2	1,10	1,21
3	1,20	1,44
4	1,30	1,69
5	1,40	1,96
6	1,50	2,25
7	1,60	2,56
8	1,70	2,89
9	1,80	3,24
10	1,90	3,61
11	2,00	4,00
12	2,10	4,41
13	2,20	4,84
14	2,30	5,29
15	2,40	5,76
16	2,50	6,25

	A	B
1	-1	1
2	-0,9	0,81
3	-0,8	0,64
4	-0,7	0,49
5	-0,6	0,36
6	-0,5	0,25
7	-0,4	0,16
8	-0,3	0,09
9	-0,2	0,04
10	-0,1	0,01
11	-1,38778E-16	1,92593E-32
12	0,1	0,01
13	0,2	0,04
14	0,3	0,09
15	0,4	0,16
16	0,5	0,25

Bild 3-71 Tabellenwerte in Excel

## Bedingungen

Die Tabellendateien müssen folgende Bedingungen erfüllen:

- Eine Tabellendatei darf grundsätzlich nur aus zwei Spalten bestehen – falls weitere Spalten in der Tabelle vorhanden sind, wird eine Fehlermeldung in einem Dialogfenster ausgegeben.
- Beide Spalten müssen gleich viele Werte enthalten. Ist dies nicht der Fall, so gibt der D7-SYS additionalComponentBuilder eine Fehlermeldung in einem Dialogfenster aus und die Tabellenwerte werden abgelehnt.

Der D7-SYS additionalComponentBuilder erwartet folgende Datenformat:

- [+/-] xxx.yyy – Realwert, Nachkommastellen werden durch „.“ angegeben (z.B. 145.123)
- [+/-] xxx,yyy – Realwert, Nachkommastellen werden durch „.“ angegeben (z.B. 145,122)
- [+/-] xxx.yyyE+/-mm – Realwert in Exponentialdarstellung, Nachkommastellen werden durch „.“ angegeben (z.B. 145.122E+12)
- [+/-] xxx,yyyE+/-mm – Realwert in Exponentialdarstellung, Nachkommastellen werden durch „.“ angegeben (z.B. 187,122E+12)

Bei Typbeschreibung „Table DINT“:

- [+/-]xxx – Integer bzw. Double-Integer (z.B. 145)

Weiterhin gelten folgende Bedingungen für die Tabellendateien:

- ASCII-Dateien
- Trennung der Tabellenspalten mittels Semikolon bzw. Tabulatorzeichen
- Trennung der Zeilen durch Zeilenumbruch bzw. Semikolon



**Tabellen speichern**

Tabellen, die mit MS Excel erzeugt werden, und im \*.csv-Format oder als „Text (Tabs getrennt)“ abgespeichert werden, erfüllen diese Bedingungen.

Die folgende Abbildung zeigt zwei Beispieldateien mit Tabellenwerten, die im csv-Format gespeichert wurden:

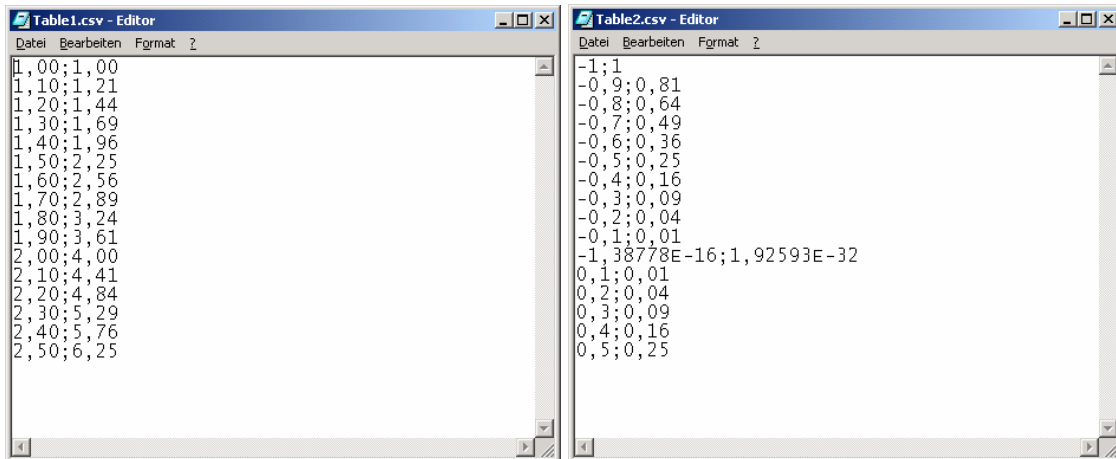


Bild 3-72 Tabellenwerte, die durch Semikolon getrennt wurden (\*.csv-Format)

### 3.10.4.2 Arbeiten mit dem D7-SYS additionalComponentBuilder

Nachdem die Tabellendateien im csv-Format abgespeichert wurden, können sie in den D7-SYS additionalComponentBuilder importiert werden.

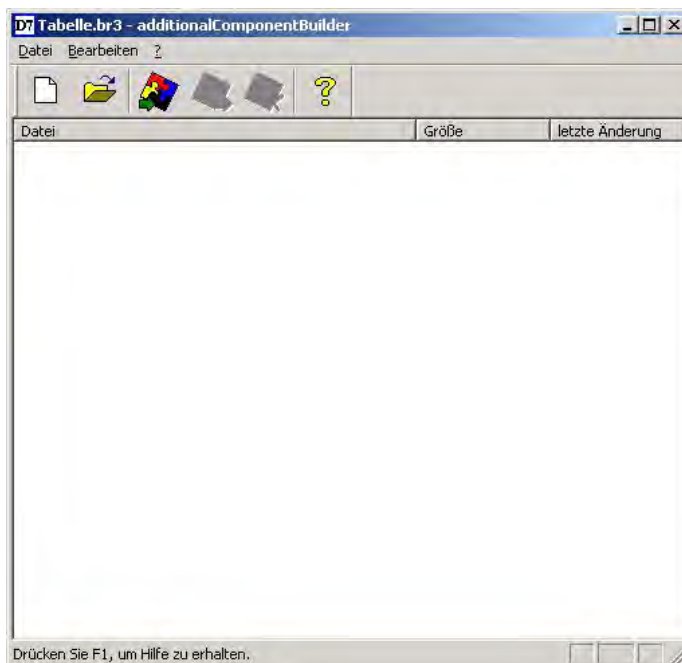



Bild 3-73 D7-SYS additionalComponentBuilder

Als nächster Schritt wird mit  eine neue Komponentendatei angelegt. Hierzu werden zuerst die Eigenschaften in folgendem Dialogfeld angegeben.

### Neue Komponente



Bild 3-74 Einstellen der Eigenschaften

Folgende Einstellungen sind vorzunehmen:

Diese Eigenschaften können zu einem späteren Zeitpunkt nicht mehr geändert werden und sind dann grau hinterlegt.

- **D7-SYS Version**  
Listbox, in der die Version angegeben wird, für die die Komponente erzeugt werden soll
- **Komponententyp**  
Listbox mit den festen Einträgen „USER“, „IT1“ und „IT2“. Defaultwert ist „USER“

Die Einträge haben dabei folgende Bedeutung:

- USER = Durch Anwender erzeugte Komponentendatei, z.B. Tabellendateien
- IT1/IT2 = System-Komponentendatei für ITSP-Baugruppen

- **Typbeschreibung**  
Listbox mit den Einträgen „Table REAL“ und „Table DINT“. Defaultwert beim Komponententyp „USER“ ist „Table REAL“. „Table DINT“ wird für Tabellen im DINT-Format verwendet.

Die Einträge haben folgende Bedeutung:

- Table REAL:      Tabellendatei mit Datentyp REAL
- Table DINT:      Tabellendatei mit Datentyp Double Integer

Eine neue Typbeschreibung kann in die Listbox eingegeben und mit RETURN bestätigt werden. Diese neue Typbeschreibung wird dann in die Listbox übernommen und kann beim nächsten Mal aus der Listbox ausgewählt werden.

### Speichern

Ist die Festlegung der Einstellungen abgeschlossen kann die neue Komponentendatei angelegt werden. Die neue Komponentendatei wird standardmäßig in C:\temp angelegt. Wird ein anderer Speicher-Pfad angegeben, so wird dieser bei einem erneuten Programmstart als Standard-Speicher-Pfad verwendet.

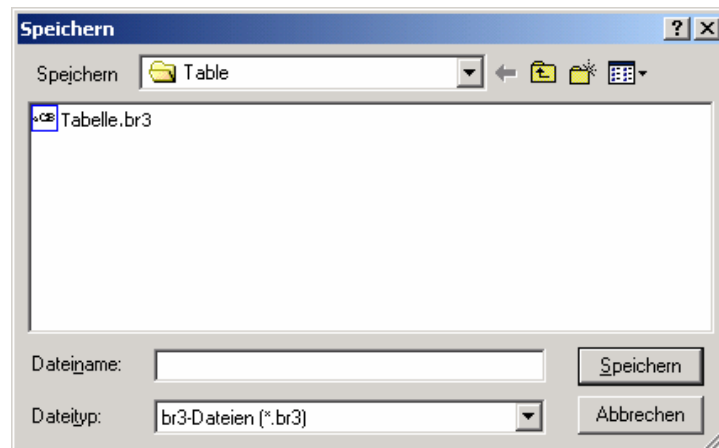



Bild 3-75 Speichern der neuen Komponentendatei

Nun können Tabellendateien hinzugefügt werden. Mit  wird ein Dateiauswahlfenster geöffnet, mit dem die gewünschten Tabellendateien selektiert werden können.

### HINWEIS

In einer Komponente mit Typenbeschreibung "Table" können nur Tabellen in einem einheitlichen Werteformat enthalten sein! D.h., Table REAL enthält nur Tabellen mit REAL-Werten.

Folgende Abbildung zeigt den Inhalt des D7-SYS additionalComponentBuilder nach dem Import der beiden erzeugten Tabellendateien:

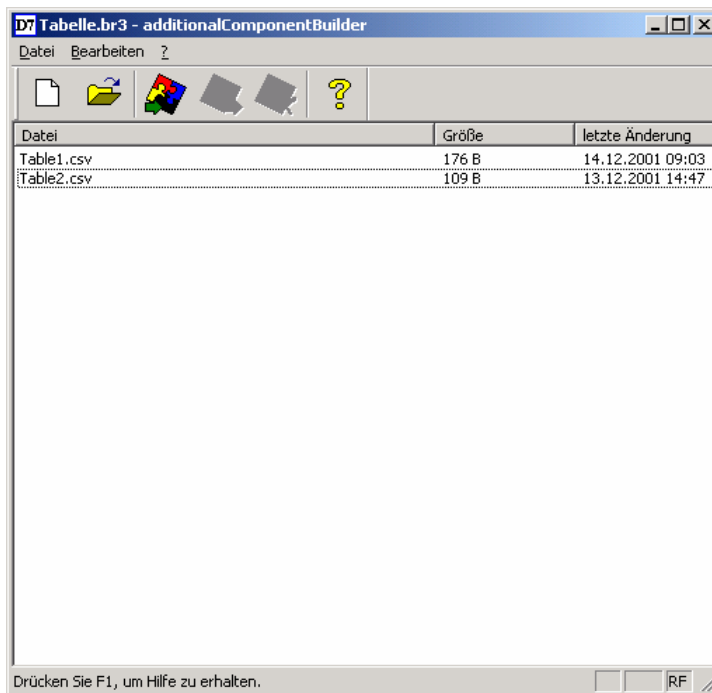


Bild 3-76 D7-SYS additionalComponentBuilder mit importierten Tabellendateien

Es können jederzeit weitere Tabellendateien hinzugefügt oder importierte wieder gelöscht werden. Der D7-SYS additionalComponentBuilder übernimmt automatisch die Verwaltung der Tabellendateien und speichert die modifizierten Komponentendateien.

## Öffnen

Beim Öffnen von bestehenden Komponenten ist „C:\temp“ der Standard-Suchpfad des D7-SYS additionalComponentBuilders. Wird ein anderer Pfad ausgewählt, so wird bei einem erneuten Programmstart dieser als Standard-Suchpfad verwendet.

### 3.10.4.3 Laden

Nach dem die Komponentendatei mit dem D7-SYS additionalComponentBuilder angelegt wurde, kann sie im allgemeinen Ladedialog geladen werden.

#### (1) Öffnen des Ladedialogs in D7-SYS mit "Zielsystem → Laden"

Mit diesem Dialog kann die aktuelle Projektierung die optionalen Komponenten auf eine Speicherkarte laden (Off-/Online).

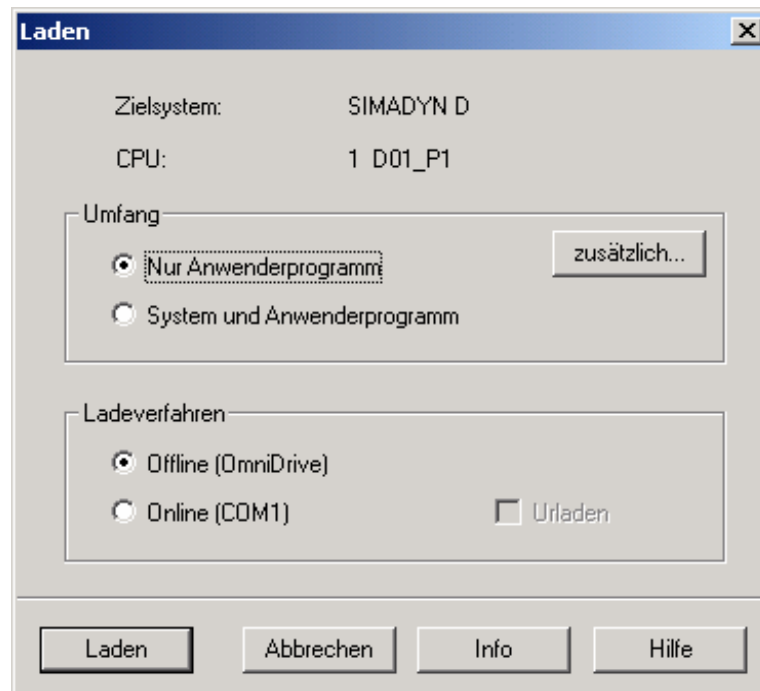


Bild 3-77 Laden-Dialog über Zielsystem → Laden in D7-SYS

#### (2) Öffnen des Dialogs für optionale Komponenten

Es können maximal 2 Komponenten ausgewählt werden. Durch Klick auf den Button "NEU" kann für die ausgewählte Komponente eine Datei ausgewählt werden.



Bild 3-78 Auswahldialog für optionale Komponenten, wie z.B. Tabellendaten

### (3) Zum Auswählen einer zusätzlichen Komponente öffnet sich ein Dateiauswahldialog

Die vorher mit dem D7-SYS additionalComponentBuilder erzeugte Komponentendatei wird nun der Komponente IT1 zugewiesen und beim anschließenden Ladevorgang auf die Speicherkarte geschrieben.

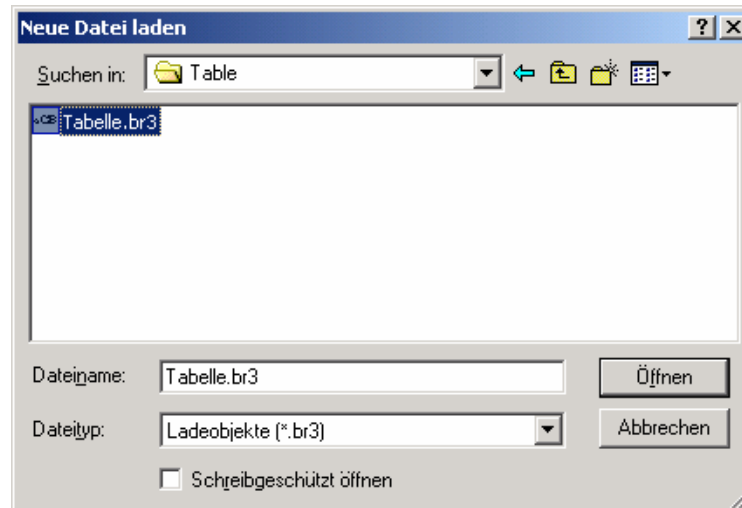


Bild 3-79 Laden einer Komponentendatei

#### 3.10.4.4 Projektierung der Funktionsbausteine

Für die Betriebsart „Automatikbetrieb Speicherkarte“ müssen nur die Funktionsbausteine TAB und/oder TAB\_D projektiert werden, je nachdem ob Tabellenwerte des Datentyps REAL und/oder DINT verwaltet werden sollen. Jede Tabelle darf nur Werte eines Datentyps enthalten. Sollen mehrere Tabellen unterschiedlicher Datentypen verwaltet werden, so ist für jede Tabelle ein TAB bzw. TAB\_D zu projektieren.

Die Funktionsbausteine TAB und TAB\_D sollten in einer Abtastzeit größer gleich 32ms projektiert werden. Folgende Anschlusseinstellungen sind nötig:

- CTS=** 0
- US =** Nicht belegt
- NAM =** Name der Tabellendatei (mit Dateinamenserweiterung, wie beim „Speichern“ festgelegt, z.B. MS Excel)
- AUT =** 1 (Automatikbetrieb aktiviert)

Die folgende Abbildung zeigt die Projektierung:

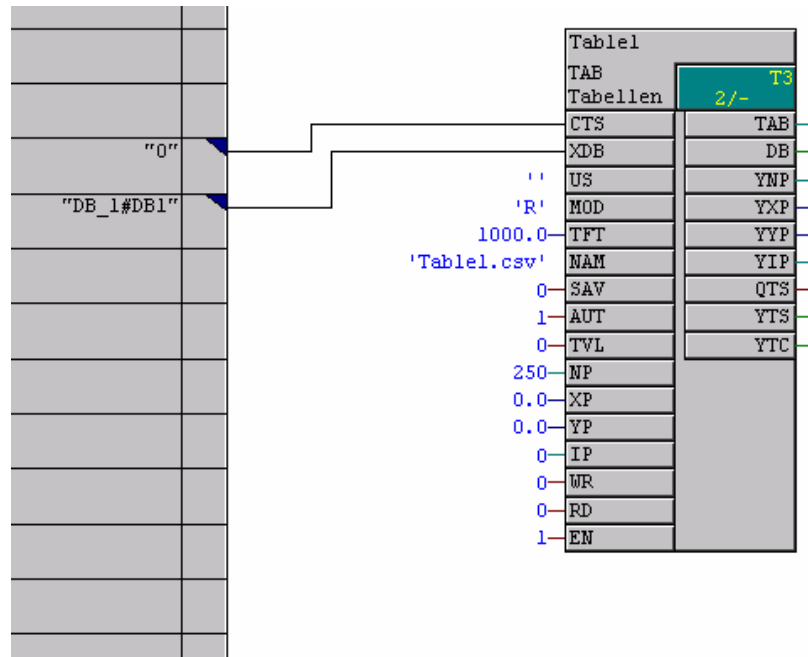


Bild 3-80 Projektierung des TAB-Funktionsbausteins

In der folgenden Abbildung sind die Tabellen Funktionsbausteine für 2 Tabellen dargestellt. Die Tabellenwerte, die von den Funktionsbausteinen nun verwaltet werden, können nun von weiteren Funktionsbausteinen, wie z.B. FB TABCAM, genutzt werden.

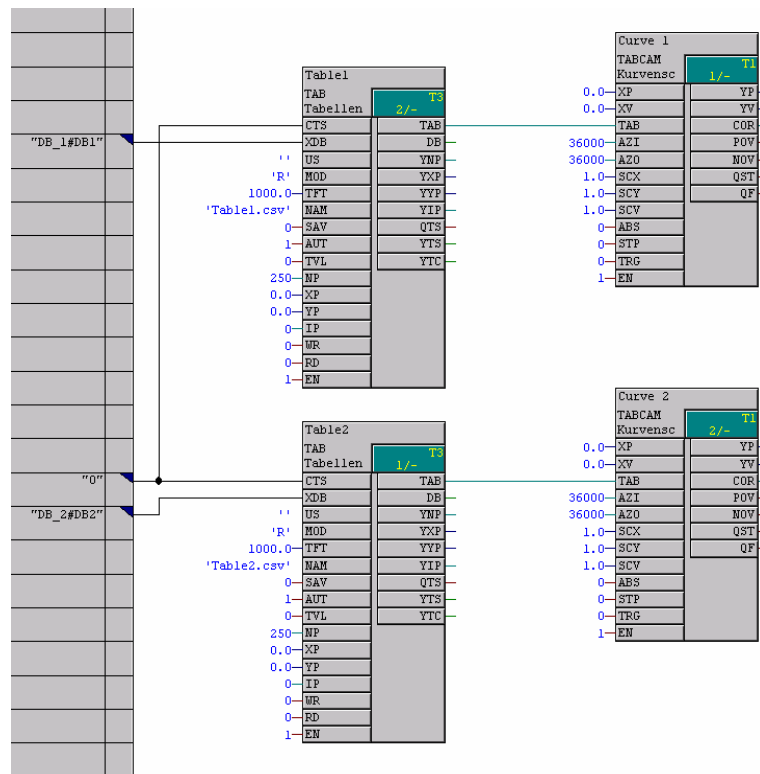


Bild 3-81 Projektierungsbeispiel

## 3.11 Kommunikations-Dienst Meldesystem

<b>Allgemeines</b>	Das Meldesystem bietet dem Anwender die Möglichkeit, bestimmte, von ihm ausgewählte Ereignisse protokollieren zu lassen. Eine Beschreibung dieser Ereignisse wird im Meldefolgepuffer gesammelt und ist für den Anwender über eine Datenschnittstelle verfügbar.
<b>Projektierung</b>	Das Meldesystem arbeitet rein CPU-lokal. Es muss genau ein Zentralbaustein und mindestens ein Meldeauswertebaustein projektiert werden. Bezüglich der Anzahl der Bausteine gibt es keine Projektierungsvorschriften.
<b>Funktionsbausteine für das Meldesystem</b>	<p>Das Meldesystem besteht aus 3 Arten von Funktionsbausteinen:</p> <ul style="list-style-type: none"><li>• <b>Zentralbaustein @MSC</b> Der Zentralbaustein legt benötigte Datenstrukturen an und verwaltet sie. Er ist auch für die Auswertung von Kommunikations- und Systemfehlermeldungen verantwortlich.</li><li>• <b>Meldeeintragsbausteine MER ...</b> Meldeeintragsbausteine generieren Meldungen als Folge einer Änderung eines Eingangs. Meldeeintragsbausteine können sich gegenseitig unterbrechen. Deshalb müssen die Meldungen nicht in der Reihenfolge, in der sie aufgetreten sind, in den Meldefolgepuffer eingetragen werden. Die Meldeeintragsbausteine unterscheiden sich in<ul style="list-style-type: none"><li>– der Anzahl der generierbaren Meldungen und</li><li>– den Möglichkeiten, zusätzlich eingehende Prozesszustände in Form von Messwerten zu verarbeiten.</li></ul></li><li>• <b>Meldeauswertebausteine MSI ...</b> Meldeauswertebausteine geben die von den Meldeeintragsbausteinen generierten Meldungen über eine Datenschnittstelle aus und machen sie dem Anwender verfügbar.</li></ul>

### 3.11.1 Eintragslogik der Meldeeintragsbausteine

#### 3.11.1.1 Meldeeintragsbausteine für eine kommende Meldung

<b>Eintragslogik</b>	<p>Bei Meldeeintragsbausteinen, die nur eine kommende Meldung generieren, müssen folgende Bedingungen für einen Meldeeintrag erfüllt sein:</p> <ul style="list-style-type: none"><li>• Der Anschluss EN muss gesetzt sein.</li><li>• Am Anschluss I1 muss eine positive Flanke anliegen.</li><li>• Anschluss Q1 oder SM muss zurückgesetzt sein.</li></ul> <p>Wenn die Bedingungen erfüllt sind, dann wird eine Meldung generiert und der Anschluss Q1 gesetzt.</p>
----------------------	---



Wenn die Bedingungen nicht erfüllt sind, dann wird, falls der Anschluss SM rückgesetzt ist, der Anschluss Q1 zurückgesetzt.

### 3.11.1.2 Meldeeintragsbausteine für eine kommende und eine gehende Meldung

**Eintragslogik** Bei Meldeeintragsbausteinen, die eine kommende und eine gehende Meldung generieren, müssen folgende Bedingungen für einen Meldeeintrag erfüllt sein:

- Der Anschluss EN muss gesetzt sein.
- Für eine kommende Meldung muss am Anschluss I1 eine positive Flanke anliegen und Anschluss Q1 oder SM rückgesetzt sein.
- Für eine gehende Meldung muss am Anschluss I1 eine negative Flanke anliegen und Anschluss Q2 oder SM rückgesetzt sein.

Wenn die Bedingungen erfüllt sind, dann wird

- bei einer positiven Flanke eine kommende Meldung generiert und Anschluss Q1 gesetzt,
- bei einer negativen Flanke eine gehende Meldung generiert und Anschluss Q2 gesetzt.

Wenn die Bedingungen nicht erfüllt sind, dann wird, falls der Anschluss SM zurückgesetzt ist, der Anschluss Q1 und Q2 zurückgesetzt.

**Besonderheiten bei MER16, MERF16, MER0, MERF0** Bei den Meldeeintragsbausteinen MER16, MERF16, MER0, MERF0, die als Meldeanschluss einen Vektor haben und 16 oder 32 Meldungen generieren, müssen beim Meldeanschluss IS1 und Ausgangsanschluss QS1 bzw. QS2 die entsprechenden Bitpositionen die Bedingungen der Eintragslogik erfüllen. Zusätzlich existiert bei diesen Bausteinen noch der Ausgangsanschluss QN, der anzeigt, ob überhaupt eine Meldung generiert wurde.

### 3.11.2 Projektierungsbeispiel für Meldesystem

**Voraussetzungen für Meldesystem**

- Baugruppenträger
- mindestens eine CPU im Baugruppenträger
- eine Datenschnittstelle mit dem Namen "D01" ist vorhanden

**Notwendige Funktionsbausteine** Im Beispiel werden nur die für das Meldesystem notwendigen Bausteine aufgeführt. Zentrale Kommunikationsbausteine (z.B. für die Datenschnittstelle) werden nicht aufgeführt.

Das projektierte Meldesystem besteht aus:

- 1 Zentralbaustein @MSC
- 2 Eintragsbausteine (MER und MERF0)
- 2 Meldeauswertebausteine (MSI und MSIPRI)

**Name und Meldepuffer**

Der Name des Meldesystems ist "MELD". Dieser Name ist an allen CMS-Anschlüssen der Meldebausteine projektiert. Der Meldepuffer hat eine Größe von 30 Meldungen (Anschluss NOM am @MSC), liegt im flüchtigen RAM (Anschluss SAV am @MSC) und ist für Meldeinträge freigegeben (Anschluss MUN am @MSC).

**Zuordnung von Meldung und Baustein**

Durch die RP- und RS-Anschlüsse, wobei jeder Baustein des Meldesystems mindestens einen RP-Anschluß besitzt, ist eine Zuordnung von generierten Meldungen zu den Bausteinen möglich. Dazu wurde wie folgt vorgegangen:

- **Präfix 0**  
Kennzeichnet eine Meldung, die vom @MSC generiert wird (Kommunikations- und Systemfehlermeldungen). Deshalb wurde der Anschluss RP vom @MSC mit dem Wert 0 belegt. Das Suffix wird vom @MSC selbstständig generiert, je nach Meldungsart.
- **Präfix 1**  
Kennzeichnet eine Meldung, die vom MSI generiert wird (Überlaufmeldungen). Deshalb wurde der Anschluss RP vom MSI mit dem Wert 1 belegt. Das Suffix wird vom MSI selbstständig generiert (Anzahl übergelaufener Meldungen).
- **Präfix 2**  
Kennzeichnet eine Meldung, die vom MSIPRI generiert wird.
- **Präfix 3**  
Kennzeichnet eine Meldung, die von einem Meldebaustein (MER oder MERF0) generiert wird. Deswegen wurde der Anschluss RP vom MER und MERF0 mit dem Wert 3 belegt. Das Suffix wird hier gegenüber den anderen Bausteinen nicht selbstständig generiert. Hier sind Anschlüsse vorhanden, an denen das Suffix projektiert werden kann. Im Beispiel werden 33 verschiedene Meldungen generiert (1 Meldung MER, 16 kommende Meldungen MERF0, 16 gehende Meldungen MERF0), die von 0 - 32 durchnummeriert werden:
  - Die Meldung des Bausteins MER bekommt das Suffix 0 (RS-Anschluß MER).
  - Die 16 kommenden Meldungen des Bausteins MERF0 bekommen das Suffix 1-16 (RS1-Anschluß MERF0).
  - Die 16 gehenden Meldungen des Bausteins MERF0 bekommen das Suffix 17-32 (RS2-Anschluß MERF0).
- **Suffix**  
Beim Baustein MERF0 wird beim Suffix ein Basiswert spezifiziert, auf dem die Bitnummer des meldungsgenerierenden Bits des Meldesignalvektors IS1 aufaddiert wird.

**Funktionale Zusammenfassung der Meldungen**

Durch Präfix und Suffix ist somit nicht nur eine eindeutige Zuordnung der Meldungen zu den generierenden Bausteinen möglich, sondern auch eine funktionale Zusammenfassung der Meldungen. Im Projektierungsbeispiel generieren die Bausteine MER und MERF0 Meldungen mit demselben Präfix, was auf eine logische Zusammengehörigkeit schließen lässt.

**Kanal auf der  
Datenschnittstelle**

Im Projektierungsbeispiel legen beide Meldeauswertebausteine einen Kanal auf der Datenschnittstelle D01 im Modus „Select“ an (Es könnte somit auch derselbe Kanalname projektiert werden).

**Messwerteingang  
und Meldesignale**

Der Messwerteingang des Bausteins MER ist in diesem Beispiel nicht verdrahtet. Am Messwerteingang wird normalerweise ein Prozesszustand angelegt. Ebenso verhält es sich mit den Meldesignalen des Funktionsbausteins MERF0.

**Generieren und  
Auslesen von  
Meldungen**

Durch eine positive Flanke am Anschluss I1 des Bausteins MER oder einen sich ändernden Wert am Anschluss IS1 des Bausteins MERF0 werden Meldungen generiert. Die erste Meldung würde sofort von den Meldeauswertebausteinen aus dem Meldepuffer ausgelesen und in den Datenkanal übertragen werden, da beide Bausteine „enabled“ sind (Anschluss EN=1). Weitere Meldungen werden erst dann in den Datenkanal übertragen, wenn die vorherige Meldung aus dem Kanal ausgelesen worden ist.

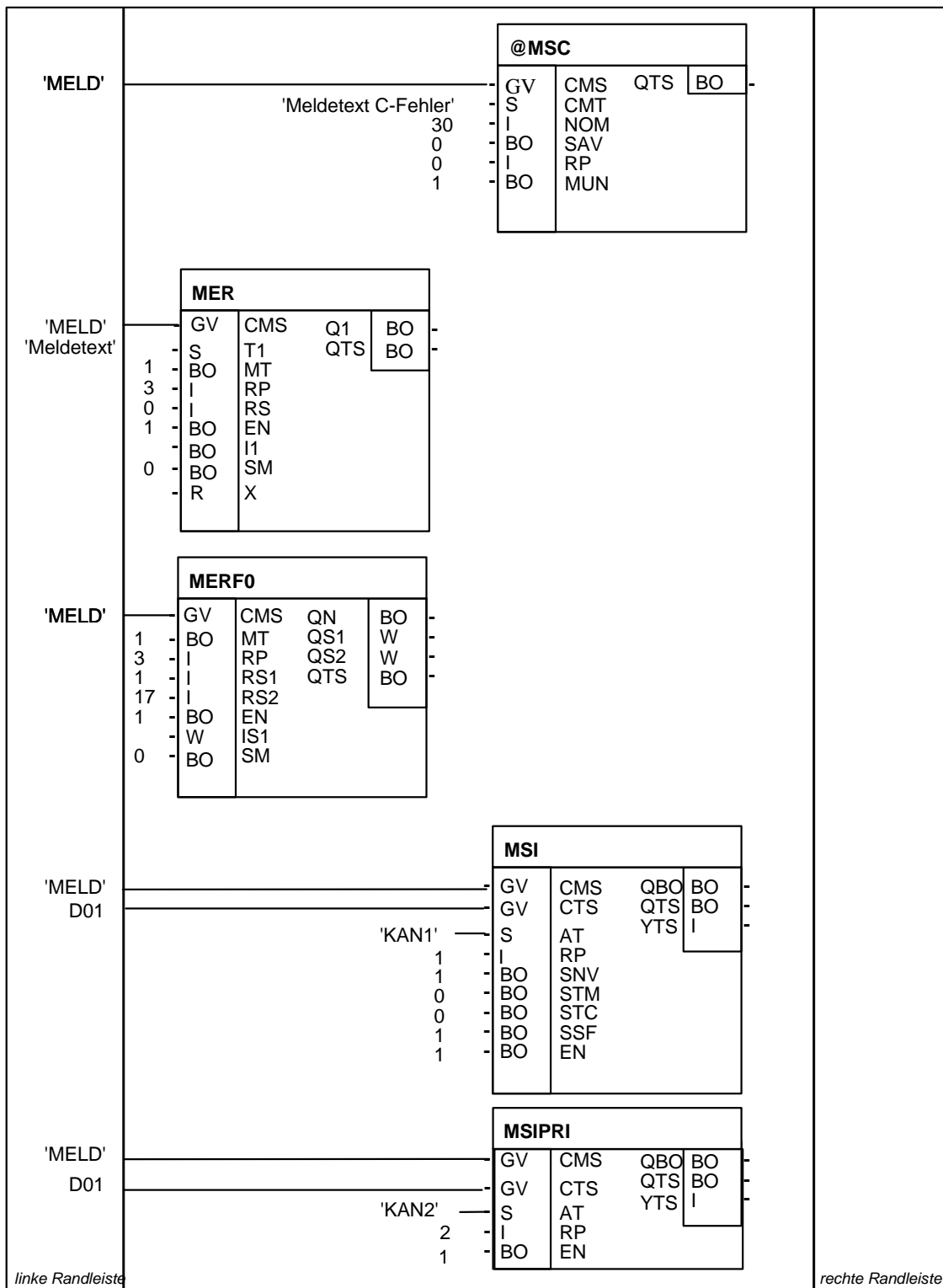


Bild 3-82 Beispielprojektierung Kommunikations-Dienst Meldesystem

### 3.11.3 Ausgabeformate des Meldeauswertebausteins MSI

#### 3.11.3.1 Aufbau einer Fehler- oder Warnmeldung

<b>Allgemeines</b>	<p>Der Meldeauswertebaustein MSI hat zur Formatwahl vier Anschlüsse:</p> <ul style="list-style-type: none"> <li>• Anschluss SNV</li> <li>• Anschluss STM</li> <li>• Anschluss STC</li> <li>• Anschluss SSF</li> </ul> <p>Für den Empfänger einer Meldung und deren Interpretation ist das Format der Meldung wichtig.</p>
<b>Länge des Meldetextes</b>	<p>Der Anschluss STC bestimmt die Länge des Meldetextes. Durch STC=1 wird er auf eine konstante Länge (60 Zeichen) gesetzt. Ist ein Meldetext kürzer als die Maximallänge oder nicht vorhanden, so wird er mit Leerzeichen aufgefüllt. Der Vorteil liegt in der konstanten Anzahl der übertragenen Daten, was manche Kommunikationspartner fordern. Auf den sonstigen Aufbau der Meldung und auf die Meldetypbeschreibung hat dieser Anschluss keine Auswirkung.</p>
<b>Format des Meldetextes</b>	<p>Die Anschlüsse SNV, STM und SSF werden während der Initialisierungsphase einmalig ausgewertet und bestimmen dann das Format der ausgegebenen Meldungen. Die Meldungen werden in dem am Anschluss AT angegebenen Kanal auf der am Anschluss CTS angegebenen Datenschnittstelle ausgegeben.</p>

#### 3.11.3.2 Übersicht der Meldungsformate

<b>Spontankennung</b>	<p>Die Spontankennung hat den konstanten Wert 0 und ist nicht von Bedeutung.</p>
<b>Sequenznummer</b>	<p>Die Sequenznummer ist aus Zuverlässigkeitsgründen vorhanden und zählt die gesendeten Meldungen, damit der Empfänger erkennen kann, dass Meldungen verlorengegangen sind. Die Sequenznummer liegt im Wertebereich von 0-255. Hat die Sequenznummer den Maximalwert 255 erreicht, dann wird bei der nächsten Meldung wieder der Minimalwert 0 gesendet.</p>
<b>Meldetypbeschreibung</b>	<p>Prinzipiell wird zwischen standardisiertem und HEX-Format unterschieden. Beim standardisiertem Format werden die einzelnen Werte in der IEEE 754 bzw. ISO 646-Norm übertragen, die eine normierte 32-Bit Floating-Point-Darstellung beschreibt. Die Meldungen erhalten sowohl im standardisierten als auch im HEX-Format eine Meldetypbeschreibung, die Auskunft über das durch die Initialisierungsanschlüsse gewählte Format und andere Teile der Meldung gibt. Die Meldetypbeschreibung ist ein Bitvektor, der wie folgt zu interpretieren ist:</p>

- Bit 0: Ist dieses Bit gesetzt, werden Meldungsnummern ausgegeben (Kopie des Anschlusses SNV).
- Bit 1: Ist dieses Bit gesetzt, wird ein Meldetext ausgegeben (Kopie des Anschlusses STM, es sein denn, der Meldetext ist leer).
- Bit 2: Ist dieses Bit gesetzt, werden die Meldungen im standardisierten Format ausgegeben, ansonsten im HEX-Format (Kopie des Anschlusses SSF).
- Bit 3: Ist dieses Bit gesetzt, so ist ein Messwert vorhanden.
- Bit 4: Ist dieses Bit gesetzt, so ist ein Einheitentext vorhanden. Der Einheitentext kann nur dann vorhanden sein, wenn auch ein Messwert vorhanden ist. Ist kein Messwert oder Einheitentext vorhanden, so sind die entsprechenden Felder der Meldung bedeutungslos und in undefiniertem Zustand.
- Bit 5-7: unbesetzt.

**Meldungstyp**

Der Meldungstyp besteht aus einem Zeichen, welches den Typ des Meldeereignisses angibt, wobei 'S' Systemfehler, 'C' Kommunikationsfehler, 'F' Fehlermeldungen und 'W' Warnmeldungen kennzeichnen. Die ersten beiden Meldungstypen werden nur vom Zentralbaustein des Meldesystems generiert.

**Meldungspräfix**

Entspricht dem am Eintragsbaustein anliegenden Anschlusswert RP.

**Meldungssuffix**

Entspricht dem am Eintragsbaustein anliegenden Anschlusswert RS.

**Messwerteinheit und Skalierfaktor**

Im HEX-Format besteht die Messwertbeschreibung aus

- einem 32-Bit großem Skalierfaktor, der im Float-Format ausgegeben wird,
- dem vom Eintragsbaustein erfassten Messwert,
- einem Messwert-Datentyp (SIMADYN D / SIMATIC TDC -Datentyp als ASCII-Zeichenfolge),
- einer 8 Zeichen langen Messwerteinheit.

**HEX-Format und standardisiertes Format**

Da im HEX-Format bei der Initialisierung des Übertragungskanal das genaue Datenformat anzugeben ist, auf der anderen Seite der Messwert in der Größe der Darstellung (0,2 oder 4 Bytes) variiert, werden beim Messwert grundsätzlich 4 Bytes übertragen. Belegt der Messwert weniger als 4 Bytes - was am Messwert-Datentyp erkannt werden kann - so werden die nachfolgenden Bytes nicht belegt.

Im standardisierten Format wird nur der skalierte Messwert und die 8 Zeichen lange Messwerteinheit übertragen.

**Meldezeitpunkt**

Der Meldezeitpunkt wird im HEX-Format im MMS-Format Time and Date (Bezugspunkt 1.1.84) übertragen. Im standardisiertem Format wird der Meldezeitpunkt als ASCII-Zeichenkette übertragen, die Datum (Tag, Monat, Jahr) und Uhrzeit (Stunde, Minute, Sekunde, Millisekunde) enthält. Datum und Uhrzeit sind

durch ein Leerzeichen voneinander getrennt. Die Zeichenkette ist 24 Zeichen lang (Beispiel: "01.05.1993 08:01:15:0045").

**Meldetext** Der Meldetext wird immer als ASCII-Zeichenkette übertragen. Dabei wird keine Längenangabe übertragen. Diese ist aus der Gesamtanzahl der empfangenen Daten zu berechnen. Der Meldetext kann maximal 60 Zeichen lang sein.

### 3.11.3.3 Aufbau einer Überlaufmeldung

**Überlaufmeldung** Tritt ein Überlauf des Meldfolgepuffers auf, so wird vom Baustein MSI / MSPR eine Überlaufmeldung generiert:

- Die Überlaufmeldung hat den Meldungstyp Warnung ('W').
- Das Präfix enthält den Wert des Anschlusses RP des Funktionsbausteins MSI, der die Meldung generiert.
- Das Suffix enthält die Anzahl der verlorengegangenen Meldungen.
- Es ist kein Messwert vorhanden. Dies wird in der Meldetypbeschreibung angezeigt.
- Als Meldezeitpunkt wird der Zeitpunkt eingetragen, an dem der Meldeauswertebaustein die Überlaufmeldung generiert hat.
- Als Meldetext wird, falls der Anschluss STM des Funktionsbausteins MSI gesetzt ist, der Text "sequence buffer overflow" ausgegeben.

### 3.11.3.4 Aufbau einer Kommunikationsfehlermeldung

**Kommunikationsfehlermeldung** Der Zentralbaustein wertet die im System auftretenden Kommunikationsfehler aus und generiert folgende Kommunikationsfehlermeldungen:

- Eine Kommunikationsfehlermeldung hat den Meldungstyp C-Fehler ('C').
- Das Präfix enthält den Wert des Anschlusses RP des Zentralbausteins, der die Meldung generiert hat.
- Das Suffix enthält die (immer positive) Fehlernummer der C-Fehlermeldung.
- Wenn ein Messwert nicht vorhanden ist, dann wird dies in der Meldetypbeschreibung angezeigt.
- Als Meldetext wird, falls der Anschluss STM des Funktionsbausteins MSI gesetzt ist, der am Zentralbaustein am Anschluss CMT projektierte Text ausgegeben.
- Ist ein Überlauf des Kommunikationsfehlerfeldes aufgetreten, so wird nach Ausgabe aller C-Fehlermeldungen eine Meldung generiert, die als Suffix die negative Anzahl der verlorengegangenen Meldungen enthält. Nach dieser Meldung wird keine weitere C-Fehlermeldung

vom MSI ausgegeben. Als Meldezeitpunkt wird der Zeitpunkt eingetragen, an der der Zentralbaustein den Überlauf des Kommunikationsfehlerfeldes entdeckt hat.

### 3.11.3.5 Aufbau einer Systemfehlermeldung

#### Systemfehlermeldung

Eine Systemfehlermeldung hat prinzipiell den gleichen Aufbau wie eine Kommunikationsfehlermeldung. Änderungen bestehen hinsichtlich des Meldetextes, für den immer "system message" eingesetzt wird, sowie dem Meldetyp ('S'). Des weiteren wird maximal eine Systemfehlermeldung generiert, die in der Initialisierungsphase des Zentralbausteins erkannt wird.

Als Suffix wird eine Kennung über den Systemfehler eingetragen, der folgende Bedeutung hat:

Wert Suffix	Bedeutung
1	Spannungsausfall 5 V
2	Spannungsausfall 15 V
3	Softwareablauf gestört
4	Fehler bei Zugriff auf L-Bus-Koppelspeicher
5	Fehler bei Zugriff auf C-Bus-Koppelspeicher
6	Fehler bei Zugriff auf Standardperipherie
7	Fehler bei Zugriff auf Sonderperipherie
8	undefinierter L-Bus-Zugriff
9	undefinierter C-Bus-Zugriff
10	(not used)
11	nicht identifizierbarer HW-Fehler
12	(not used)
13	nicht identifizierbarer Fehler
14	Fehlermeldung (Ready Intern) vom lokalen Erweiterungsbus (LE-Bus)
15	Fehler bei Zugriff auf lokale Peripherie (LP-Bus)
16	Overrun des System-Bus-Controllers

Tabelle 3-28 Suffix Systemfehlermeldung

### 3.11.3.6 Detaillierte Beschreibung der Meldungsformate des Funktionsbausteins MSI

#### Allgemeines

Die Beschreibung der Meldungsformate besteht aus 3 Teilen:

- Belegung der Initialisierungsanschlüsse SNV, STM und SSF
- Grundsätzliches Format und maximale Länge der Meldung. Diese Länge entspricht der Größe des vom MSI angemeldeten Kanals.



- Aufbau der Nutzdatenstruktur, die für die Kanalinitialisierung benötigt wird.
- Der Anschluss STC fehlt in dieser Aufstellung. Bei STC=1 stimmt die Längenangabe beim Meldungstext immer mit der Maximallänge überein, bei STC=0 entspricht sie der tatsächlichen Länge des Meldungstextes.

<b>SNV=TRUE (Meldungsnummern vorhanden)</b> <b>STM=TRUE (Meldungstext vorhanden)</b> <b>SSF=TRUE (Standardisiertes Format)</b>				
Inhalt	Meldungsstruktur (max. 108 Bytes)	Nutzdatenstruktur	Datenformat	Anzahl der Daten
Spontankennung	Unsigned8	1. variable Einheit	Unsigned8	2
Sequenznummer	Unsigned8			
Meldetypbeschreibung	1 Octet	2. variable Einheit	Octet-String	2
Meldungstyp	1 Octet			
Präfix	Floating-Point	3. variable Einheit	Floating-Point	3
Suffix	Floating-Point			
Messwert	Floating-Point			
Messwertdimensionstext	8 Zeichen	4. variable Einheit	Visible-String	92
Meldezeitpunkt	24 Zeichen			
Meldungstext	max. 60 Zeichen			

Tabelle 3-29 Standardformat mit Nummer und Text

<b>SNV=FALSE (Meldungsnummern nicht vorhanden)</b> <b>STM=TRUE (Meldungstext vorhanden)</b> <b>SSF=TRUE (Standardisiertes Format)</b>				
Inhalt	Meldungsstruktur (max. 100 Bytes)	Nutzdatenstruktur	Datenformat	Anzahl der Daten
Spontankennung	Unsigned8	1. variable Einheit	Unsigned8	2
Sequenznummer	Unsigned8			
Meldetypbeschreibung	1 Octet	2. variable Einheit	Octet-String	2
Meldungstyp	1 Octet			
Messwert	Floating-Point	3. variable Einheit	Floating-Point	1
Messwertdimensionstext	8 Zeichen	4. variable Einheit	Visible-String	92
Meldezeitpunkt	24 Zeichen			
Meldungstext	max. 60 Zeichen			

Tabelle 3-30 Standardformat ohne Nummer mit Text

<b>SNV=TRUE (Meldungsnummern vorhanden)</b> <b>STM=FALSE (Meldungstext nicht vorhanden)</b> <b>SSF=TRUE (Standardisiertes Format)</b>				
<b>Inhalt</b>	<b>Meldungsstruktur (max. 48 Bytes)</b>	<b>Nutzdatenstruktur</b>	<b>Datenformat</b>	<b>Anzahl der Daten</b>
Spontankennung	Unsigned8	1. variable Einheit	Unsigned8	2
Sequenznummer	Unsigned8			
Meldetypbeschreibung	1 Octet	2. variable Einheit	Octet-String	2
Meldungstyp	1 Octet			
Präfix	Floating-Point	3. variable Einheit	Floating-Point	3
Suffix	Floating-Point			
Messwert	Floating-Point			
Messwertdimensionstext	8 Zeichen	4. variable Einheit	Visible-String	32
Meldezeitpunkt	24 Zeichen			

Table 3-31 Standardformat mit Nummer ohne Text

<b>SNV=FALSE (Meldungsnummern nicht vorhanden)</b> <b>STM=FALSE (Meldungstext nicht vorhanden)</b> <b>SSF=TRUE (Standardisiertes Format)</b>				
<b>Inhalt</b>	<b>Meldungsstruktur (max. 48 Bytes)</b>	<b>Nutzdatenstruktur</b>	<b>Datenformat</b>	<b>Anzahl der Daten</b>
Spontankennung	Unsigned8	1. variable Einheit	Unsigned8	2
Sequenznummer	Unsigned8			
Meldetypbeschreibung	1 Octet	2. variable Einheit	Octet-String	2
Meldungstyp	1 Octet			
Messwert	Floating-Point	3. variable Einheit	Floating-Point	1
Messwertdimensionstext	8 Zeichen	4. variable Einheit	Visible-String	32
Meldezeitpunkt	24 Zeichen			

Table 3-32 Standardformat ohne Nummer und Text

<b>SNV=TRUE (Meldungsnummern vorhanden)</b> <b>STM=TRUE (Meldungstext vorhanden)</b> <b>SSF=FALSE (HEX-Format)</b>				
Inhalt	Meldungsstruktur (max. 92 Bytes)	Nutzdatenstruktur	Datenformat	Anzahl der Daten
Spontankennung	Unsigned8	1. variable Einheit	Unsigned8	2
Sequenznummer	Unsigned8			
Meldetypbeschreibung	1 Octet	2. variable Einheit	Octet-String	2
Meldungstyp	1 Octet			
Präfix	Unsigned16	3. variable Einheit	Unsigned16	2
Suffix	Unsigned16			
Messwert-Skalierfaktor	Floating-Point	4. variable Einheit	Floating-Point	1
Messwert	4 Octets	5. variable Einheit	Octet-String	6
Messwert-Datentyp	2 Octets			
Messwertdimensionstext	8 Zeichen	6. variable Einheit	Visible-String	8
Meldezeitpunkt	Time and Date	7. variable Einheit	Time and Date	1
Meldungstext	max. 60 Zeichen	8. variable Einheit	Visible-String	60

Tabelle 3-33 HEX-Format mit Nummer und Text

<b>SNV=FALSE (Meldungsnummern nicht vorhanden)</b> <b>STM=TRUE (Meldungstext vorhanden)</b> <b>SSF=FALSE (HEX-Format)</b>				
Inhalt	Meldungsstruktur (max. 88 Bytes)	Nutzdatenstruktur	Datenformat	Anzahl der Daten
Spontankennung	Unsigned8	1. variable Einheit	Unsigned8	2
Sequenznummer	Unsigned8			
Meldetypbeschreibung	1 Octet	2. variable Einheit	Octet-String	2
Meldungstyp	1 Octet			
Messwert-Skalierfaktor	Floating-Point	3. variable Einheit	Floating-Point	1
Meßwert	4 Octets	4. variable Einheit	Octet-String	6
Messwert-Datentyp	2 Octets			
Messwertdimensionstext	8 Zeichen	5. variable Einheit	Visible-String	8
Meldezeitpunkt	Time and Date	6. variable Einheit	Time and Date	1
Meldungstext	max. 60 Zeichen	7. variable Einheit	Visible-String	60

Tabelle 3-34 HEX-Format ohne Nummer mit Text

<b>SNV=TRUE (Meldungsnummern vorhanden)</b> <b>STM=FALSE (Meldungstext nicht vorhanden)</b> <b>SSF=FALSE (HEX-Format)</b>				
Inhalt	Meldungsstruktur (max. 32 Bytes)	Nutzdatenstruktur	Datenformat	Anzahl der Daten
Spontankennung	Unsigned8	1. variable Einheit	Unsigned8	2
Sequenznummer	Unsigned8			
Meldetypbeschreibung	1 Octet	2. variable Einheit	Octet-String	2
Meldungstyp	1 Octet			
Präfix	Unsigned16	3. variable Einheit	Unsigned16	2
Suffix	Unsigned16			
Messwert-Skalierfaktor	Floating-Point	4. variable Einheit	Floating-Point	1
Messwert	4 Octets	5. variable Einheit	Octet-String	6
Messwert-Datentyp	2 Octets			
Messwertdimensionstext	8 Zeichen	6. variable Einheit	Visible-String	8
Meldezeitpunkt	Time and Date	7. variable Einheit	Time and Date	1

Tabelle 3-35 HEX-Format mit Nummer ohne Text

<b>SNV=FALSE (Meldungsnummern nicht vorhanden)</b> <b>STM=FALSE (Meldungstext nicht vorhanden)</b> <b>SSF=FALSE (HEX-Format)</b>				
Inhalt	Meldungsstruktur (max. 28 Bytes)	Nutzdatenstruktur	Datenformat	Anzahl der Daten
Spontankennung	Unsigned8	1. variable Einheit	Unsigned8	2
Sequenznummer	Unsigned8			
Meldetypbeschreibung	1 Octet	2. variable Einheit	Octet-String	2
Meldungstyp	1 Octet			
Messwert-Skalierfaktor	Floating-Point	3. variable Einheit	Floating-Point	1
Messwert	4 Octets	4. variable Einheit	Octet-String	6
Messwert-Datentyp	2 Octets			
Messwertdimensionstext	8 Zeichen	5. variable Einheit	Visible-String	8
Meldezeitpunkt	Time and Date	6. variable Einheit	Time and Date	1

Tabelle 3-36 HEX-Format ohne Nummer und Text

### 3.11.3.7 Ausgabeformat des Meldeauswertebausteins MSIPRI

#### Allgemeines

Gegenüber dem Meldeauswertebaustein MSI ist das Format der Meldungen des Auswertebausteins MSIPRI nicht frei wählbar. Hier wird nur ein Format ausgegeben. Entsprechend entfallen bei der Projektierung des Bausteins die Anschlüsse zur Formatwahl. Der Baustein MSIPRI ist speziell zur Ausgabe der Meldungen auf einen Drucker entwickelt worden. Alle Meldungen werden in textueller Form ausgegeben und mit

Zeilenvorschub formatiert. Eine Meldung besteht aus maximal zwei Zeilen.

### Aufbau der 1. Zeile

Zeichen der 1. Zeile	Bedeutung	Ausgabeformat
1-24	Datum/Uhrzeit	Tag.Monat.Jahr, Stunde:Minute: Sekunde: Millisekunde
25-27	Text: "P:"	
28-32	Präfix	maximal 5 Zeichen und rechtsbündig
33-35	Text: "S:"	
36-40	Suffix	maximal 5 Zeichen und rechtsbündig
41-45	Text: "Typ:"	
46	Meldungstyp ('C','F','W' oder 'S')	ein Zeichen
47-50	Text: "Nr:"	
51-53	Sequenznummer	maximal 3 Zeichen und rechtsbündig
54	Text: " "	
55-67	Messwert (optional: wird nur eingetragen, falls die Meldung einen Messwert enthält)	Wird als Float-Wert in folgender Reihenfolge ausgegeben: <ul style="list-style-type: none"> <li>• Vorzeichen (positiv = „+“, negativ = „-“)</li> <li>• Vorkomastelle, gefolgt von einem Punkt sowie 6 Nachkommastellen</li> <li>• Exponent, eingeleitet durch das Zeichen 'e'</li> <li>• Vorzeichen (positiv = „+“, negativ = „-“) sowie 2 Exponentstellen</li> </ul>
68	Leerzeichen (optional)	
69-76	Messwerteinheit (optional: wird nur eingetragen, falls die Meldung einen Messwert enthält)	8 Zeichen
77, 78	Sonderzeichen CR und LF	Zeilenvorschub

Tabelle 3-37 Aufbau der Meldung des Auswertebausteins MSIPRI 1. Zeile

**Aufbau der 2. Zeile** Die zweite Zeile enthält den Meldetext und wird nur ausgegeben, falls ein Meldetext vorhanden ist. Ansonsten entfällt diese komplett.

Zeichen der 2. Zeile	Bedeutung	Ausgabeformat
1-60	Messwerttext (optional)	variable Länge
61, 62	Sonderzeichen CR und LF	Zeilenvorschub

Tabelle 3-38 Aufbau der Meldung des Auswertebausteins MSIPRI 2. Zeile

### Beispiel für Meldungsausgabe

"01.05.1993 08:01:15:0045 P: 123 S: 10 Typ: W Nr: 25 -1.123456e+12 ms " "Dies ist ein Meldetext"
---

Tabelle 3-39 Beispiel für Meldungsausgabe

---

**HINWEIS** Überlauf-, Kommunikationsfehler- und Systemfehlermeldungen haben den selben logischen Aufbau wie der Baustein MSI.

---

## 3.12 Kommunikations-Dienst Prozessdaten

**Anwendung** Der Kommunikations-Dienst Prozessdaten unterstützt die „reine“ Datenübertragung in Sende- und Empfangsrichtung, d.h. die Funktionsbausteine übertragen ausschließlich Prozessdaten. Die Daten selbst werden weder ausgewertet, noch logisch interpretiert.

Für die Datenübertragung gibt es zwei Bausteinklassen:

- Empfangs- und Sendebausteine: CRV und CTV
- Kanal-Rangierbausteine: CCC4 und CDC4

Die Bausteine CRV und CTV decken die meisten Kommunikationsanwendungen ab.

### 3.12.1 Empfangs- und Sendebausteine

**Allgemeines** Es gibt jeweils einen Empfangs- und Sendebaustein. Sie heißen CRV (communication receive virtual) und CTV (communication transmit virtual).

Mit einem Empfangs- oder Sendebaustein wird ein Telegramm projiziert, das von oder zu einer Kopplungsbaugruppe übertragen wird. Struktur und Inhalt des Telegramms werden durch die Projektierung von virtuellen Verbindungen festgelegt.

#### 3.12.1.1 Virtuelle Verbindungen

**Allgemeines** Eine virtuelle Verbindung ist eine „unsichtbare“ Verbindung zwischen Bausteinanschlüssen. Auf der Projektierungsoberfläche wird keine Verbindung gezogen, sondern es entsteht nur eine Randleistenverbindung.

Welche Werte von Bausteinausgängen oder an Bausteineingänge übertragen werden, legt der Projektierer durch die „Verbindungsangabe Empfangen/Senden“ an den Empfängern oder Sendern und der „virtuellen Verbindungsangabe“ und der „Reihenfolgennummer“ an den zu bearbeitenden Bausteingängen oder -ausgängen fest.

**Verbindungsangabe** Die Verbindungsangabe besteht aus einem Ausrufezeichen („!“) und maximal 6 Zeichen (Großbuchstaben oder Ziffern). Die Zeichenkette steht direkt hinter dem Ausrufezeichen (z.B. „!SEND“). Das Ausrufezeichen muss nicht explizit projiziert werden, es wird automatisch generiert.

Eine virtuelle Verbindung besteht aus:

- virtuelle Verbindungsangabe
- Reihenfolgennummer

Verbindungsangabe und Reihenfolgennummer sind durch einen Punkt voneinander getrennt (z.B. „ISEND.0056“; der Punkt zwischen der Verbindungsangabe und der Reihenfolgennummer muss nicht explizit projiziert werden, er wird automatisch generiert).

**Datentypen**

Virtuelle Verbindungen können an Anschlüssen mit folgenden Datentypen projiziert werden:

- BOOL (BO), BYTE (BY)
- WORD (W), DOUBLE WORD (DW)
- INTEGER (I), DOUBLE INTEGER (DI)
- REAL ® und SDTIME (TS)

---

**HINWEIS**

An Anschlüssen vom Datentyp STRING (S) oder GLOBAL VARIABLE (GV) können keine virtuellen Verbindungen projiziert werden.

---

**Telegrammstruktur**

Die virtuellen Verbindungen mit derselben Verbindungsangabe definieren ein Telegramm mit einer bestimmten Struktur. Die Reihenfolge der Daten innerhalb des Telegramms ist durch die Reihenfolgennummer festgelegt. Das Datum mit der niedrigsten Nummer steht zu Beginn des Telegramms, das mit der höchsten am Ende. Die Reihenfolgennummer legt die relative Position des Datums im Telegramm fest. Lücken bei den Reihenfolgennummern werden ignoriert.



Empfangen und Senden mit virtuellen Verbindungen.

**Beispiel-Projektierung**

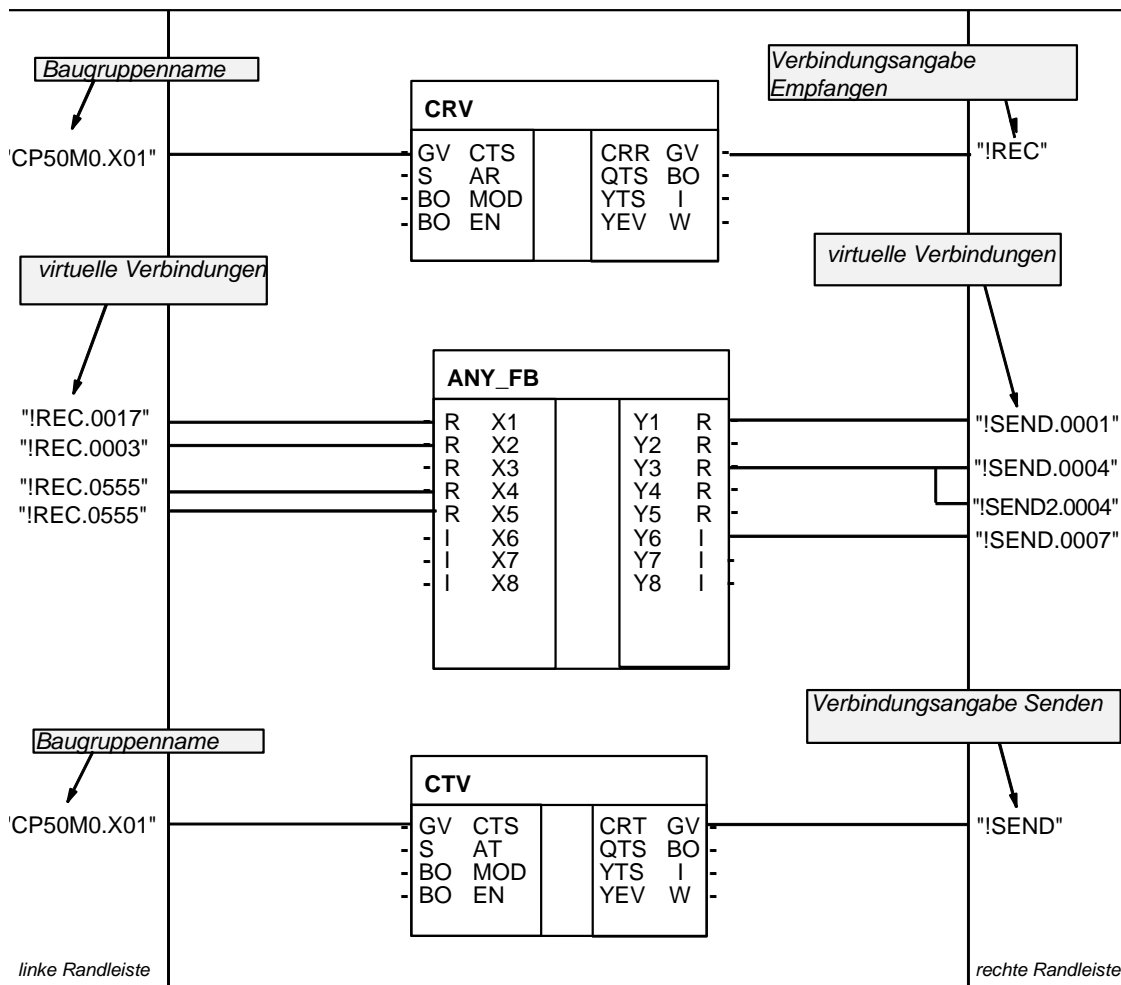


Bild 3-83 Projektierung: Empfangen und Senden mit virtuellen Verbindungen

Projektierungsregeln mit Verweis auf das Beispiel:

- Die virtuellen Verbindungen, die zu einer virtuellen Verbindungsangabe gehören, können an Bausteinanschlüssen mit unterschiedlichen Datentypen (ANY\_FB.Y1 „REAL“ und ANY\_FB.Y6 „INTEGER“) und in beliebiger Reihenfolge projektirt werden.
- Virtuelle Verbindungen (Empfang) an Bausteineingängen können mehrfach projektirt werden, wenn die Eingänge vom gleichen Datentyp sind. Diese Eingänge werden mit identischen Daten versorgt (ANY\_FB.X4 und X5).
- Dieselbe virtuelle Verbindung (Senden) mit identischer Verbindungsangabe und Reihenfolgennummer darf nicht mehrfach an Bausteinausgängen projektirt werden.

- An einem Bausteinausgang können mehrere verschiedene virtuelle Verbindungen (Senden) projektiert werden (ANY\_FB.Y3). Die Verbindungen können sich sowohl in der Verbindungsangabe als auch in der Reihenfolgennummer unterscheiden.

Telegrammstruktur der Verbindungsangabe „!REC“ aus dem Beispiel:

<b>Anschluss</b>	<b>virtuelle Verbindung</b>	<b>Datentyp</b>	<b>Länge</b>
ANY_FB.X2	!REC.0003	R	4
ANY_FB.X1	!REC.0017	R	4
ANY_FB.X4/X5	!REC.0555	R	4
			Gesamtlänge = 12 Byte

Tabelle 3-40 Telegrammstruktur der Verbindungsangabe „!REC“

Telegrammstruktur der Verbindungsangabe „!SEND“ aus dem Beispiel:

<b>Anschluss</b>	<b>virtuelle Verbindung</b>	<b>Datentyp</b>	<b>Länge</b>
ANY_FB.Y1	!SEND.0001	R	4
ANY_FB.Y3	!SEND.0004	R	4
ANY_FB.Y6	!SEND.0007	I	2
			Gesamtlänge = 10 Byte

Tabelle 3-41 Telegrammstruktur der Verbindungsangabe „!SEND“

Die Struktur der projektierten Telegramme erscheint in ähnlicher Form in den CFC-Referenzdaten in der Ansicht „Querverweise Operanden“ oder im CPU-Maplisting (des CFC) unter dem Stichwort „virtuelle Verbindungen“. Anhand dieser Listen kann die Projektierung überprüft werden.

- 
- HINWEIS**
- Die virtuellen Verbindungsangaben sind CPU-lokal bekannt. Es können Daten aus verschiedenen Funktionsplänen zu einem Telegramm zusammengefasst werden, nicht aber von verschiedenen CPUs.
  - Die Bearbeitung der Daten geschieht durch die Empfangs-/Sendebausteine in deren Abtastzeit. Die Abtastzeiten der Bausteine mit den virtuellen Verbindungen haben keinen Einfluss auf den Bearbeitungszyklus der Telegramme.
  - Es ist Aufgabe des Projektors dafür zu sorgen, dass die Telegrammstruktur und -länge mit dem Koppelpartner kompatibel ist (fehler Kapitel Funktionsweise der Kopplungen). Diese Vorschrift ist abhängig von der unterlagerten Kopplung. Im Fehlerfall schaltet sich der Empfang-/Sendebaustein mit einem Eintrag ins Kommunikations-Fehlerfeld ab (z.B. PROFIBUS DP oder Baugruppenträgerkopplung) oder es kommt keine Kommunikation zustande (z.B. INDUSTRIAL ETHERNET).
- 

### 3.12.1.2 Anschlüsse der Bausteine CRV, CTV

- Anschluss CTS** Am Anschluss CTS des Bausteins wird der projektierte Name der Kopplungsbaugruppe angegeben, über die die Kommunikation erfolgen soll. Bei den Baugruppentypen CP50M0/CP51M1 ist eine zusätzliche Steckerangabe (X01 bei CP50M0/CP51M1 oder X02 bei CP50M0) notwendig.
- Anschluss AR, AT** Am Anschluss AR, AT wird der Adressparameter für die Kommunikation angegeben. Er besteht aus einem Kanalnamen und den optionalen Adressstufen. Die Bedeutung der Adressparameter ist abhängig von der verwendeten Kopplung (z.B. PROFIBUS oder DUST).
- Eingang MOD** Am Eingang MOD wird der Übertragungsmodus projektiert (z.B. „R“ für Refresh oder „H“ für Handshake).
- Eingang EN** Der Eingang EN legt fest, ob die Daten im aktuellen Arbeitszyklus übertragen werden oder nicht.
- Anschluss CRR, CRT** Am Anschluss CRR bzw. CRT wird die virtuelle Verbindungsangabe Empfang bzw. Senden projektiert.

### 3.12.2 Kanal-Rangierbausteine CCC4 und CDC4

- Anwendung** Zum Aufspalten oder Zusammenfassen von Kanälen werden die Kanal-Rangierbausteine eingesetzt.

### 3.12.2.1 Sammelbaustein CCC4

- Allgemeines** Der Funktionsbaustein CCC4 (Communication Collect Channel 4) fasst bis zu 4 Kanäle zu einem zusammen. Die Kanäle dürfen unterschiedliche Adressangaben besitzen, auf unterschiedlichen Datenschnittstellen liegen und unterschiedliche Übertragungsmodi sowie Kanallängen besitzen.
- Voraussetzungen** Damit der Funktionsbaustein arbeiten kann, müssen mindestens 2 Kanäle zusammengefasst werden (CT1- und CT2-Anschlußangaben sind unbedingt notwendig).
- Angabe an Anschlüssen CT3, CT4** Wenn nur 2 Kanäle zusammengefasst werden sollen, so ist an den Initialisierungsanschlüssen CT3 und CT4 eine „0“ (Null) zu projektieren. Die Anschlüsse AR3, AR4, MO3, MO4, LT3 und LT4 werden dann in diesem Fall nicht mehr ausgewertet.
- Angabe an Anschlüssen CTS, AT, MOD** An den Anschlüssen CTS, AT und MOD wird der Sendekanal spezifiziert. Die Länge der zu sendenden Nutzdaten ergibt sich aus der Summe der Empfangsdaten. Die Empfangskanäle 1-4 werden der Reihe nach zu einem großen Nutzdatenblock zusammengefügt.

#### Beispiel

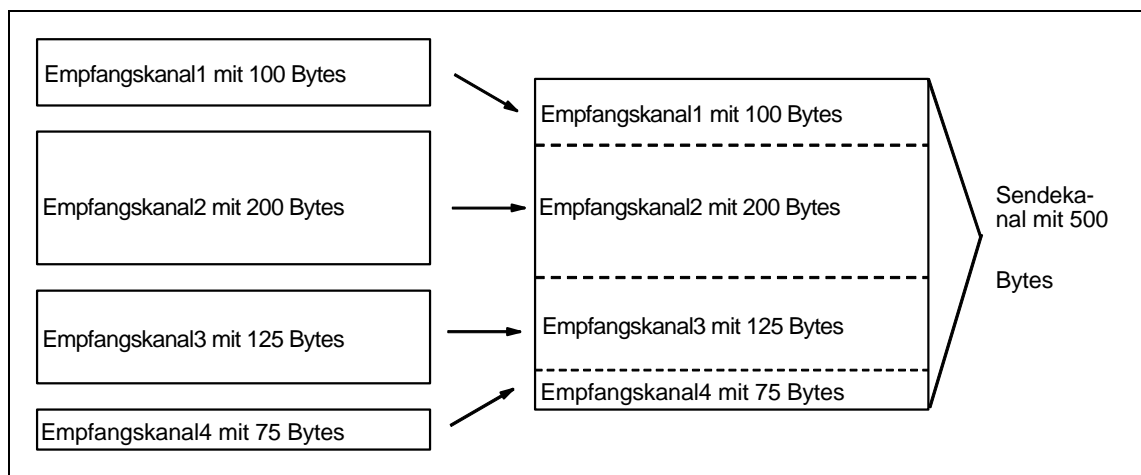


Bild 3-84 Zusammenfassung von 4 Empfangskanälen zu einem Sendekanal

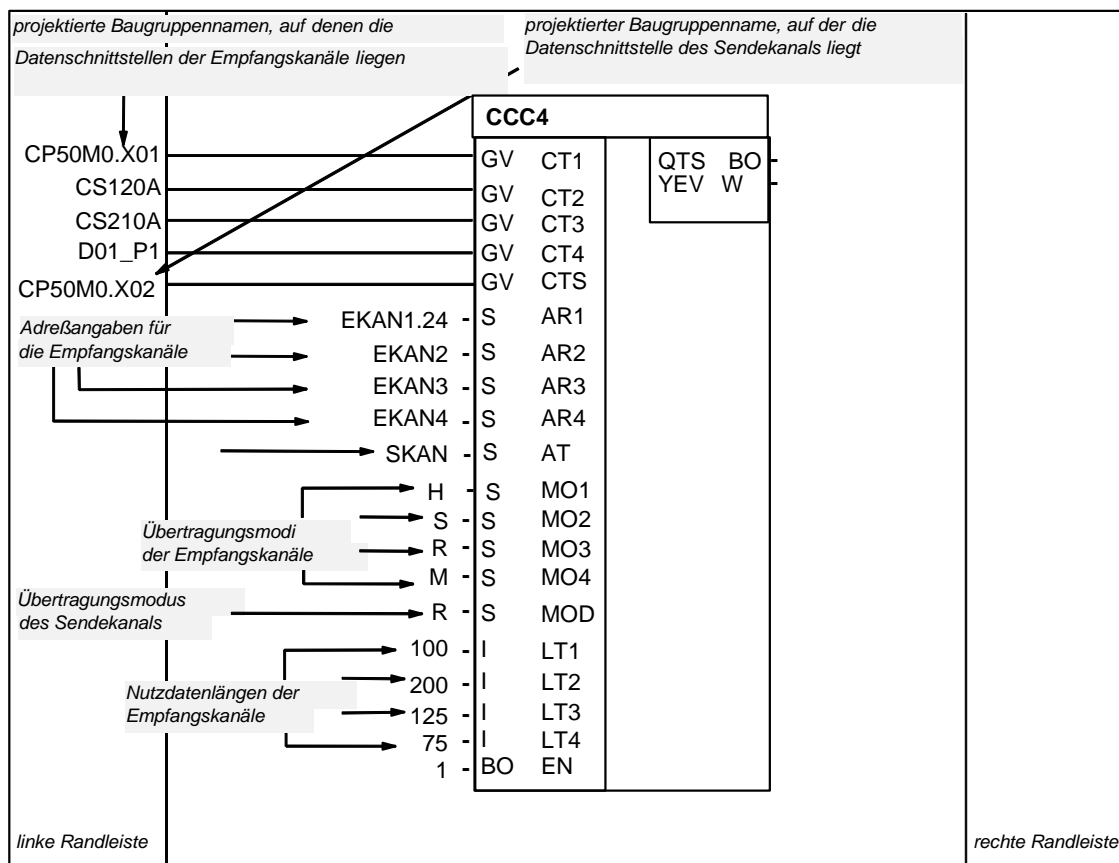


Bild 3-85 Beispiel-Projektierung: CCC4-Anschlüsse bei Zusammenfassung von 4 Kanälen

### 3.12.2.2 Verteilerbaustein CDC4

#### Allgemeines

Der Funktionsbaustein CDC4 (Communication Distribute Channel 4) teilt einen Kanal in bis zu 4 Kanäle auf. Die Kanäle dürfen unterschiedliche Adressangaben besitzen, auf unterschiedlichen Datenschnittstellen liegen und unterschiedliche Übertragungsmodi sowie Kanallängen besitzen.

#### Voraussetzungen

Damit der Funktionsbaustein arbeiten kann, muss der Empfangskanal in mindestens 2 Sendekanäle aufgeteilt werden (CT1- und CT2-Anschlußangabe sind unbedingt notwendig).

#### Angabe an den Anschlüssen CT3, CT4

Bei Aufteilung in nur 2 Kanäle ist an den Initialisierungs-Anschlüssen CT3 und CT4 eine „0“ (Null) zu projektieren. Die Anschlüsse AR3, AR4, MO3, MO4, LT3 und LT4 werden dann in diesem Fall nicht mehr ausgewertet.

#### Angabe an den Anschlüssen CTS, AT, MOD

An den Anschlüssen CTS, AT und MOD wird der Empfangskanal spezifiziert. Die Länge der zu empfangenden Nutzdaten ergibt sich aus der Summe der Sendedaten.

**HINWEIS**

Wenn einer der Sendekanäle im Modus Handshake projektiert wird und genau dieser Kanal auf der Empfangseite nicht ausgelesen wird, kann der CDC4-Funktionsbaustein solange nicht arbeiten, bis dieser eine Kanal ausgelesen wird. Der Baustein ist dann temporär blockiert.

**3.12.2.3 Kompatible Nutzdatenstruktur**

Die Nutzdaten sind bei den Bausteinen CCC4 und CDC4 unstrukturiert (Datentyp Octet-String). Sie sind damit zu jeder beliebigen Nutzdatenstruktur kompatibel. Damit sich Sender und zugehöriger Empfänger korrekt aufeinander synchronisieren, muss lediglich die Nutzdatenlänge identisch sein.

**3.12.3 Diagnoseausgänge**

**Allgemeines**

Am YEV-Ausgang der Sende- und Empfangsbausteine (CTV, CRV) sowie der Kanal-Rangierbausteine (CCC4, CDC4) wird nach jedem Bearbeitungszyklus das Ergebnis der Bearbeitung an der Datenschnittstelle(n) ausgegeben. Der YEV-Ausgang ist vom Typ WORD; die 16 Bits sind in drei Bereiche unterteilt:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Kanalzustände (nur CCC4, CDC4)				Kanalzuordnung (nur CCC4, CDC4)				Störungsursache							

Tabelle 3-42 Diagnoseausgänge

**3.12.3.1 Störungsursache**

**Hexadezimal-Wert**

In Bit 0-7 wird die eventuelle Störungsursache in Form eines Hexadezimal-Wertes (nicht bitcodiert auszuwerten) angezeigt:

Hex.-Wert	Bedeutung	Abhilfe
0	Keine Störung, Datenübertragung erfolgreich.	
1	Baustein dauerhaft abgeschaltet nach Initialisierung wegen Projektierungsfehler oder nach internem Fehler (genaue Meldung siehe Kommunikationsfehlerfeld oder YTS-Ausgang).	Projektierung korrigieren.
2	Kommunikationspartner nicht startklar oder Kommunikationsweg physikalisch unterbrochen. (genaue Meldung siehe YTS)	Koppelpartner, Leitungen und Stecker überprüfen.
3	Kommunikationspartner Sende-/Empfangspause (abhängig vom Enable-Eingang des Kommunikationspartners). Der Funktionsbaustein sendet/empfangt nicht, weil der Kommunikationspartner mitgeteilt hat, dass auch er keine Daten überträgt.	Kommunikationspartner aktivieren
4	Nur beim Sender: Es konnten keine Daten gesendet werden. (Normalerweise im Handshake-/Select-Modus: der Kommunikationspartner hat die letzten Daten noch nicht ausgelesen; selten im Refresh-Modus: Kommunikationspartner liest gerade).	Sender langsamer oder Empfänger schneller projektieren.

5	Nur beim Empfänger: Es konnten keine neuen Daten empfangen werden. (Der Kommunikationspartner hat seit dem letzten Datenempfang keine neuen Daten gesendet.)	Empfänger langsamer oder Sender schneller projektieren.
6	Daten inkonsistent (Baugruppenträgerkopplung: bei Abschalten des Master-Baugruppenträgers)	keine (Es geht von selbst weiter mit erneuter Initialisierung)
7	Nur Select-Sender: Kanal belegt. Es sendet gerade ein anderer Funktionsbaustein.	Alle Select-Sender über Enable-Eingang koordinieren.
8	Nur Multiple-Empfänger: Empfang fehlerhaft. Das Auslesen der Daten hat zu lange gedauert; der Sender hat zwischenzeitlich bereits neue Daten in den Kanal geschrieben.	Empfänger in einer schnelleren (höherpriorien) Abtastzeit projektieren.
9	Initialisierung läuft noch. Der Sende-/Empfangsbetrieb konnte deshalb noch nicht aufgenommen werden.	

Tabelle 3-43 Störungsursache

**Anmerkung zu Nummer 4 und 5**

Beim Modus Handshake ist ein sporadisches Auftreten dieser Nummern akzeptabel, da eine vollständige Synchronisierung zwischen den Kommunikationspartnern nicht immer möglich ist. Empfänger und Sender sollten ungefähr im gleichen Zyklus arbeiten.

Beim Refresh-Modus ist das Auftreten dieser Nummern zu vermeiden, wenn der Sender grundsätzlich schneller arbeitet als der Empfänger.

**3.12.3.2 Kanalzuordnung**

**Allgemeines**

Der Bereich wird nur bei den Funktionsbausteinen CCC4 und CDC4 benutzt. Hier wird durch eine Nummer angezeigt, auf welchen Kanal sich die Störungsursache (Bit 0-7) bezieht. Da die Kanal-Rangierbausteine bis zu fünf Kanäle bearbeiten können, ist die Nummerierung wie folgt:

Nummer	Kanal
0	Haupt-Sender-/Empfänger (entsprechend CTS-, AT- oder AR-Anschlußangabe).
1	Sende-/Empfangs-Teil 1 (entsprechend CT1- und AT1- oder AR1-Anschlußangabe)
2	Sende-/Empfangs-Teil 2
3	Sende-/Empfangs-Teil 3
4	Sende-/Empfangs-Teil 4

Tabelle 3-44 Kanalzuordnung

### 3.12.3.3 Kanalzustände

#### Allgemeines

Der Bereich wird nur bei den Funktionsbausteinen CCC4 und CDC4 benutzt. Hier wird angezeigt, welche Kanäle nicht störungsfrei bearbeitet werden konnten.

Im Bereich „Kanalzustände“ wird angegeben, auf welchen Kanälen insgesamt Störungen bei der Kanalbearbeitung festgestellt wurden.

Dieser Bereich ist bitorientiert aufgebaut:

- 1=keine Störung
- 0=Störung

Bit	Kanal
11	Sende-/Empfangs-Teil 1
12	Sende-/Empfangs-Teil 2
13	Sende-/Empfangs-Teil 3
14	Sende-/Empfangs-Teil 4
15	Haupt-Sender/-Empfänger

Tabelle 3-45 Kanalzustände

### 3.12.4 Einführung „Zeiger-basierte Kommunikationsbausteine

Die Projektierung von seriellen oder parallelen Datenübertragungen bei SIMATIC-Regelsystemen wurde bis zur D7-SYS-Version 6 mit der Methode der sog. „virtuellen Kommunikationsverbindungen“ vorgenommen (Darstellung in CFC-Plänen z. B.: „!VNAME.0001“).

Ausnahme: Die Lichtwellenleiter-Antriebskopplung SIMOLINK wird mit speziellen SIMOLINK-Bausteinen projektiert.

Ab D7-SYS Version 6 können Kommunikationsverbindungen wie z. B. PROFIBUS-DP, SIMATIC-CPU ↔ FM 458-1 DP sowie für SIMATIC TDC oder T400 und SIMADYN D alternativ auch mit Hilfe neu verfügbarer Kommunikations-Bausteine projektiert werden.

Dabei erfolgt der Zugriff auf die Schnittstellendaten von der CFC-Oberfläche aus mit Hilfe von neuen Bausteinen, die über eine spezielle Zeiger-Schnittstelle verbunden sind.

Beide Projektierungsweisen (virt. Verbindungen und zeiger-basierte Kommunikation) können auf derselben HW-Plattform, in derselben Projektierung und sogar für dieselbe Schnittstelle nebeneinander benutzt werden.



### 3.12.4.1 Prinzipielle Funktionsweise

Telegrammbausteine (CRV\_T, CTV\_P sowie S7RD\_P, S7WR\_P) ermöglichen einen Zugriff auf die empfangenen oder zu sendenden Datenblöcke (Telegramme) in dem sie einen Zeiger auf den jeweiligen Datenblock zur Verfügung stellen.

Dieser Zeiger wird auf Schreib-/Lesebausteine verdrahtet (DRD..., DWR...). Zusammen mit einer Offsetangabe kann dann ein Schreibbaustein das Datum an seinem Eingangsanschluss an der gewünschten Stelle im Puffer ablegen. Ein Lesebaustein holt entsprechend ein Datum aus der angegebenen Stelle des Empfangspuffers und stellt es an seinem Ausgang zur Verfügung.

Im Prinzip wird also eine virtuelle Verbindung durch einen (Schreib-/Lese-) Baustein und eine „normale“ CFC-Verbindung ersetzt.

### 3.12.4.2 Anwendungen

<b>Große Datenmenge</b>	Besonders vorteilhaft ist die zeiger-basierte Kommunikation bei einer sehr großen Anzahl von den Daten. Große Datenmengen sind deutlich einfacher und schneller zu projektieren, zu ändern und flexibler zu verschalten.
<b>Zugriff auf E/A-Bereich (P-Bus) bei FM 458-1 DP</b>	<p>Über den E/A-Bereich des P-Bus können je Richtung 128 Bytes von der FM 458-1 DP zur S7-CPU übertragen werden.</p> <p>Mit den neuen Bausteinen S7RD_P/S7WR_P können alle 128 Bytes mit einem Baustein rechenzeit-optimiert in einen Puffer kopiert werden, auf den über die Zeiger-Schnittstelle mit den Schreib-/Lesebausteinen flexibel und indiziert zugegriffen werden kann.</p> <p>Mit Offset- und Längenangaben kann auch auf Teilbereiche zugegriffen werden.</p>
<b>Daten in einem Datenblock speichern</b>	Daten können in einem universell verwendbarer Datenspeicher abgelegt werden, auf den über Zeiger-Schnittstelle mit Schreib-/Lesebausteinen zugegriffen werden kann. Da in diesem Datenblock auch mehrere gleichartige Puffer angelegt werden können, lassen sich z. B. leicht Rezepturen speichern und abrufen.

### 3.12.4.3 Merkmale der zeiger-basierten Kommunikation

- Reduzierung des Projektierungsaufwands bei der CFC-Planerstellung, insbesondere wenn sehr viele virtuelle Verbindungen erstellt werden müssten.
- Verbindungen zu den Telegrammdaten können online neu eingefügt und geändert werden (Zeiger, Puffer-Offset).

- Kommunikationsverbindungen sind mit bzw. innerhalb von Plan-Bausteinen kopierbar und mit ihnen zentral änderbar. Damit sind besonders leicht und schnell z. B. gleichartige Kommunikationsverbindungen zu einer großen Anzahl von Antrieben projektierbar.
- Mit Hilfe von 2 Offset-Angaben kann auf die Telegrammpufferdaten indiziert zugegriffen werden. Damit können sehr einfach modulare Programme (z. B. Plan-Bausteine) erstellt und genutzt werden.
- Größere Datenmengen können transparent (z. B. blockweise) verarbeitet (kopiert) werden, z. B. mit Kopierbaustein CPY\_P in den Datenblock DB\_P.
- für FM 458-1 DP:
  - Über K-Bus können mit „B-Receive“ (BRCV) sehr große Datenmengen von S7-CPU zur FM 458-1 DP übertragen werden.
  - Über den E/A-Bereich des P-Bus können 128 Bytes einfach zu projektieren und schnell sowie rechenzeit-arm transferiert werden.
- Für jeden Datentyp (BYTE, INT, DINT, REAL) steht ein spezieller Schreib-/Lesebaustein zur Verfügung.
- Vor dem Zugriff auf REAL-Daten wird der Typ überprüft.
- Diese Projektierungsmöglichkeiten sind prinzipiell bei allen Plattformen der SIMATIC-Regelsysteme einsetzbar, also FM 458-1 DP, SIMATIC TDC, T400 und SIMADYN D da die Baustein-Verarbeitung unabhängig von der unterlagerten HW ist.
- Aus gleichem Grund ist diese Bausteinkommunikation prinzipiell für alle Arten der seriellen und parallelen Übertragungswege nutzbar, bei denen heute die „virt. Kommunikation“ eingesetzt wird.

**Für alle Plattformen und Schnittstellen der SIMATIC-Regelsysteme**

### 3.12.4.4 Zugehörige Funktionsbausteine

Im Bausteinkatalog des CFC sind die einsetzbaren Bausteine unter dem Familiennamen „ZeigrKom“ bzw „PointCom“ eingeordnet.

Zur einfachen Identifizierung und leichten Zuordnung zu dieser Bausteingruppe erhalten die Bausteine, die in ihrer Funktion bereits vorhandenen Bausteinen entsprechen und die für diese Anwendung nun einen Zeiger herausgeben, am Namensende „\_P“ (Pointer).

Typ-Name	Funktion
CPY_P	Kopieren von Pufferbereichen
CRV_P	Telegrammbaustein Empfang (Schnittstellen-Verarbeitung)
CTV_P	Telegrammbaustein Senden (Schnittstellen-Verarbeitung)
DB_P	Datenblock
DRD	Data Read REAL
DRD_D	Data Read DINT

Typ-Name	Funktion
DRD_I	Data Read INT
DRD_8	Data Read 8*REAL
DRD_8D	Data Read 8*DINT
DRD_8I	Data Read 8*DINT
DRD_BY	Data Read BYTE
DWR	Data Write REAL
DWR_D	Data Write DINT
DWR_I	Data Write INT
DWR_8	Data Write 8*REAL
DWR_8D	Data Write 8*DINT
DWR_8I	Data Write 8*INT
DWD_BY	Data Write BYTE
S7RD_P	Empfang 128 Bytes über P-Bus (nur für FM 458-1 DP)
S7WR_P	Senden 128 Bytes über P-Bus (nur für FM 458-1 DP)
BRCV	Block-Datenempfang über S7-Verbindung (nur für FM 458-1 DP)

### 3.12.4.5 Zeiger-Schnittstelle

Bei der zeiger-basierten Kommunikation wird **ein Zeiger auf den Telegrammdatenpuffer** zwischen den beteiligten Bausteinen übergeben:

Dieser Zeiger ist tatsächlich ein Zeiger auf eine Struktur, die neben dem Zeiger auf die Nutzdaten auch Informationen für Überwachungszwecke enthält, wie z. B. Abtastzeit, Bausteinklasse, Byte-/Wort-Drehung. Er besitzt den Anschlusskommentar „ZeigPuffer“.

### 3.12.4.6 Projektierungshinweise

- Die Telegrammbausteine sowie die Schreib-/Lese-Bausteine müssen in **derselben Abtastzeit** projektiert werden, um Konsistenz sicher zu stellen. (Überprüfung bei Initialisierung).
- Die **Offsetangaben** müssen sorgfältig vorgenommen werden.
  - a) Bei der zeiger-basierten Kommunikation muss der Projektteur sehr genau auf den Offset (in Byte) des zu adressierenden 16bit-Wertes (INT) bzw. 32bit-Wertes (REAL, DINT) achten.
  - b) Der Offset muss immer kleiner als die Puffergröße sein. Vor einem Zugriff auf Pufferdaten wird eine eventuelle Bereichsüberschreitung durch zu groß eingestellte Offsets überprüft.

- Wenn Daten zu einer PROFIBUS-DP-Station oder zur SIMATIC-CPU transferiert werden, müssen Bytes (bei INT) und ggf. Worte des zu übertragenen Wertes (bei REAL, DINT) gedreht werden. Dazu besitzen die Schreib-/Lesebausteine einen „Swap“-Anschluss SWP.
- Um Telegramme über eine Schnittstelle zu übertragen, ist es ausreichend, zunächst nur die Telegrammbausteine mit einer entsprechenden Längenangabe zu projektieren (CRV\_T, CTV\_P sowie S7RD\_P, S7WR\_P). Es müssen noch keine Schreib-/Lesebausteine projiziert sein. Damit lässt sich z. B. mit sehr wenig Aufwand ein Test der Schnittstelle oder die Rechenzeitbelastung durch die Schnittstelle projektieren.

### 3.12.4.7 Beispiele als CFC-Screenshots

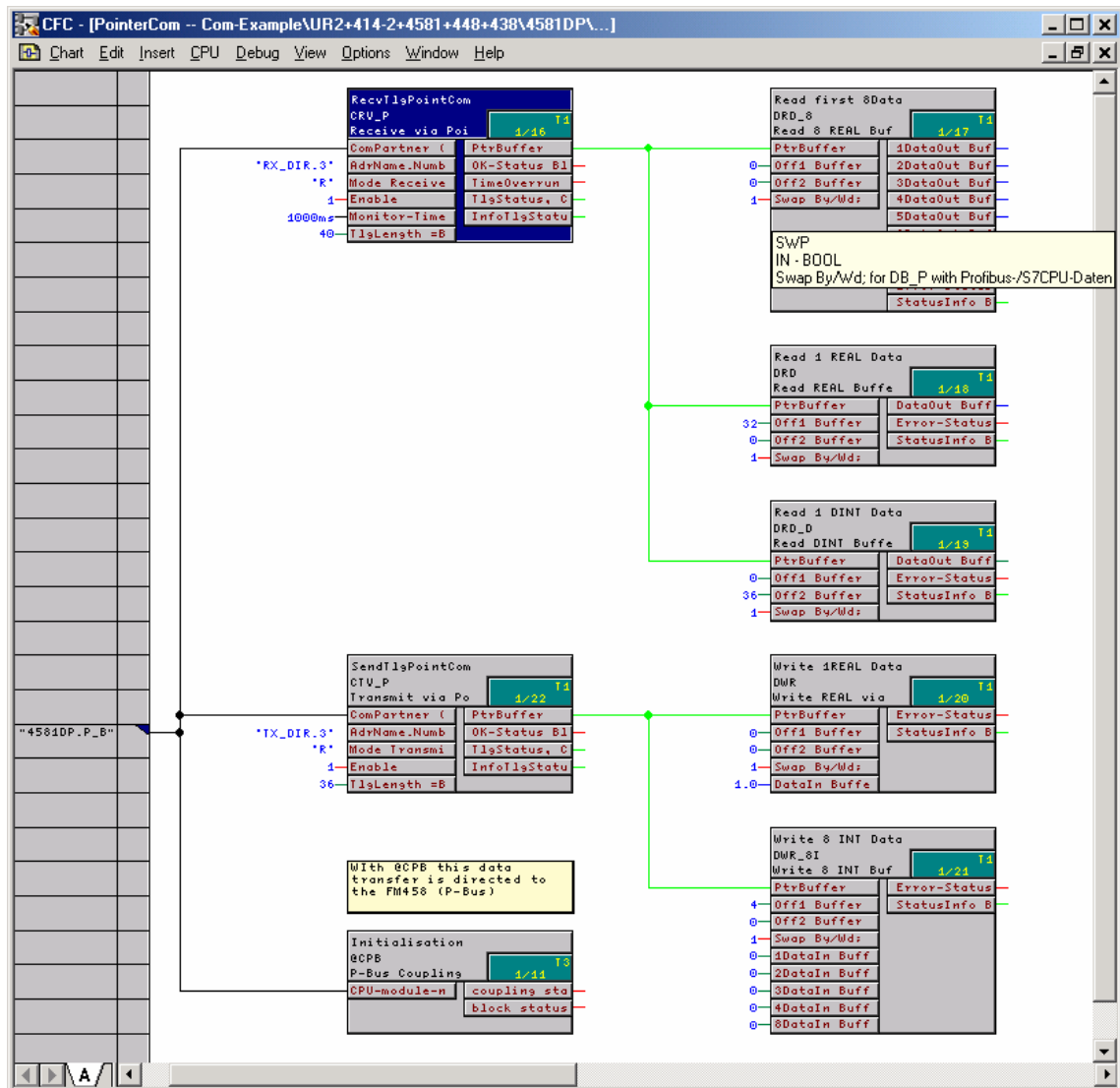


Bild 3-86 CFC-Screenshot: Datentransfer mit Telegrammbausteinen und Schreib-/Lesebausteinen; hier für die Schnittstelle P-Bus der FM 458-1 DP (@CPB); wegen Datenhaltung auf SIMATIC-CPU müssen Bytes/Worte gedreht werden: SWP(Swap)=1

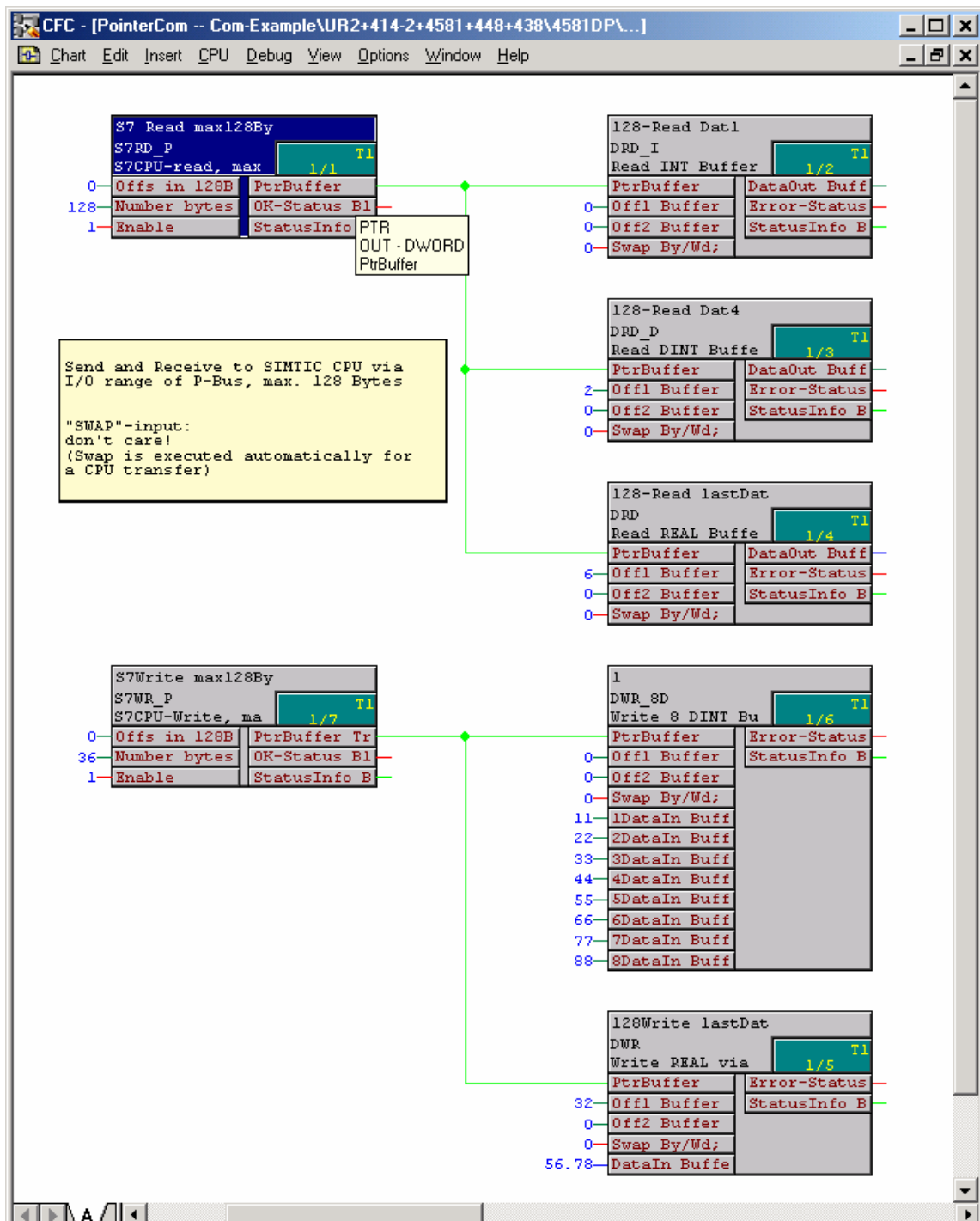


Bild 3-87 CFC-Screenshot: Datentransfer SIMATIC-CPU ↔ FM 458-1 DP über P-Bus E/A-Bereich

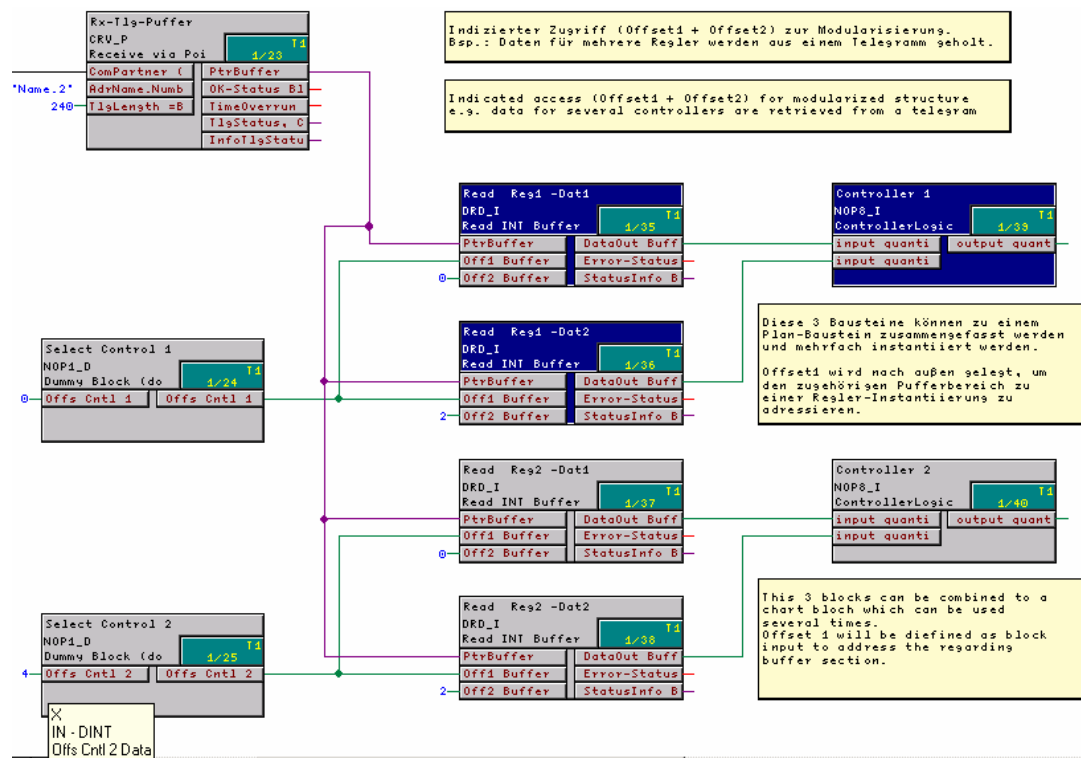


Bild 3-88 CFC-Screenshot: Indizierte Adressierung der Telegramm Daten mit 2 Offsets

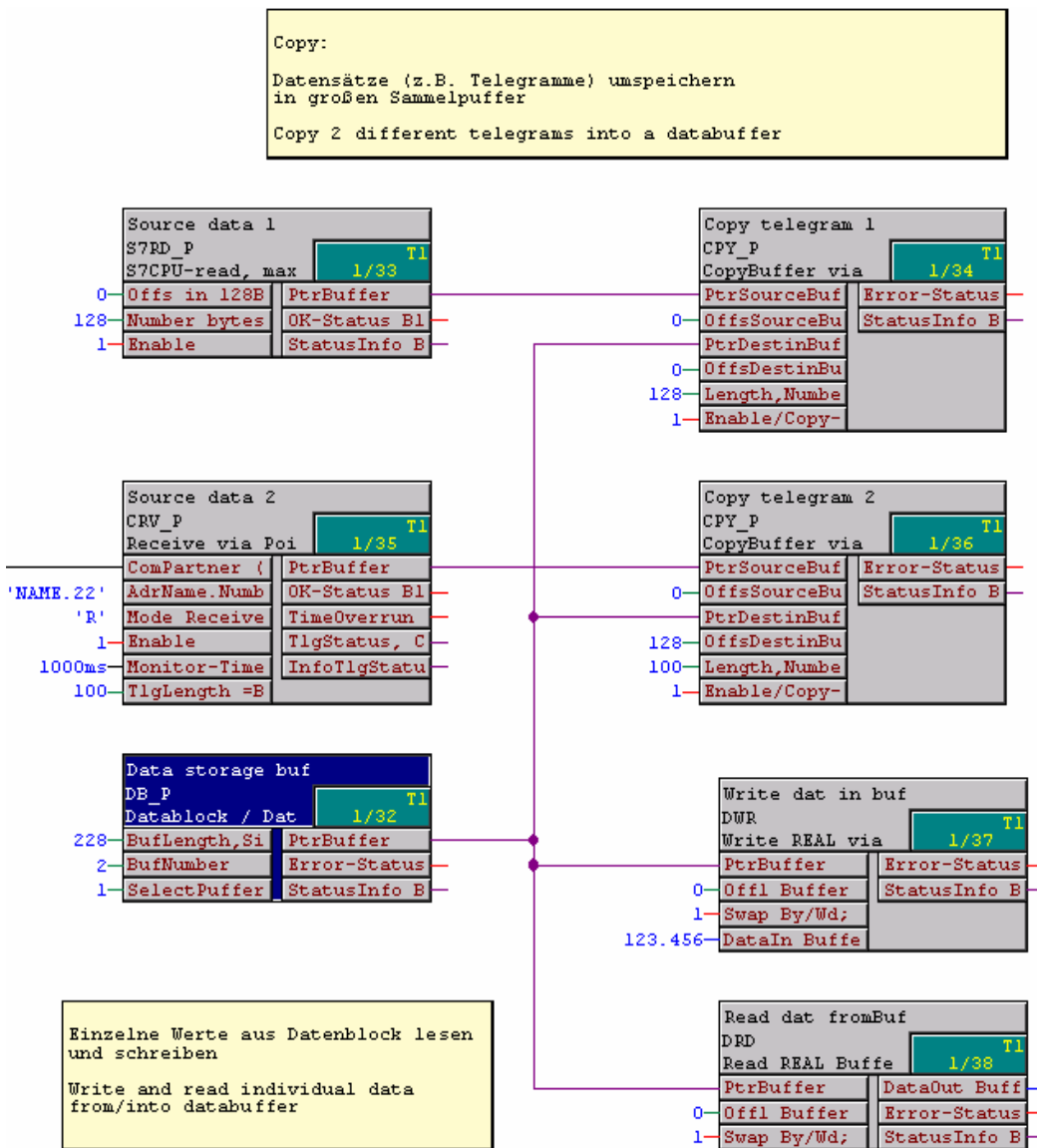


Bild 3-89 CFC-Screenshot: Umspeichern von 2 empfangenen Telegrammen in einen Datenblock(-Baustein) und Einzelzugriffe auf den Datenspeicher



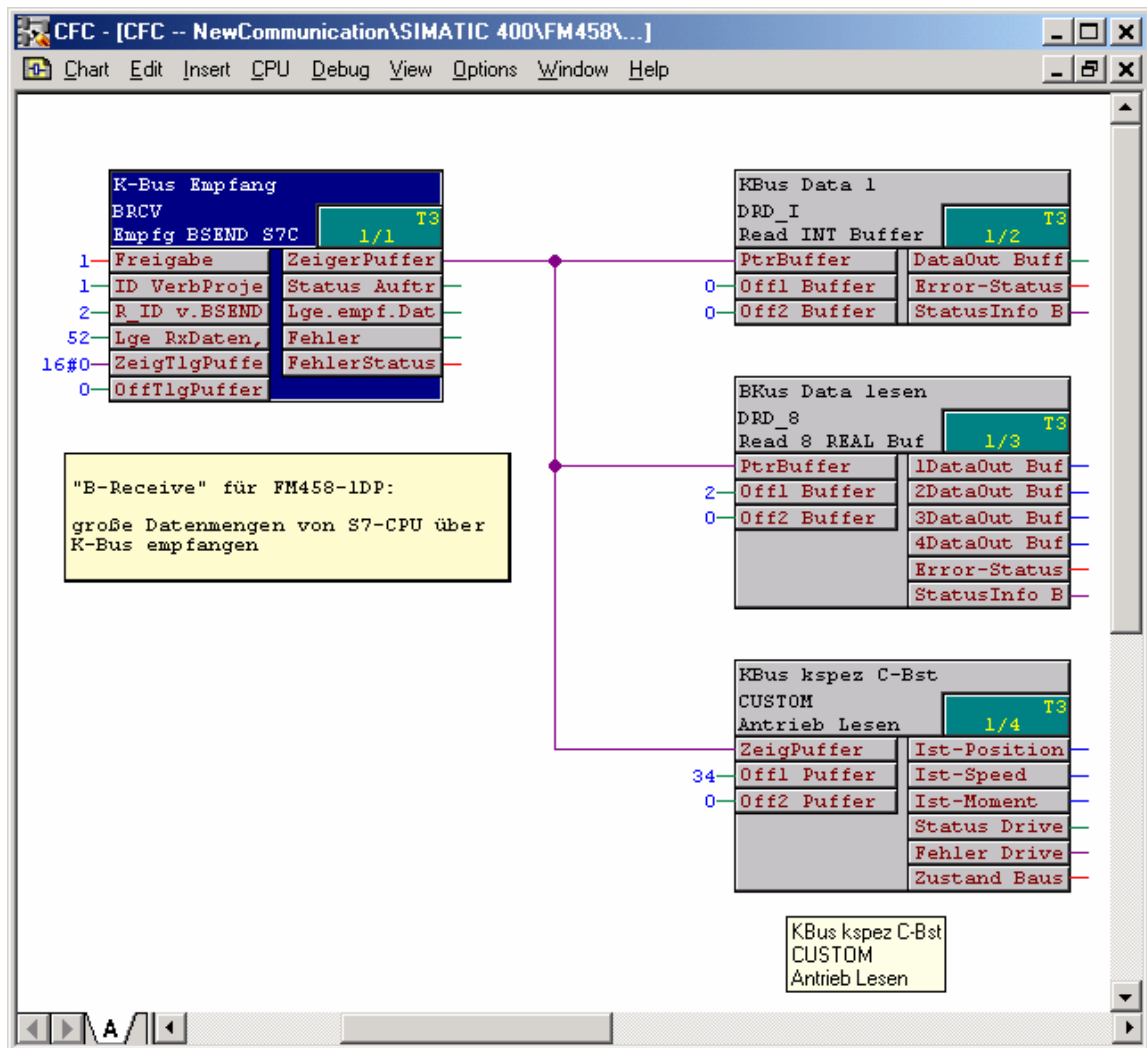


Bild 3-90 CFC-Screenshot: Große Datenmengen von SIMATIC-CPU empfangenen über K-Bus mit BRCV

### 3.13 Kommunikations-Dienst Service

<b>Kurzbeschreibung</b>	<ul style="list-style-type: none"><li>• Stellt einen Pool von Auskunftsfunktionen bereit, damit Anwender zu CPU-lokalen Systeminformationen Zugang haben</li><li>• Hilfsmittel für die Einsatzgebiete Inbetriebnahme und Debuggen</li></ul>
<b>Inbetriebnahme</b>	Hier werden projektierte Daten (Soll-/Ist-Werte) angezeigt und/oder verändert sowie die Projektierung optimiert (Verschaltungen, Reglerzeiten verändern).
<b>Debuggen</b>	<p>Hier lassen sich Ursachen für Anlagenstörungen (Absturz, Hochlaufprobleme) und Störungen, die ihre Ursache innerhalb der eigenen CPU-Baugruppe haben, ermitteln.</p> <p>Alle Aktivitäten des Kommunikations-Dienstes Service werden über Aufträge gesteuert, die über eine Kopplung (entsprechend den CTS- und US-Anschlussangaben) eintreffen.</p> <p>Bediengeräte für den Kommunikations-Dienst Service:</p> <ul style="list-style-type: none"><li>• Windows-PC mit CFC (z. B. im Testmodus)</li><li>• Windows-PC mit SIMATIC Manager</li></ul>
<b>Lokaler Service</b>	Über die lokale RS232-Schnittstelle einer CPU kann mit CFC, SIMATIC Manager oder Einfach-Servicetool auf diese eine CPU zugegriffen werden. Dazu ist keinerlei zusätzliche Projektierung notwendig.
<b>HINWEIS</b>	<p>Mit dem CFC und dem SIMATIC Manager können Sie den Baugruppenzustand der CPU auslesen.</p> <p><b>Weitere Informationen</b></p> <p>zum Baugruppenzustand der CPU siehe Benutzerdokumentation „SIMATIC TDC, Basissoftware D7-SYS, Kap. „Diagnose“.</p>
<b>Zentraler Service</b>	<p>Über eine im Baugruppenträger projektierte MPI- bzw. Industrial Ethernet -Kopplung kann auf jede CPU dieses einen Baugruppenträgers zugegriffen werden.</p> <p>Dazu ist zu projektieren:</p> <ul style="list-style-type: none"><li>• Einmal pro Baugruppenträger:<ul style="list-style-type: none"><li>– MPI-Kopplung: Ein CP50M1- / CP50M0-Modul und ein Zentralbaustein MPI-Kopplung „@MPI“</li><li>– Industrial Ethernet-Kopplung: Ein CP51M1-Modul und ein Zentralbaustein „@TCPIP“</li></ul></li><li>• Mindestens einmal pro CPU:<ul style="list-style-type: none"><li>– Service-Funktionbaustein „SER“</li></ul></li></ul> <p><b>Weitere Informationen</b></p> <p>zu Details der MPI-Kopplung siehe Kapitel „Kopplung MPI“ bzw. zu</p>

Details der Industrial Ethernet-Kopplung siehe Kapitel „Kopplung Industrial Ethernet (CP51M1) „.

### 3.13.1 Funktionsbaustein SER

#### Angaben an den Anschlüssen

Der Funktionsbaustein „SER“ hat einen Anschluss für eine Kopplung. Er kann mehrfach pro CPU projektiert werden.

Der Anschluss **CTS** bezeichnet die Kopplungsbaugruppe und die Schnittstelle, über die ein Bediengerät angeschlossen ist.

Am Anschluss **US** wird ein Kanalname und die Adressstufe 1 angegeben.

- **Kanalname**
  - maximal 6 Zeichen
  - ASCII-Zeichen außer „Punkt“ und @
  - Kanalname auf einer Datenschnittstelle muss eindeutig sein
- Eingabe von „.“ nach Kanalname
- **Adressstufe 1**
  - Steckplatznummer der CPU. Über diese Nummer adressiert das Bedienprogramm die CPU
  - Die Angabe muss zweistellig sein: z.B. „01“, „02“, ..., „24“

#### Beispiel: Projektierung mit CFC

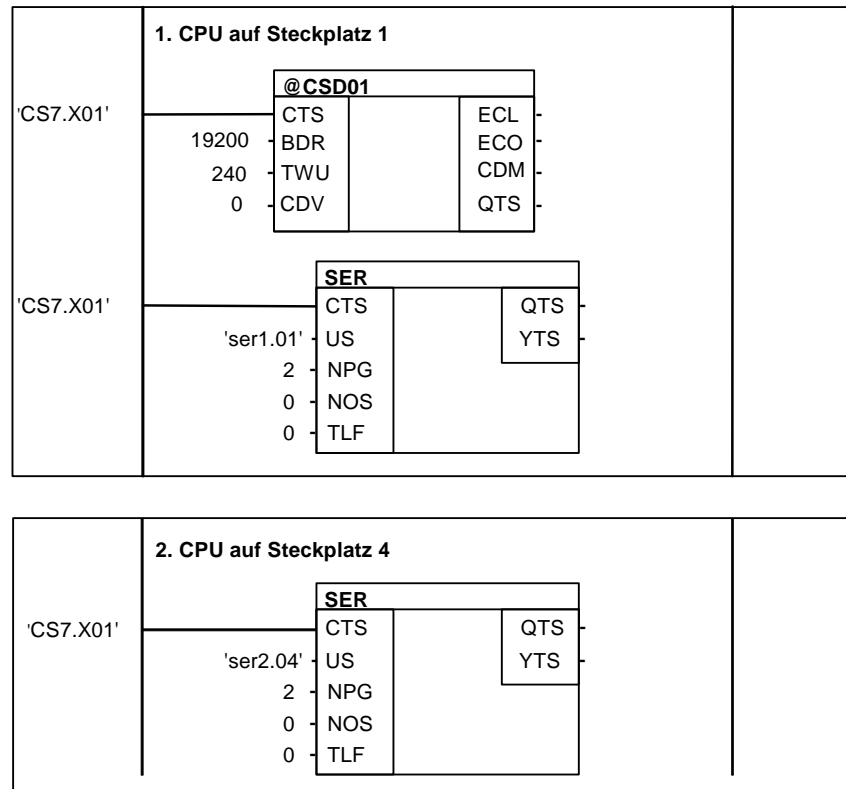


Bild 3-91 Beispiel: Projektierung mit CFC

### 3.13.2 Systembelastung, Antwortzeiten

**Allgemeines** Die eigentliche Verarbeitung des Service geschieht in einer Abtastzeit von etwa 32 ms (die an den SER-Bausteinen angegebenen Abtastzeiten sind also für die Verarbeitung nicht maßgeblich). In der verwendeten Abtastzeit haben die Servicebausteine eine gewisse Rechenzeit zur Verfügung und zwar maximal einen Grundtakt (T0).

---

**HINWEIS** Das Verhältnis von Grundtakt T0 zur verwendeten Abtastzeit bestimmt die zur Verfügung stehende CPU-Leistung und damit die Systembelastung.

---

**Beispiel1** **Grundtakt T0=1ms; gewählte Abtastzeit=32ms**

- Es wird alle 32 ms jeweils 1 ms für Dienst Service reserviert
- Systembelastung= $1 \text{ ms} / 32 \text{ ms} = 0.03125 = 3.125 \%$

**Beispiel2** **Grundtakt T0=2ms; gewählte Abtastzeit=16ms**

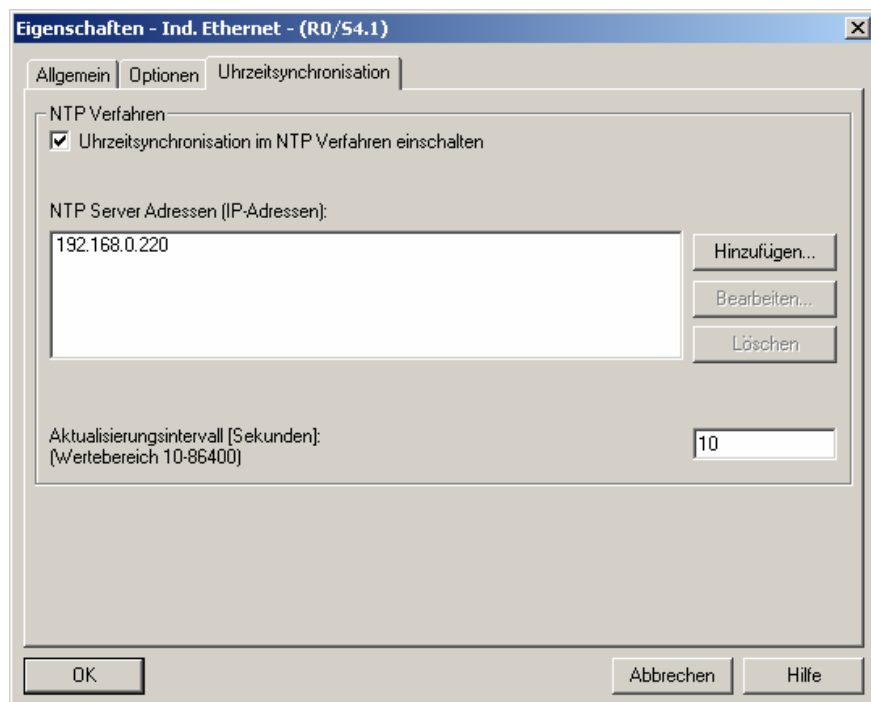
- Es werden alle 16 ms jeweils 2 ms für Dienst Service reserviert
- Systembelastung= $2 \text{ ms} / 16 \text{ ms} = 0.125 = 12.5 \%$

**Rechenzeit** Die zur Verfügung stehende Rechenzeit wird von allen Servicebausteinen gleichberechtigt genutzt. D.h solange die Zeit reicht, werden möglichst alle SER-Bausteine einmal durchlaufen. Je Takt bearbeitet ein SER-Baustein maximal einen Auftrag. Wird die reservierte Rechenzeit nicht vollständig genutzt, weil z.B. kein Auftrag zur Bearbeitung ansteht, wird sie dem System zur Verfügung gestellt.

**Ressourcenverteilung** Bei Mehrfachprojektierung und gleichzeitigem Zugriff auf nur einmal vorhandene Systemressourcen (z.B. EEPROM), bekommt derjenige die Ressource zugeteilt, der als erster die Anforderung stellt. Alle anderen werden abgewiesen und bringen nach spätestens 1 s eine Fehlermeldung („Ressource belegt“).

### 3.14 Kommunikations-Dienst Uhrzeitsynchronisation

<b>Allgemeines</b>	Der Kommunikations-Dienst Uhrzeitsynchronisation ermöglicht eine einheitliche Systemzeit über mehrere SIMATIC TDC-Baugruppenträger.
<b>Uhrzeit</b>	<p>Als Uhrzeit-Quelle können dienen:</p> <ul style="list-style-type: none"><li>• Die am weitesten links steckende CPU eines Baugruppenträgers</li><li>• Die Kopplungs-Baugruppe CP52A0 / CP53M0</li></ul> <p>Die Industrial Ethernet-Baugruppe CP51M1 Die Uhrzeit wird verteilt:</p> <ul style="list-style-type: none"><li>• Innerhalb eines SIMATIC TDC-Baugruppenträgers über eine Koppelspeicherbaugruppe</li><li>• Zu anderen SIMATIC TDC -Baugruppenträgern über die Baugruppenträger-Kopplung</li></ul>
<b>Funktionsbaustein</b>	<p>Zur Verteilung der Systemzeit ist pro Baugruppenträger genau einmal der Funktionsbaustein RTCM zu projektieren.</p> <p><b>Weitere Informationen</b> zur Projektierung von Funktionsbausteinen siehe Referenzhandbuch „Regelsystem SIMATIC TDC, Funktionsbaustein-Bibliothek“.</p> <p>Zum Auslesen der Systemzeit dienen die Funktionsbausteine:</p> <ul style="list-style-type: none"><li>• RTCABS: absolute Zeit im Datum/Uhrzeit-Format</li><li>• RTCREL: relative Zeit in Sekunden seit dem 01.01.88</li></ul> <p>Diese Bausteine können beliebig projektiert werden.</p>
<b>CP51M1</b>	<p>Die CP51M1 IE-Baugruppe ist in der Lage, sich die Uhrzeit von bis zu vier NTP-Uhrzeitservern (z.B. SICLOCK TM) aktiv zu holen.</p> <p><b>Weitere Informationen</b> zum NTP-Uhrzeitsynchronisationsverfahren siehe: <a href="http://www.ntp.org">www.ntp.org</a></p> <p>In HW-Konfig sind im Registerblatt „Uhrzeitsynchronisation“ des „Ind. Ethernet Anschluss der CP51M1“ die folgenden Einstellungen erforderlich:</p> <ul style="list-style-type: none"><li>• NTP-Verfahren einschalten</li><li>• IP-Adresse des bzw. der NTP-Server hinzufügen</li><li>• Aktualisierungsintervall (Sekunden) einstellen</li></ul>



Die CP51M1 Baugruppe holt sich im Zeitabstand des Aktualisierungsintervalls (empfohlen 10 Sekunden) die aktuelle Uhrzeit von allen projektierten NTP-Servern. Da immer die Uhrzeit des aktuell „besten“ NTP-Servers verwendet wird, darf sich gelieferte Uhrzeit aller projektierten NTP-Server nicht merklich ( $\ll 1$  ms) unterscheiden. Bei Abweichung der Uhrzeit der NTP-Server untereinander sind sonst Sprünge bei der Uhrzeit möglich. Der „beste“ NTP-Server wird ständig neu u.a. anhand der Laufzeit der Uhrzeitanfrage ermittelt.

Hinweis: Sind mehr als ein NTP-Server projektiert, werden nur die NTP-Server für die Uhrzeitsynchronisation verwendet, die sich innerhalb der letzten 24 Stunden mit einer Referenzuhr (z.B. DCF77 oder GPS) abgeglichen haben. Ist nur ein NTP-Server projektiert, so wird dieser auch dann verwendet, wenn die Synchronisation mit einer Referenzuhr nicht erfolgt ist.

Im Diagnosepuffer der CP51M1 erfolgen Einträge bei wichtigen Ereignissen, wie z.B.:

- Synchronisation mit NTP-Server (kommend / gehend)
- NTP-Server wird nicht akzeptiert
- Verlust der Synchronisation des NTP-Servers mit Referenzuhr

Hinweis: Ist ein SNTPR-FB für diese CP51M1 projektiert und zusätzlich NTP projektiert, so wird die vom SNTPR ermittelte Uhrzeit im TDC-Rack verteilt. Die über das NTP-Verfahren ermittelte Uhrzeit wird lediglich zum Stellen der Baugruppenuhr der CP51M1 Baugruppe verwendet. Es erfolgt ein entsprechender Diagnosepuffereintrag.

### 3.15 Kommunikation mit SIMATIC Operator Panels

**Einleitung** Anhand dieser Musterprojektierung soll das grundsätzliche Vorgehen eines Projektors bei der Realisierung einer Kopplung von SIMATIC TDC mit einem SIMATIC OP7 beschrieben werden.

---

**HINWEIS** Mit der Projektierung von Kopplungen zu den SIMATIC Operator Panels OP27, OP37 und dem SIMATIC Touch Panel TP37 verfahren Sie bitte auf ähnliche Art und Weise.

---

Die hier beschriebene Musterkonfiguration beinhaltet sämtliche zur Verfügung stehende SIMATIC TDC-Funktionsbausteine und zeigt deren prinzipielle Anwendung. Der Funktionsumfang der Musterprojektierung wurde bewusst sehr „mager“ gehalten, um einen schnellen Einstieg in die Thematik zu schaffen. Erweiterungen der Funktionalität und/oder der HW-Komponenten sind ohne weiteres möglich. Es sind hierbei jedoch die Angaben in der jeweiligen Funktionsbaustein-Dokumentation zu beachten.

Die verwendeten Bezeichnungen für Datenbausteine, Merker, Variablen etc. sind willkürlich gewählt und nur für diese Musterkonfiguration verbindlich.

---

**HINWEIS**

- Die Abspeicherung von Wertänderungen durch SIMATIC OPs erfolgt auf der SIMATIC TDC-CPU im SAVE-Bereich.
- Bei Ausfall der Batteriepufferung wird der projektierte Wert des Eingangs als Vorbesetzungswert verwendet.

---

**Voraussetzung** Der Aufbau dieser Projektierungsanleitung gibt den zeitlichen Ablauf der Arbeitsschritte wieder, mit der die gesamte Projektierung erstellt werden kann. Dies ist allerdings nur als Empfehlung zu verstehen und muss nicht zwingend eingehalten werden.

Der Umgang mit dem SIMATIC Manager (inkl. HWKonfig und CFC), die Projektierung von SIMATIC TDC sowie die OP7-Projektierung mit ProTool/Lite werden als bekannt vorausgesetzt.

**Literatur zu diesen Themen:**

- Benutzerdokumentation SIMATIC TDC
- SIMATIC-Gerätehandbuch OP7/17
- SIMATIC HMI, Benutzerhandbuch Projektierungssoftware ProTool/Lite

### 3.15.1 Musterkonfiguration

**Funktionsumfang** Die Musterkonfiguration unterstützt folgende Funktionen des OP7:

- Lesen und Schreiben von Variablen
- Ausgabe von Betriebsmeldungen
- Ausgabe von Störmeldungen inkl. Quittierung
- Abfrage der Funktionstastatur
- Aktualisierung von Datum und Uhrzeit

**Hardware** Für die Musterkonfiguration wurden folgende Geräte bzw. Komponenten ausgewählt und wie folgt platziert:

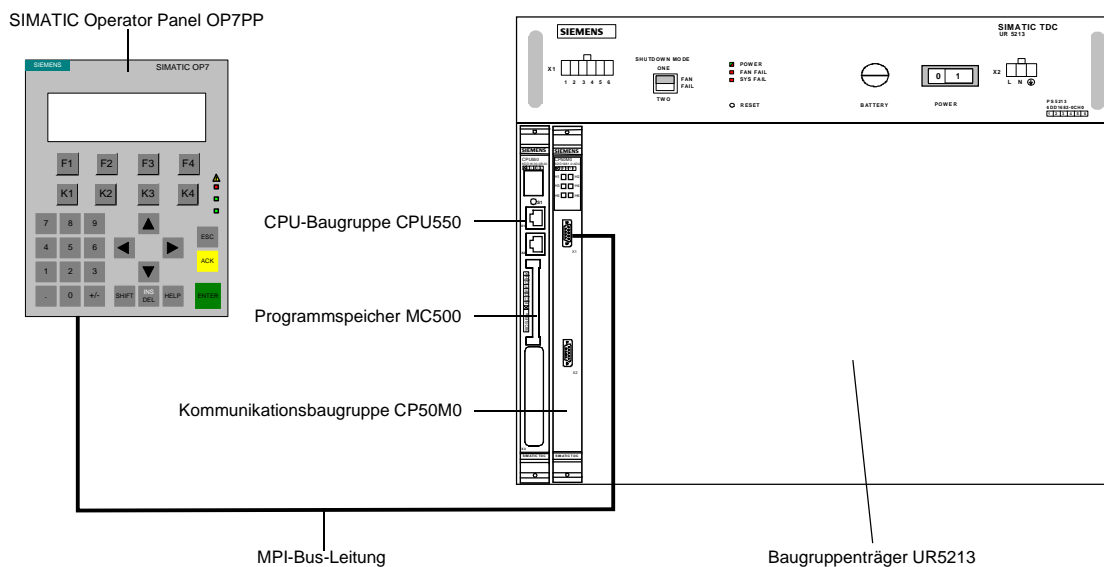


Abbildung 3-92 Aufbau der Musterkonfiguration

### 3.15.2 Projektieren von SIMATIC TDC

**Allgemeines** Im SIMATIC Manager werden alle Projektierungen, welche die SIMATIC TDC-Seite betreffen, vorgenommen. Die Arbeiten teilen sich in die Bereiche „Auswahl der Komponenten im HWKonfig“ und „Projektieren mit CFC“ auf.

#### 3.15.2.1 Auswahl der Komponenten im HWKonfig

Im HWKonfig wird die Musterkonfiguration projektiert. Es können dabei die Standardvorgaben des Programms übernommen werden. Einzige Änderungen sind:

- Abtastzeit **T4** der CPU550 = **64ms**



- Höchste MPI-Adresse der MPI-Schnittstelle auf CP50M1 / CP50M0 = **126** (in ProTool/Lite ist 126 standardmäßig vorgegeben)
- Erste Schnittstelle als MPI konfigurieren

### 3.15.2.2 Projektieren mit CFC

Nach Ausführung von „Speichern und übersetzen“ im „HWKonfig“ wurde im SIMATIC Manager unterhalb der „SIMATIC TDC-Station“ das Symbol „D01P01“ eingefügt.

#### Neuen Plan einfügen

Im zugehörigen Planbehälter wird dem bereits vorhandenen Plan „@SIMD“ nun ein neuer Plan namens „OP7“ hinzugefügt, in welchem im weiteren Verlauf alle Projektierungsarbeiten stattfinden werden.

#### Vereinbarungen

- Sämtliche zu projektierende Funktionsbausteine werden in der Ablauebene T4 projektiert.
- Soweit nicht explizit aufgeführt, werden die Standardbelegungen der Funktionsbaustein-Anschlüsse beibehalten.
- Es werden in den folgenden Projektierungstabellen nur die relevanten Anschlüsse aufgeführt.

#### 3.15.2.2.1 Initialisierung des OP7

#### Kurzbeschreibung

Über die Eingänge **CTS** werden die Funktionsbausteine **@MPI** und **S7OS** mit der projektierten Kopplungsbaugruppe (erste SS auf CP50M1 / CP50M0) verknüpft. Damit wird die Verbindung zwischen SIMATIC TDC und OP7 hergestellt.

#### Projektierung

FB	Anschluss	Anschlussbelegung (Bedeutung)
@MPI	CTS	<b>D0200C.X01</b> (globaler Operand, Baugruppenname)
S7OS	CTS	<b>D0200C.X01</b> (globaler Operand, Baugruppenname)
	US	<b>testop.01</b> (Adressparameter)

Tabelle 3-46 Anschlussbelegung @MPI und S7OS

#### 3.15.2.2.2 Lesen von Funktionsbaustein-Anschlüssen

#### Kurzbeschreibung

Für diese Funktion wurde ein Zähler projektiert, der kontinuierlich vom Startwert („0“) bis zu einem Endwert („50“) hochzählt, sich automatisch zurücksetzt und danach sein Werk wieder von vorne beginnt. Der Ausgang **Y** (Zählerstand) des **CTR** wurde mit einem globalen Operand (OP-Verbindung) verknüpft, dessen Inhalt am OP7 ausgelesen wird.

**HINWEIS** Die unter SIMATIC TDC vergebene Merker-Nr. für die OP-Verbindung muss auch unter ProTool/Lite der projektierten Variablen zugewiesen werden.

**Projektierung**

FB	Anschluss	Anschlussbelegung (Bedeutung)
BF	T	<b>500ms</b> (Zeitkonstante)
CTR	LU	<b>50</b> (Zählerobergrenze)
	Y	Symbolname: <b>Z_Ausgabe</b> Merker-Nr.: <b>MW10</b> (globaler Operand, OP-Verbindung)

Tabelle 3-47 Anschlussbelegung von BF und CTR

### 3.15.2.2.3 Schreiben von Funktionsbaustein-Anschlüssen

**Kurzbeschreibung** Mit einem globalen Operand (OP-Verbindung) wird ein Wert vom OP7 eingelesen, über einen Blindbaustein (NOP1\_I) geführt und mit einem weiteren globalen Operanden (OP-Verbindung) wieder zum OP7 zurückgeschickt, wo er ausgelesen wird.

**HINWEIS** Die unter SIMATIC TDC vergebenen Merker-Nr. für die OP-Verbindungen müssen auch unter ProTool/Lite den projektierten Variablen zugewiesen werden.

**Projektierung**

FB	Anschluss	Anschlussbelegung (Bedeutung)
NOP1_I	X	Symbolname: <b>OP_SOLL</b> Merker-Nr.: <b>MW20</b> (globaler Operand, OP-Verbindung)
	Y	Symbolname: <b>OP_IST</b> Merker-Nr.: <b>MW30</b> (globaler Operand, OP-Verbindung)

Tabelle 3-48 Anschlussbelegung von NOP1\_I

### 3.15.2.2.4 Projektierung von Betriebsmeldungen

**Kurzbeschreibung** Beginnt der Zähler eine neue Zählschleife, wird eine Betriebsmeldung ausgegeben. Das Signal dazu liefert der Ausgang **QO** des Funktionsbausteins **CTR**. Dieses Signal wird zeitlich verlängert (FB PDF), vom Format „Boolean“ ins Format „Word“ konvertiert (FB B\_W) und dem Funktionsbaustein **S7EMA** als erstes Betriebsmeldewort zugeführt.

Dem S7EMA wird über einen globalen Operand (OP-Verbindung) eine virtuelle Datenbausteinnummer für den Anwenderdatenbereich „Betriebsmeldungen“ zugewiesen.

**HINWEIS** Die unter SIMATIC TDC vergebene Datenbaustein-Nr. für die OP-Verbindung muss auch unter ProTool/Lite dem projektierten Bereichszeiger für Betriebsmeldungen zugewiesen werden.

**Projektierung**

FB	Anschluss	Anschlussbelegung (Bedeutung)
PDF	I	Funktionsbaustein <b>CTR</b> , Ausgang <b>QO</b> (Signal für Betriebsmeldung)
	T	<b>5000ms</b> (Zeitkonstante)
B_W		(Konvertierung von Boolean zu Word)
S7EMA	XDB	Symbolname: <b>BM</b> Datenbaustein-Nr.: <b>DB1</b> (globaler Operand, OP-Verbindung)

Tabelle 3-49 Anschlussbelegung von @MPI und S7OS

### 3.15.2.2.5 Projektierung von Störmeldungen

**Kurzbeschreibung** Beginnt der Zähler eine neue Zählschleife, wird (gleichzeitig zur Betriebsmeldung) eine Störmeldung ausgegeben. Das Signal dazu liefert der Ausgang **QO** des Funktionsbausteins **CTR**. Dieses Signal wird vom Format „Boolean“ ins Format „Word“ konvertiert (FB B\_W) und dem Funktionsbaustein **S7AMA** als erstes Störmeldewort zugeführt.

Dem S7AMA wird über einen globalen Operand (OP-Verbindung) eine virtuelle Datenbaustein-Nr. für den Anwenderdatenbereich „Störmeldungen“ zugewiesen.

**HINWEIS** Die unter SIMATIC TDC vergebene Datenbaustein-Nr. für die OP-Verbindung muss auch unter ProTool/Lite dem projektierten Bereichszeiger für Störmeldungen zugewiesen werden.

Projektierung	FB	Anschluss	Anschlussbelegung (Bedeutung)
	B_W	I1	Funktionsbaustein <b>CTR</b> , Ausgang <b>QO</b> (Signal für Störmeldung)
	S7AMA	XDB	Symbolname: <b>SM</b> Datenbaustein-Nr.: <b>DB10</b> (globaler Operand, OP-Verbindung)

Tabelle 3-50 Anschlussbelegung von B\_W und S7AMA

### 3.15.2.2.6 Projektierung der Funktionstastatur

**Kurzbeschreibung** Die Projektierung der Funktionstastatur umfasst auf SIMADYN D-Seite nur den Funktionsbaustein **S7FKA**. Die eigentliche Zuweisung der Tastenfunktionen erfolgt unter ProTool/Lite.

Dem S7FKA wird über einen globalen Operand (OP-Verbindung) eine virtuelle Datenbaustein-Nr. für den Anwenderdatenbereich „Funktionstastaturabbild“ zugewiesen.

**HINWEIS**

Die unter SIMATIC TDC vergebene Datenbaustein-Nr. für die OP-Verbindung muss auch unter ProTool/Lite dem projektierten Bereichszeiger für die Funktionstastatur zugewiesen werden.

Projektierung	FB	Anschluss	Anschlussbelegung (Bedeutung)
	S7FKA	XDB	Symbolname: <b>FK_Tast</b> Datenbaustein-Nr.: <b>DB20</b> (globaler Operand, OP-Verbindung)

Tabelle 3-51 Anschlussbelegung von S7FKA

### 3.15.2.2.7 Projektierung des Schnittstellenbereichs

**Kurzbeschreibung** Mittels dieser Funktion wird Uhrzeit und Datum des OP7 zyklisch von SIMATIC TDC aus aktualisiert. Dem S7IA wird über einen globalen Operand (OP-Verbindung) eine virtuelle Datenbaustein-Nr. für den Anwenderdatenbereich „Schnittstellenbereich“ zugewiesen.

**HINWEIS**

Die unter SIMATIC TDC vergebene Datenbaustein-Nr. für die OP-Verbindung muss auch unter ProTool/Lite dem projektierten Bereichszeiger für den Schnittstellenbereich zugewiesen werden.

Projektierung	FB	Anschluss	Anschlussbelegung (Bedeutung)
	S7IA	XDB	Symbolname: <b>SB</b> Datenbaustein-Nr.: <b>DB30</b> (globaler Operand, OP-Verbindung)

### 3.15.2.3 Importieren der Symboltabelle

#### Allgemeines

Während der CPU-Projektierung im HWKonfig wird automatisch eine leere Symboltabelle angelegt, welche später die mit CFC projektierten Symbolnamen aufnimmt. Am Ende der CFC-Projektierung muss die Datei mit den Symbolnamen in die Symboltabelle importiert werden.

#### Symbol Editor

Vom Planbehälter aus wird durch doppelklicken auf „Symbole“ der Symbol Editor geöffnet.  
Über den Menübefehl „Tabelle - Importieren...“ wird die Symboldatei (symbol.asc) in die Symboltabelle geladen.

#### HINWEIS

Ergeben sich in CFC zwischen zwei Übersetzungsvorgängen Änderungen in der Symboldatei, so wird dies anhand einer Meldung kundgetan. Dieser Meldung kann auch der aktuelle Speicherpfad der Symboldatei entnommen werden.

Folgende Abbildung zeigt die komplette Symboltabelle der Testprojektierung nach dem Importvorgang:

	Symbol	Adresse	Datentyp	Kommentar
1	OP_IST	MW 30	INT	OP7.5.Y
2	OP_SOLL	MW 20	INT	OP7.5.X
3	Z_Ausgabe	MW 10	INT	OP7.3.Y
4	BM	DB 1	DB 1	OP7.11.XDB
5	FK_Tast	DB 20	DB 20	OP7.15.XDB
6	SB	DB 30	DB 30	OP7.4.XDB
7	SM	DB 10	DB 10	OP7.6.XDB
8				

Bild 3-93 Symboltabelle mit importierter Symboldatei

Über „Tabelle - Speichern“ wird die Symboltabelle gesichert und der Vorgang abgeschlossen.

### 3.15.3 Projektierung des OP7 mit ProTool/Lite

#### Allgemeines

Die Projektierung des OP7 wird an dieser Stelle nicht bis ins Detail beschrieben. Sofern nicht explizit darauf hingewiesen, können bei der Projektierung die Standardeinstellungen von ProTool/Lite übernommen werden.

#### HINWEIS

Für eine fehlerlose Kommunikation ist es unbedingt notwendig, die in CFC projektierten Merker- bzw. Datenbaustein-Nummern für die einzelnen Funktionen unverändert in ProTool/Lite zu übernehmen.

#### Symboltabelle

Von CFC wurde eine Symboltabelle erzeugt, in der sämtliche verwendete Merker und Datenbausteine gespeichert sind. Diese Symboltabelle muss für die ProTool/Lite-Projektierungsarbeiten importiert werden. Es können nun für die OP7-Projektierung die in CFC projektierten Symbolnamen in ProTool/Lite verwendet werden.

## Projektierung

Für das OP7 ist eine Projektierung mit Bildern (inkl. Variablen zum Lesen und Schreiben von Werten), Betriebs- und Störmeldungen sowie projektierten Funktionstasten zu erstellen.

Folgende Tabelle zeigt eine Übersicht der benötigten Projektierungskomponenten mit den dazugehörigen, auf die CFC-Projektierung abgestimmten Werten:

Projektierung	Einstellung
Steuerung	<b>SIMATIC S7-300/400</b>
MPI-Einstellungen	Kommunikationspartner-Steckplatz: <b>1</b>
Variable zum Lesen der Funktionsbaustein-Anschlüsse	<b>Symbolname: Z_Ausgabe</b> (VAR_1: Format „INT“, Typ „A“, Bereich „M“, MW10)
Variablen zum Schreiben der Funktionsbaustein-Anschlüsse	<b>Symbolname: OP_SOLL</b> (VAR_2: Format „INT“, Typ „E“, Bereich „M“, MW20) <b>Symbolname: OP_IST</b> (VAR_3: Format „INT“, Typ „A“, Bereich „M“, MW30)
Bereichszeiger Betriebsmeldungen	<b>Symbolname: BM</b> (DB1, DBW0, Länge „8“ Worte)
Bereichszeiger Störmeldungen	<b>Symbolname: SM</b> (DB10, DBW0, Länge „8“ Worte)
Bereichszeiger Quittierung-SPS	<b>DB10, DBW16</b> , Länge „8“ Worte
Bereichszeiger Quittierung-OP	<b>DB10, DBW32</b> , Länge „8“ Worte
Bereichszeiger Funktionstastatur	<b>Symbolname: FK_Tast</b> (DB20, DBW0, Länge „1“ Wort)
Bereichszeiger Schnittstellenbereich	<b>Symbolname: SB</b> (DB30, DBW0, Länge „16“ Worte)

## 3.15.4 Anwendungs-Hinweise

### 3.15.4.1 Rechenzeiten

#### Allgemeines

Die Rechenzeiten der Funktionsbausteine sind abhängig von der Anwendung.

Die folgende Tabelle zeigt die Rechenzeiten der Funktionsbausteine für ein OP7 in µs. Jedes weitere projektierte OP7 erhöht die Rechenzeit entsprechend.

	S7OS	S7EMA	S7AMA	S7FKA	S7IA
<b>Ein OP7</b>	105	3	84	25	19
<b>Jedes weitere OP7</b>	55	2	33	22	18

### 3.16 WinCC- Anbindung an SIMATIC TDC über Standard-Kanal (SIMATIC S7 Protocol Suite.CHN)

Dieses Kapitel beschreibt das Vorgehen bei der Projektierung des Zugriffs mit WinCC auf Prozessvariable (Bausteinanschlüsse) einer SIMATIC TDC CPU über den „Standard-Kanal“. Die folgende Grafik stellt die verschiedenen Projektierungs- und Kopplungsmöglichkeiten dar.

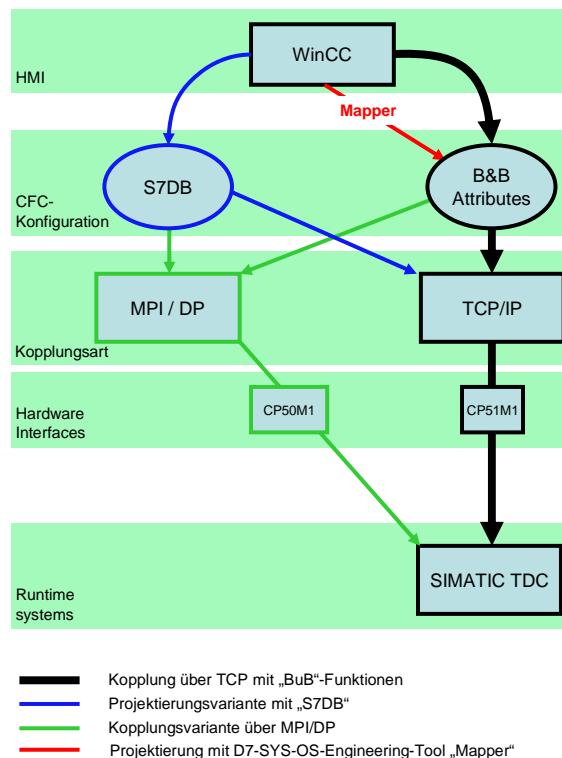
Die Kopplung über TCP/IP unter Nutzung der FB-Eigenschaft „BuB“-Funktionen (**B**edien- und **B**eobachtung), bei der die Erzeugung eines Adressbuches erforderlich ist, ist vollständig beschrieben.

Im darauf folgenden Kapitel sind die Besonderheiten der Projektierungsvariante mit dem Funktionsbaustein „S7DB\_P“ beschrieben, bei der auf die Erzeugung eines Adressbuches verzichtet werden kann.

Anschließend ist die Verwendung der Kopplungsarten MPI und PROFIBUS DP erläutert. (Bei Verwendung der PROFIBUS DP-Kopplung zu Visualisierungszwecken, ist allerdings zu berücksichtigen, dass diese, i.a. als schnelle Antriebskopplung eingesetzte Verbindung ggf. durch die HMI-Signale ausgebremst wird.)

Abschließend ist die Anwendung des D7-SYS-OS-Engineering-Tool (WinCC-Mapper) erklärt.

Übersicht Kommunikations- und Projektierungsmöglichkeiten  
WinCC – SIMATIC TDC

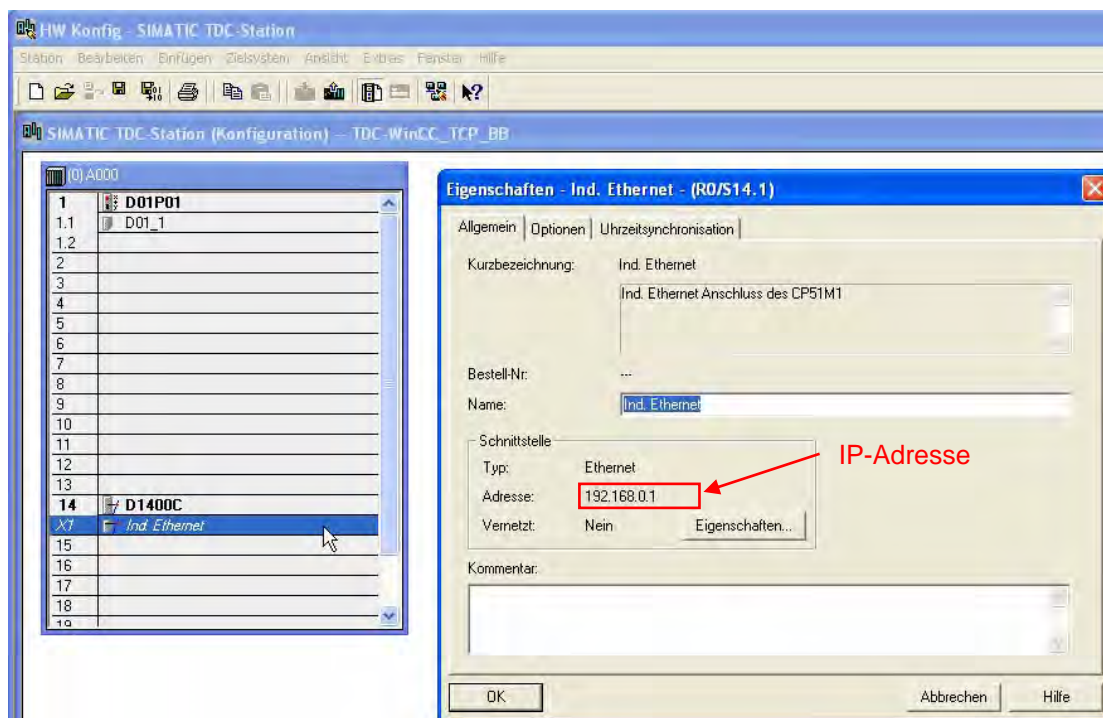


### 3.16.1 Kopplung über TCP/IP mit „BuB“- Funktionen

Dieses Kapitel beschreibt das Vorgehen bei der Projektierung des Zugriffs mit WinCC auf Prozessvariable (Bausteinanschlüsse) einer SIMATIC TDC CPU..

#### 3.16.1.1 Projektierung der kopplungsrelevanten TDC- Hardware

Für die TCP/IP-Kopplung ist im SIMATIC TDC – Rack eine CP51M1 als Kommunikationsbaugruppe zu projektieren. Über den Button „Eigenschaften“ ist ein weiteres Fenster zu öffnen, in dem ein Subnetz einzufügen und die IP-Adresse einzustellen ist.

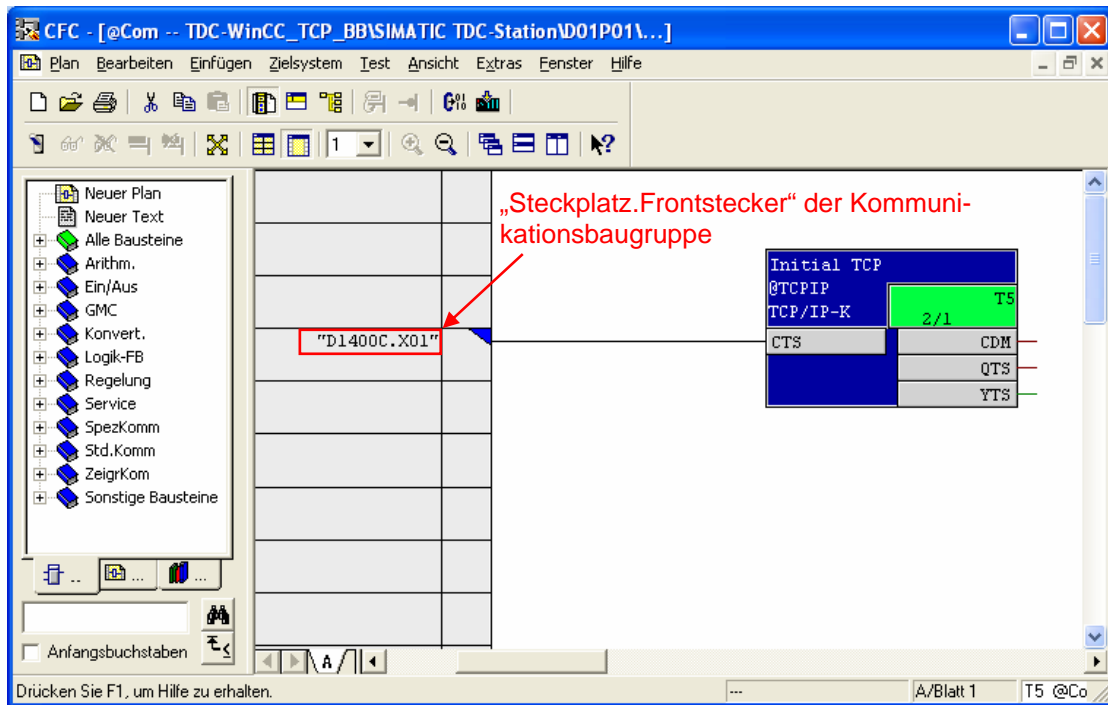




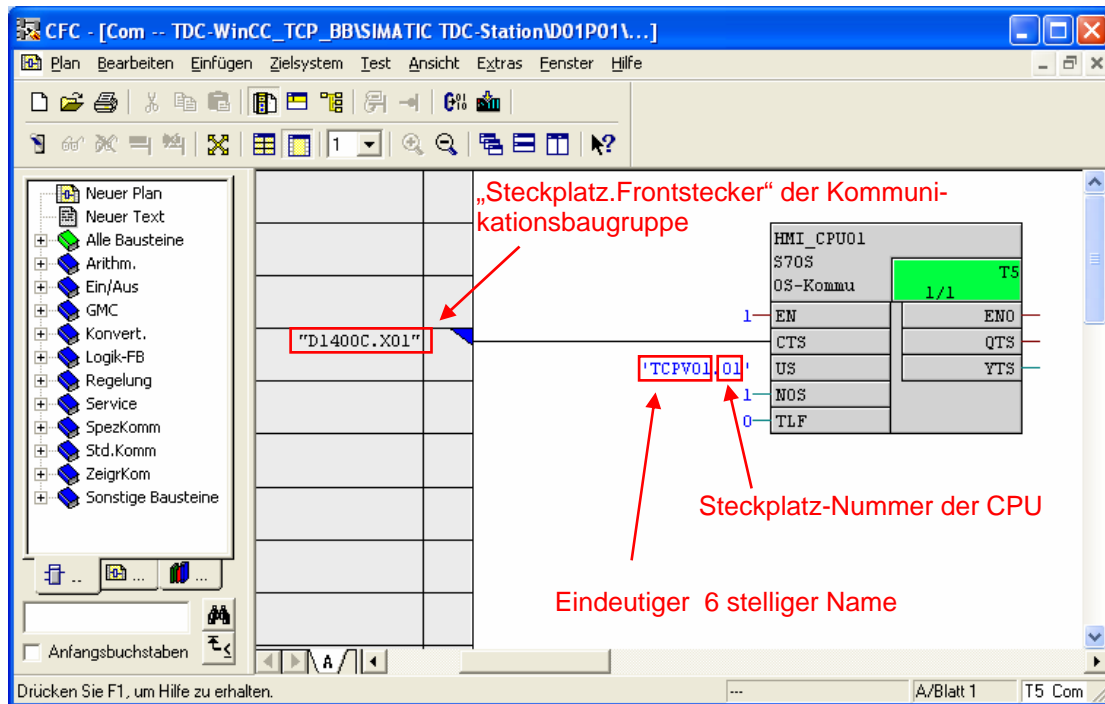
### 3.16.1.2 CFC-Projektierung

#### 3.16.1.2.1 Projektierung der kopplungsrelevanten CFC-Funktionsbausteine

Projektierung eines Funktionsbausteins zur Initialisierung der TCP/IP-Schnittstelle („X01“) der Baugruppe CP51M1 auf dem Steckplatz 14 („D1400C“). Der Funktionsbaustein ist in einer Task zwischen 32 und 256 ms zu projektieren.



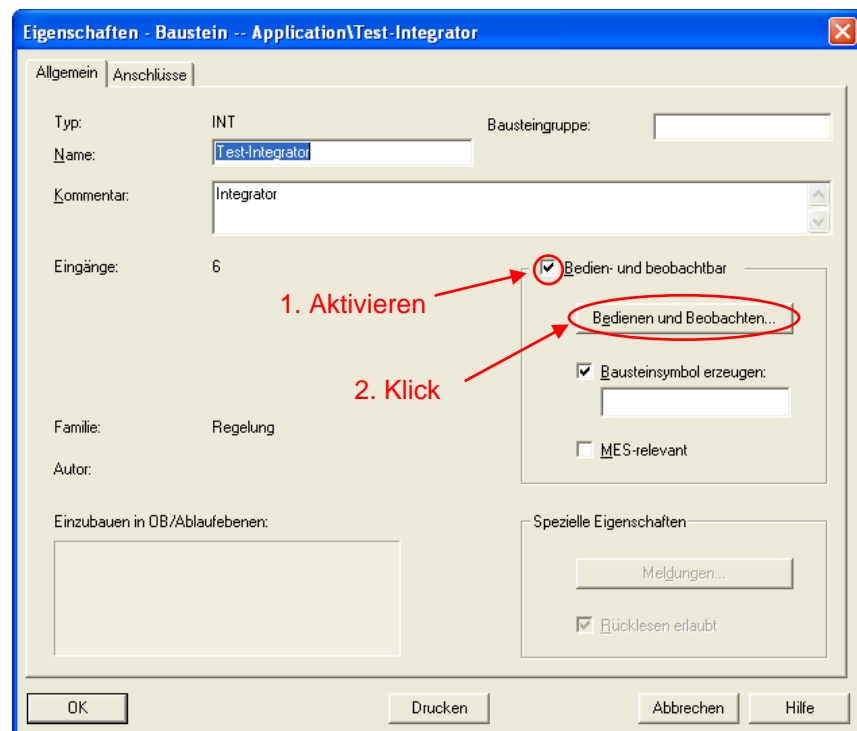
Projektierung des Funktionsbausteins „S7OS“ zur Initialisierung der OS-Kommunikation auf der „CPU 01“ auf Steckplatz 1. Statt des „S7OS“-Bausteins könnte grundsätzlich ebenso der Baustein „SER“ verwendet werden, was aber ggf. Einfluss auf die Reaktionszeiten haben kann. Details hierzu sollten der Funktionsbausteinbeschreibung entnommen werden.



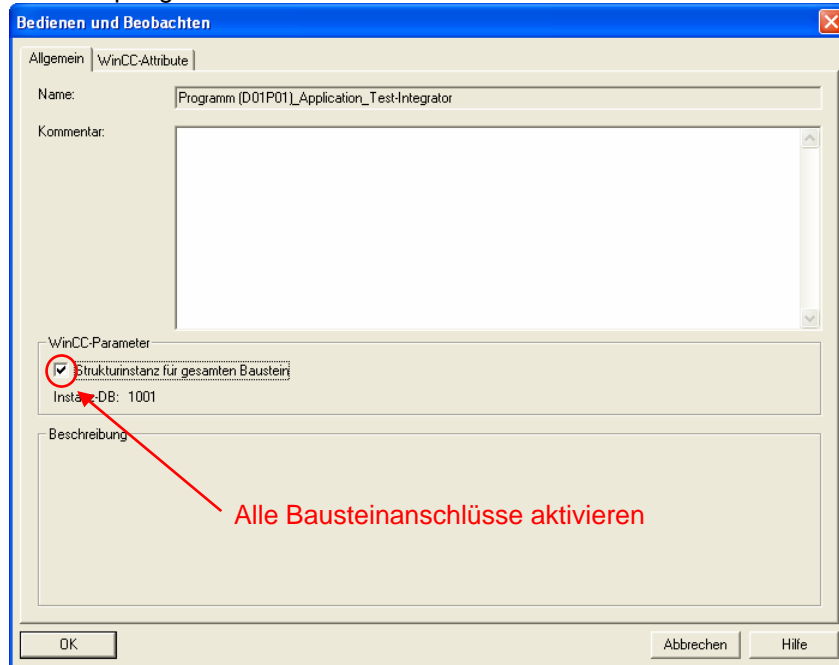
### 3.16.1.2.2 Markierung der Funktionsbausteinanschlüsse in den CFC-Plänen und Erzeugung des Adressbuches

Die Bausteinanschlüsse, welche über WinCC bedient und beobachtet (BuB) werden sollen, müssen zunächst in den CFC-Plänen als BuB-fähig markiert werden. Dazu muss in folgenden Schritten vorgegangen werden:

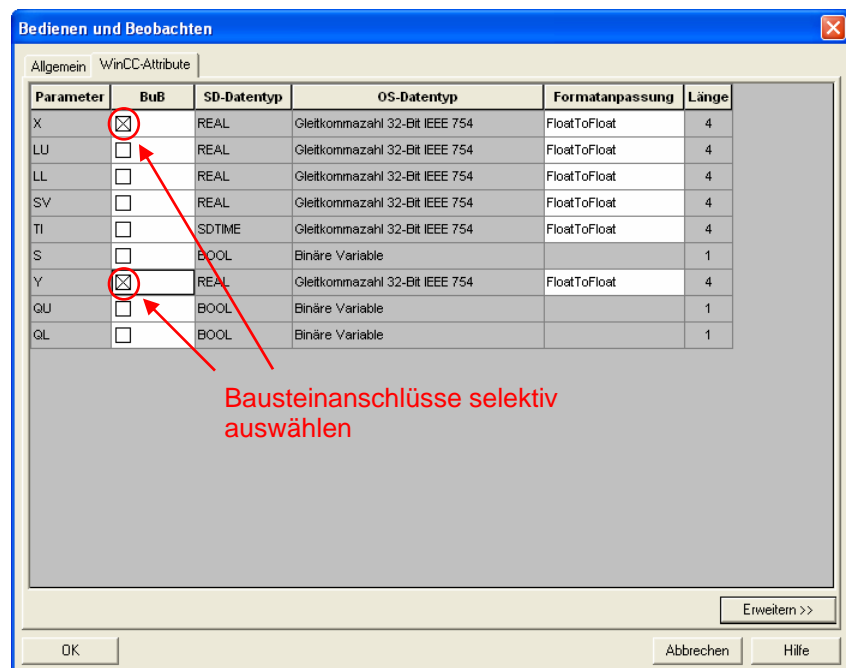
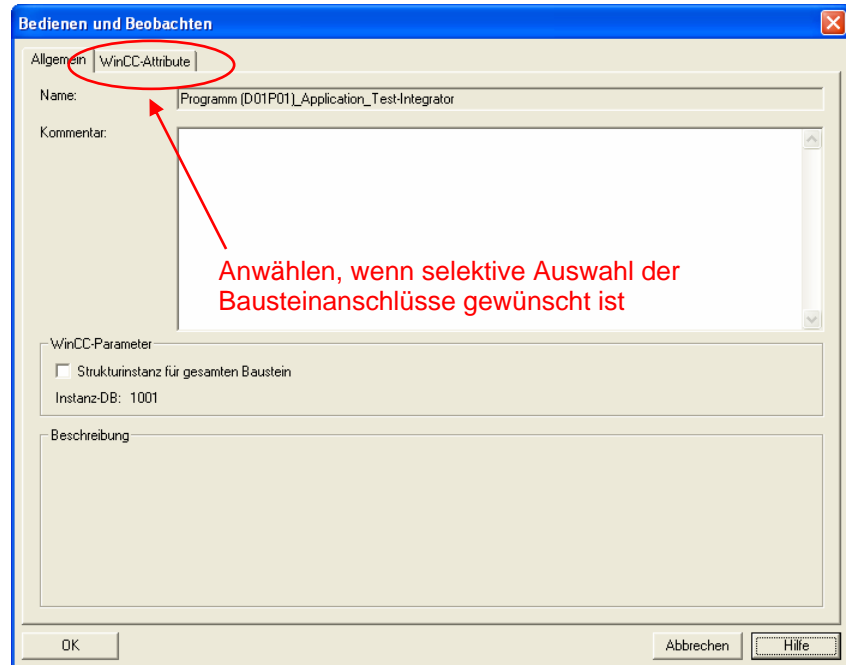
1. Den Eigenschaftsdialog des Bausteins öffnen, das Häkchen für "Bedien- und beobachtbar" setzen und anschließend den Button "Bedienen und Beobachten" drücken (siehe folgendes Bild).



2. Im dann aufgeblendeten Dialog das Häkchen bei “Strukturinstanz für gesamten Baustein” setzen, wenn alle Anschlüsse des ausgewählten Bausteins BuB-fähig sein sollen (siehe folgendes Bild). Sollen nur einzelne Anschlüsse ausgewählt werden, dann diesen Schritt überspringen und mit Schritt 3 fortfahren.

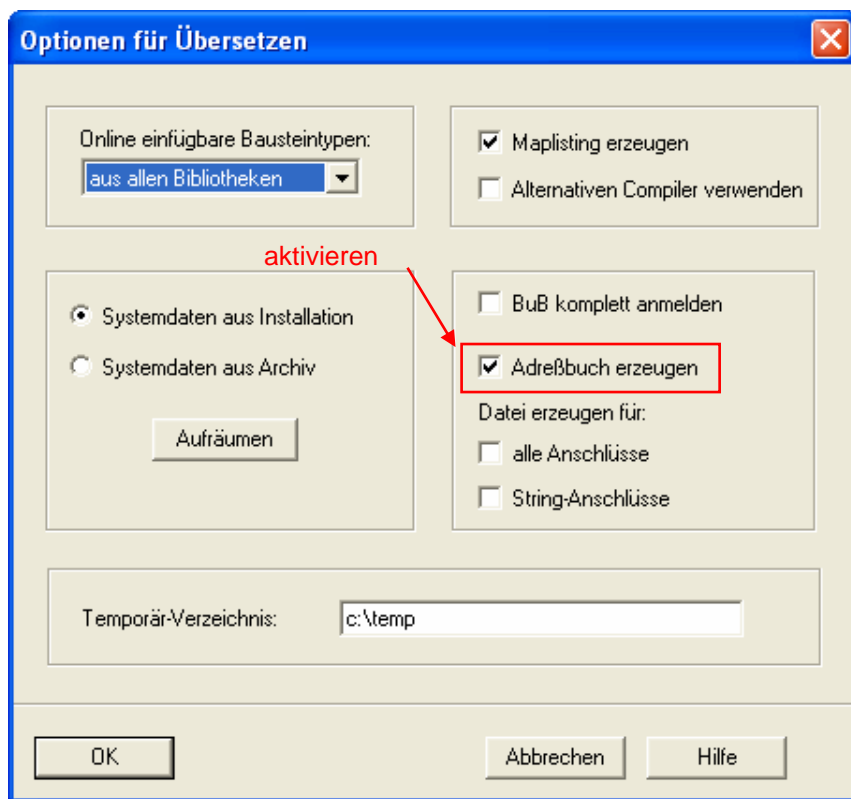
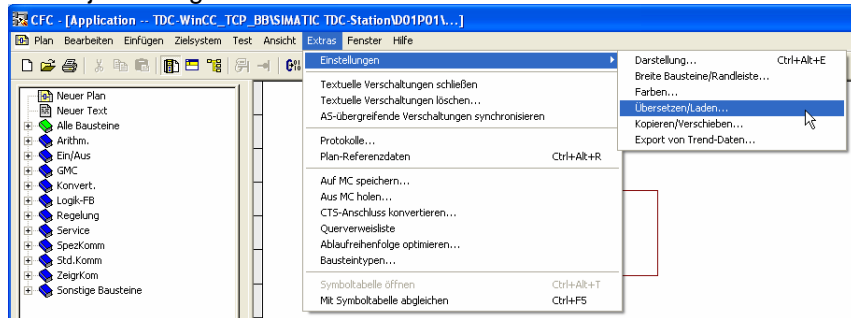


- Die BuB-fähigen Anschlüsse im Register WinCC-Attribute einzeln markieren, wenn nicht alle Anschlüsse wie in Schritt 2 ausgewählt werden sollen (siehe nächstes Bild).



- Schritte 2 und 3 für alle die Bausteine wiederholen, welche bedient und beobachtet werden sollen.

- Die Adressbuchezeugung im Optionsdialog für Übersetzen von CFC (**Extras → Einstellungen → Übersetzen/Laden**) (siehe nächstes Bild) anwählen, um die Adressinformationen für die WinCC-Projektierung zu erhalten.

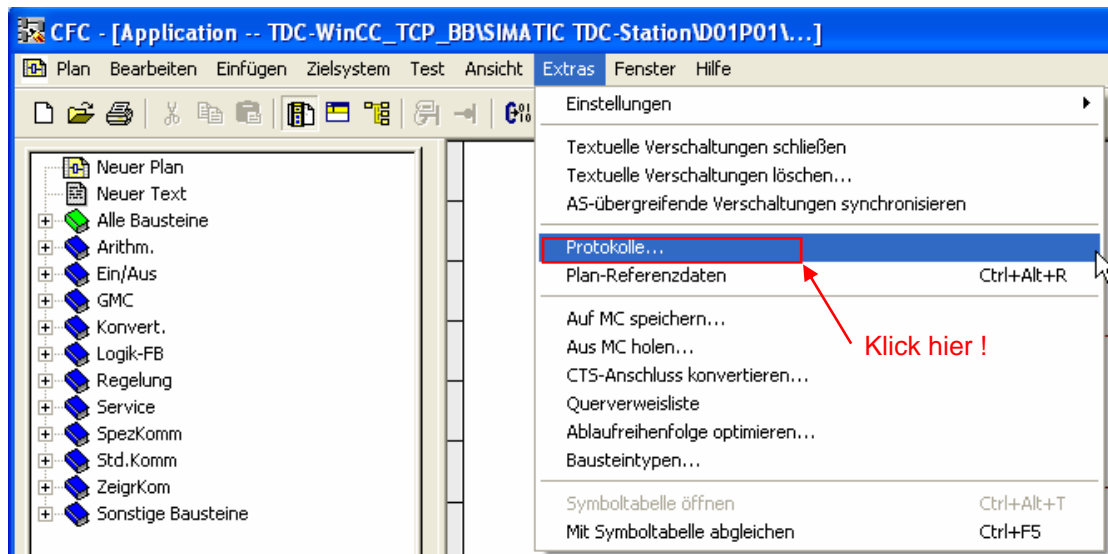


Das Adressbuch wird beim Übersetzen erzeugt, womit dann alle in den CFC-Plänen notwendigen Tätigkeiten erledigt sind. Nach dem Laden in das Zielsystem ist ein Zugriff mit WinCC grundsätzlich möglich.

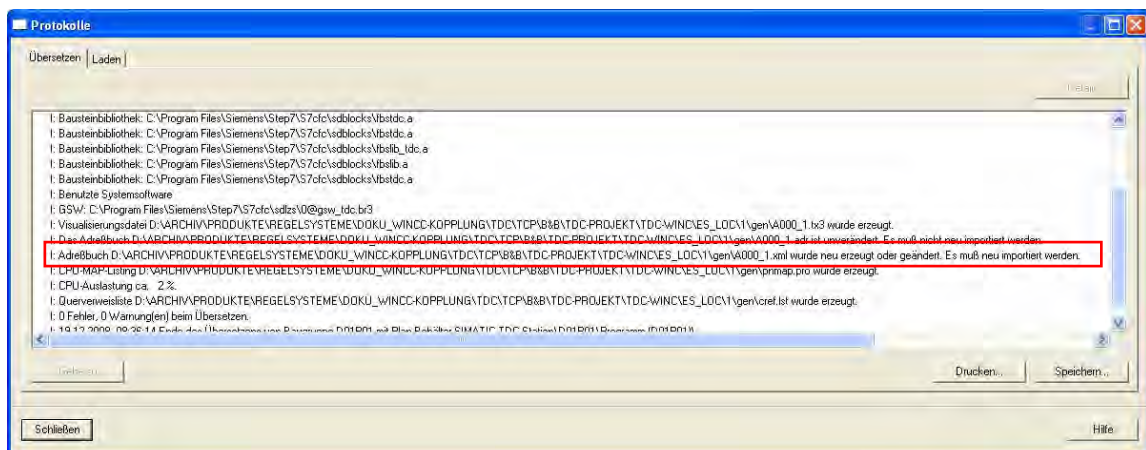
Mit D7-SYS Version 7.1 wird ein „D7-SYS-OS-Engineering“-Tool, im folgenden „Mapper“ genannt mitgeliefert. Das Tool legt durch seine Ausführung automatisch für jeden markierten Funktionsbausteinkonnektor einen Tag (WinCC-Variable) an.

**Wird der Mapper verwendet, so kann der Workflow an dieser Stelle verlassen werden und im Kapitel 3.16.4 fortgesetzt werden.**

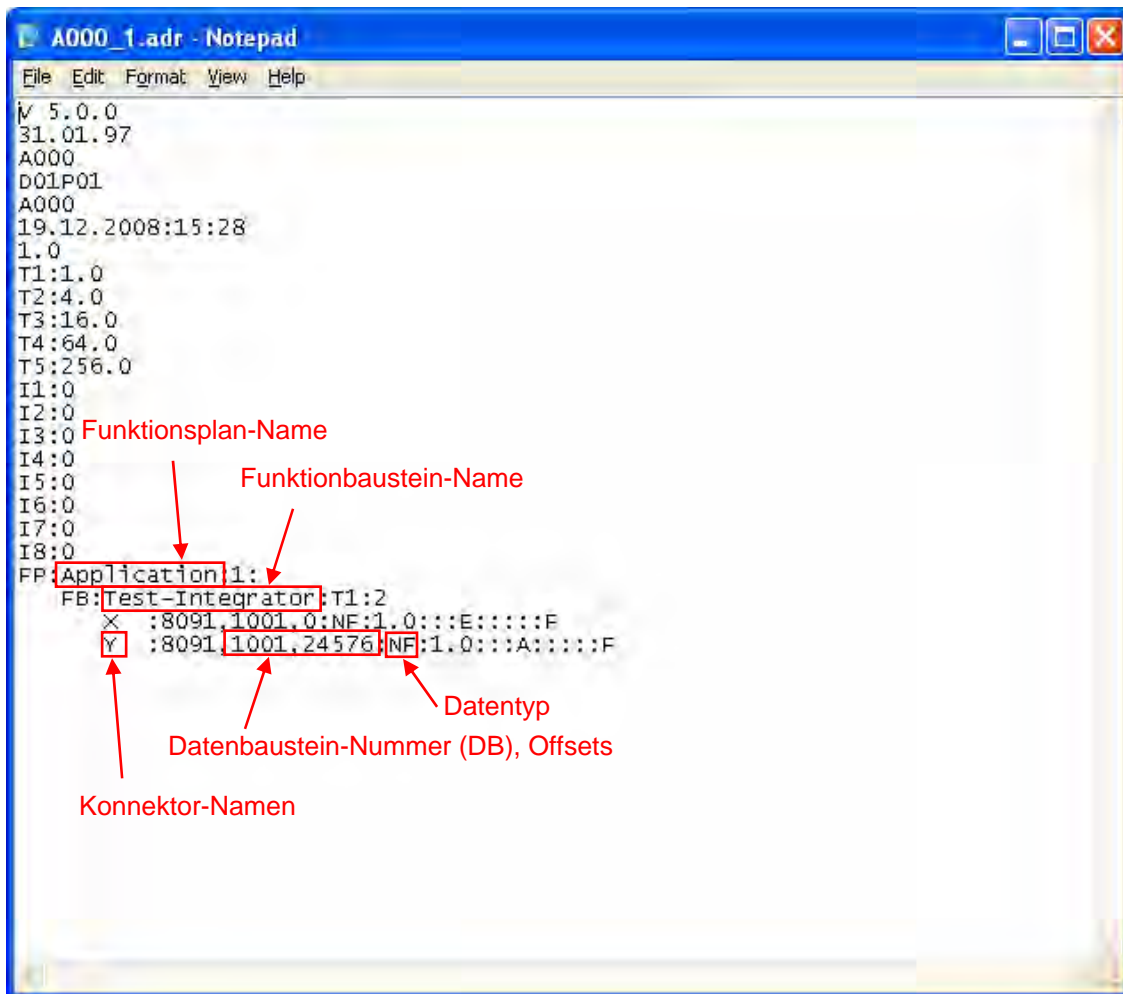
Die für die WinCC-Projektierung notwendigen DB-Nummern und Offsets der einzelnen Anschlüsse können jetzt dem Adressbuch entnommen werden. Unter dem Menüpunkt „**Extras>Protokolle**“ ist ein Protokoll zu finden, in dem hinterlegt ist, wo das Adressbuch abgelegt ist.



In der folgenden Protokollabbildung ist der Ablageort des Adressbuches markiert:



Dem Adressbuch können nun die WinCC-relevanten Informationen wie Datenbausteinnummer und Offset der gewählten Funktionsbausteinkonnektoren entnommen werden.

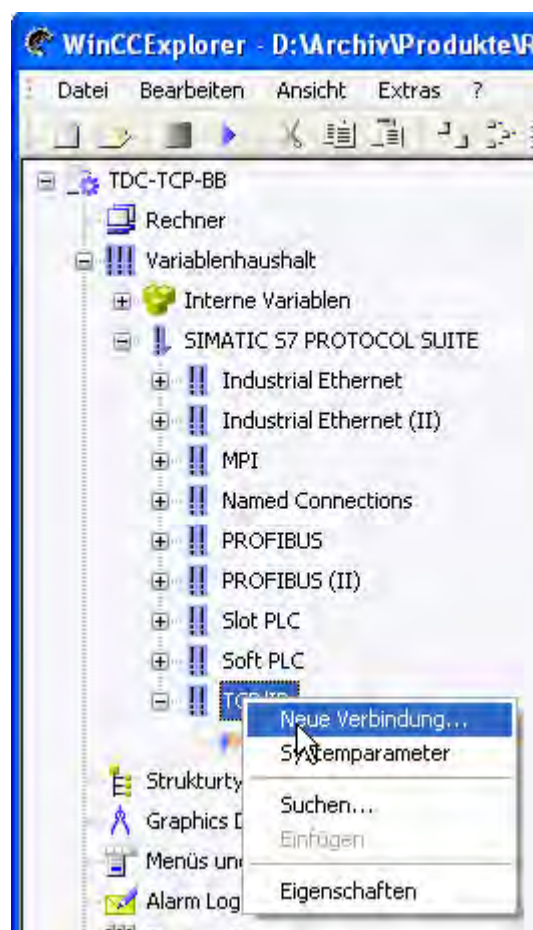




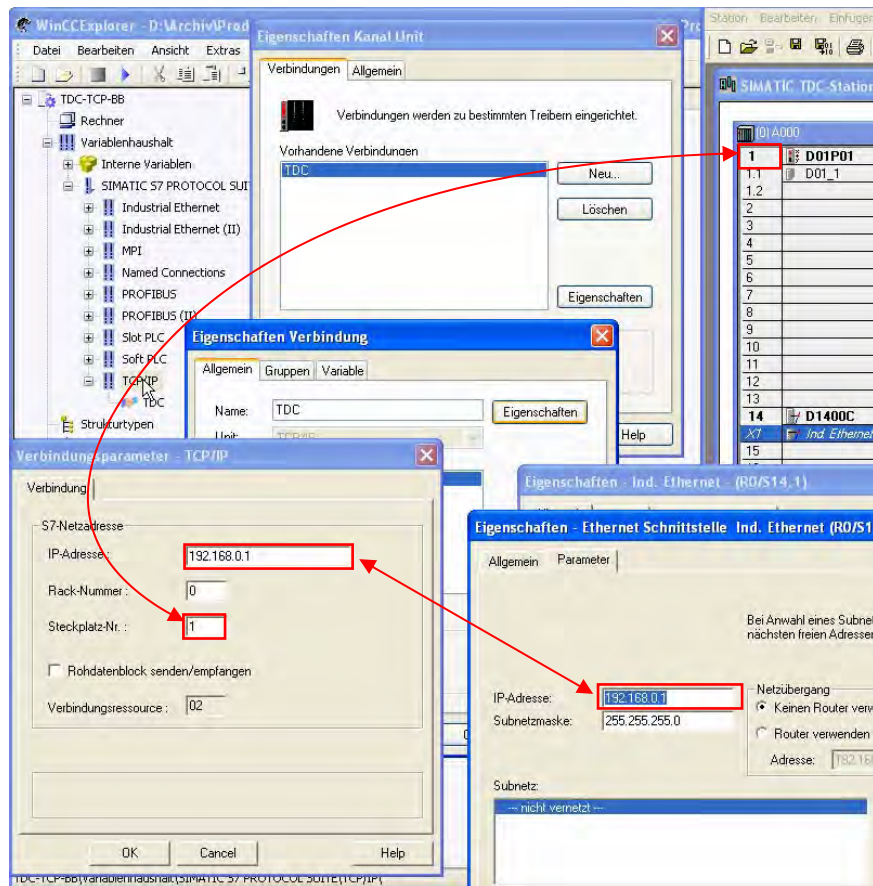
### 3.16.1.3 WinCC- Projektierung

Bei der Projektierung für WinCC ist in folgenden Schritten vorzugehen:

1. Start des WinCC Control Centers.
2. Ein neues Projekt anlegen bzw. ein bereits bestehendes öffnen.
3. Durch Anwählen von **Variablenhaushalt** → **rechte Maustaste** → **Neuen Treiber hinzufügen** → **SIMATIC S7 Protocol Suite.CHN** → **Öffnen** einen neuen Treiber anlegen. Ist dieser bereits vorhanden, dann mit dem nächsten Schritt fortfahren.
4. Durch Anwählen von **TCP/IP** → **rechte Maustaste** → **Neue Verbindung** eine neue Verbindung anlegen.

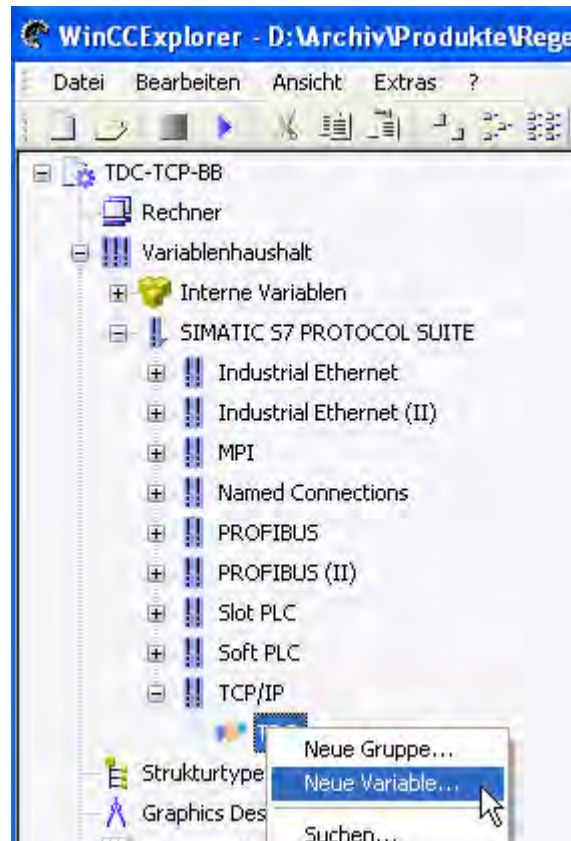


5. Dazu der Verbindung im Dialog einen Namen geben, auf den Button Eigenschaften drücken und die Parameter für die Verbindung eingeben (TCPI/IP-Adresse und Steckplatz, können aus HW-Konfig entnommen werden; siehe dazu folgendes Bild).



Nach dem Verlassen der Dialoge mit OK ist eine neue TCP/IP-Verbindung erstellt.

6. **Anlegen der Variablen:** Die eben angelegte Verbindung mit der rechten Maustaste anwählen und im eingblendeten Menu "Neue Variable" auswählen.



7. Im aufgehenden Dialog einen Variablennamen eingeben (z.B. Funktionsbausteinname\_Anschlussname, es kann aber auch jeder andere Name angegeben werden).  
 Der aus CFC-Projektierung oder dem, bei der CFC-Compilierung erzeugten Adressbuch entnommene Datentyp des gewählten Konnektors kann unter „Datentyp“ eingestellt werden (eine Referenztafel der Datentypen ist im Anschluss zu finden).  
 Der Adressdialog wird durch betätigen des Buttons "Wählen" geöffnet.  
 In diesem Adressdialog die DB-Nummer und den Offset angeben.  
 Diese Daten für den jeweiligen Anschluss aus dem, bei der CFC-Übersetzung erzeugten Adressbuch entnehmen (siehe nächstes Bild).

The screenshot shows the WinCC Explorer interface with the 'Eigenschaften Variable' dialog box open. The 'Allgemein' tab is selected, showing the variable name 'Output\_Integrator', data type 'Gleitkommazahl 32-Bit IEEE 754', length '4', and address 'DB1001.DD24576'. A second dialog, 'Eigenschaften Adresse', shows the address details: 'Datenbereich' set to 'DB', 'DB-Nr.' set to '1001', and 'DD' set to '24576'. A Notepad window titled 'BGTCR1\_1.adr' displays the variable declaration code, with red boxes highlighting '1001', '24576', and 'NF:1.0'. Red arrows and labels point from these code elements to the corresponding fields in the dialog boxes: 'DB-Nummer' points to '1001', 'Offset' points to '24576', and 'Datentyp' points to 'NF:1.0'.

**Eigenschaften Variable**

Name: Output\_Integrator  
 Datentyp: Gleitkommazahl 32-Bit IEEE 754  
 Länge: 4  
 Adresse: DB1001.DD24576

**Eigenschaften Adresse**

Adresse  
 -Adressbeschreibung  
 Cpu: [Dropdown]  
 Datenbereich: DB DB-Nr.: 1001  
 Adressierung: Doppelwort  
 DD: 24576

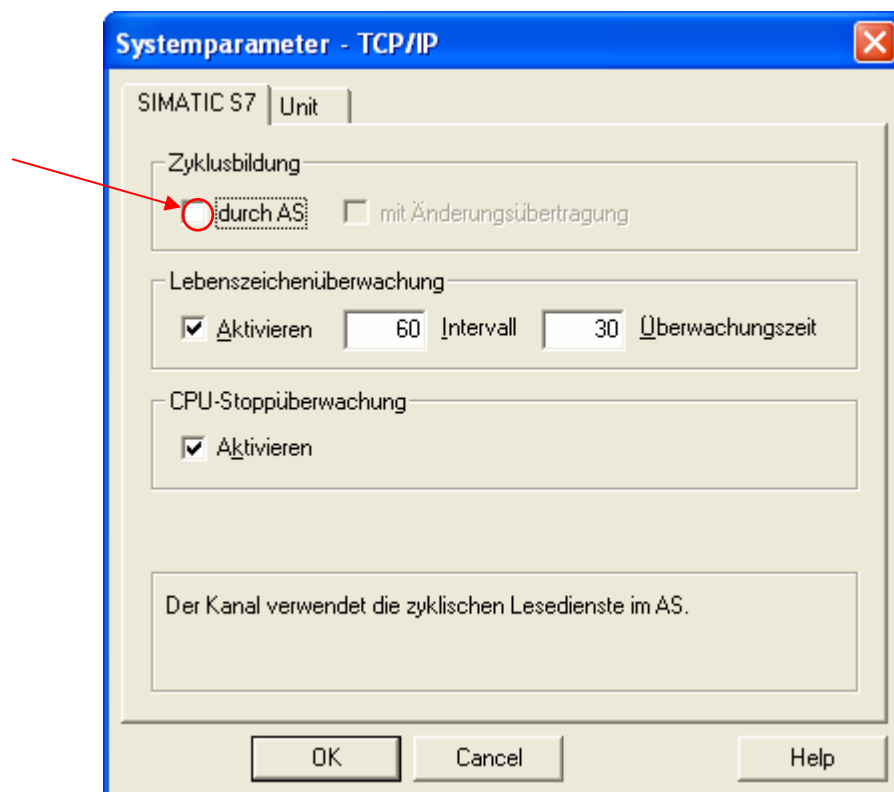
**BGTCR1\_1.adr - Notepad**

```

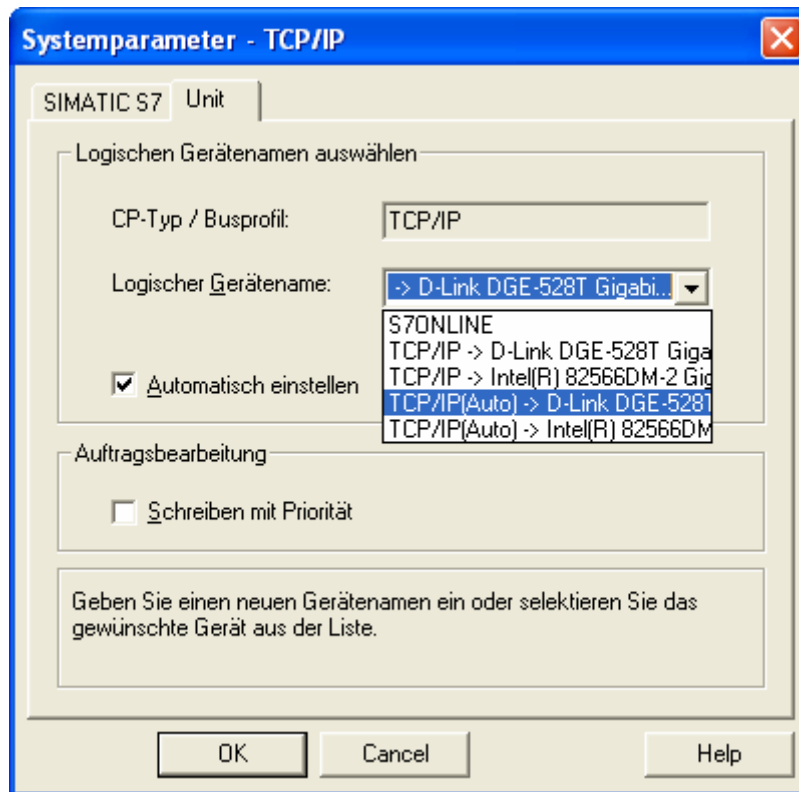
V 5.0.0
31.01.97
BGTCR1
4581DP
BGTCR1
08.01.2009:14:26
1.0
T1:1.0
T2:4.0
T3:16.0
T4:64.0
T5:256.0
I1:0
I2:0
I3:0
I4:0
I5:0
I6:0
I7:0
I8:0
FP:Application:1:
FB:Integrator:T1:2
X :8091 1001 0:NF:1.0:E:::::F
Y :8091 1001 24576:NF:1.0:::A:::::F
    
```

STRUC V.4.x Datentyp	D7-SYS Datentyp	Bezeichnung
B1	BO	Bool
I2 / N2 / O4	I	Integer
I4 / N4 / O4	DI	Double-Integer
NF	R	Real
V1	BY	Byte
V2	W	Word
V4	DW	Double-Word
NS	S	String
TF	TS	SDTime
IK, NK, CR, MR, TR, RR	GV	Global

8. Nach Eingabe der entsprechenden Daten und Verlassen der Dialoge mit OK ist für den gewählten Bausteinanschluss eine Variable in WinCC angelegt.
9. Für weitere gewünschte Bausteinanschlüsse ist die Prozedur ab Schritt 6 zu wiederholen.
10. Im Eingabefenster „Systemparameter“ der Treiber (TCP / MPI → rechte Maustaste → Systemparameter) darf das Häkchen für “Zyklische Lesedienste im AS nutzen” **nicht** gesetzt sein (siehe nächstes Bild).



11. Ebenfalls unter „Systemparameter“ ist der „Logische Geräte name“ aus zu wählen.



12. Auf die so angelegten Variablen kann jetzt in den Bildprojektierungen referenziert werden.

### 3.16.2 Projektierungsvariante „S7DB“

Bei Verwendung des Funktionsbausteins „S7DB\_P“ kann die Markierung der Anschlüsse in den CFC-Plänen und die Erzeugung des Adressbuches entfallen.

Statt dessen ist der Funktionsbaustein „S7DB\_P“ mit den entsprechenden Zeigerbausteinen zu projektieren. Die Bausteinanschlüsse, auf die von WinCC zugegriffen werden soll, müssen über „Zeigerbausteine“ auf den Funktionsbaustein S7DB\_P verdrahtet werden. Der Funktionsbaustein S7DB\_P legt für diese Daten einen Datenbaustein an. In folgenden Schritten ist vorzugehen:

1. Funktionsbaustein S7DB\_P projektieren. Mit rechtem Mausklick in den Konnektor „XDB“ „Verschaltung zu Operanden“ an wählen und die gewünschte DB-Nummer angeben.
2. Jeden zu visualisierenden Konnektor (z.B. wie im nächsten Bild: Konnektoren „X“ und „Y“ des Funktionsbausteins „Integrator“) mit einem eigenen, dem jeweiligen Konnektortyp entsprechenden Zeigerbaustein der Funktionsbaustein Familie „Zeiger Kom“, (z.B.: Funktionsbausteine „DRD“ für Realvariable lesen) verbinden.
3. Konnektor „PTR“ des S7DB\_P mit den „PTR“-Konnektoren sämtlicher Zeigerbausteine verbinden.
4. Die Datenbaustein-Nummer (z.B.: DB1) aus der Randleiste und den Offset dem Zeigerbausteinkonnektor (OF1 + OF2) der CFC-Projektierung entnehmen, den Adressdialog durch Betätigen des Buttons „Wählen“ öffnen und in diesem Adressdialog die DB-Nummer und den Offset angeben. (siehe nächstes Bild). Abschließend Eingabefenster mit OK schließen.

The screenshot shows the SIMATIC Manager interface with two dialog boxes open: 'Eigenschaften Variable' and 'Eigenschaften Adresse'. The 'Eigenschaften Variable' dialog shows the variable name 'Output\_Integrator\_S7DB' and data type 'Gleitkommazahl 32-Bit IEEE 754'. The 'Eigenschaften Adresse' dialog shows the data range 'DB', data number '1', and address 'DD 4'. Below these, a ladder logic diagram is shown with three rungs. The first rung is labeled 'DB-Nummer' and points to the 'DB 1' parameter in the 'DRD' function block. The second rung is labeled 'Offset' and points to the '4' parameter in the 'DWR' function block. The third rung is labeled 'Datentyp' and points to the 'TI' parameter in the 'DWR' function block. The diagram also shows other function blocks like 'STDB\_P' and 'Integrator'.

Alle weiteren Projektierungsschritte unterscheiden sich nicht zur Vorgehensweise bei Verwendung von B&B-Funktionen.



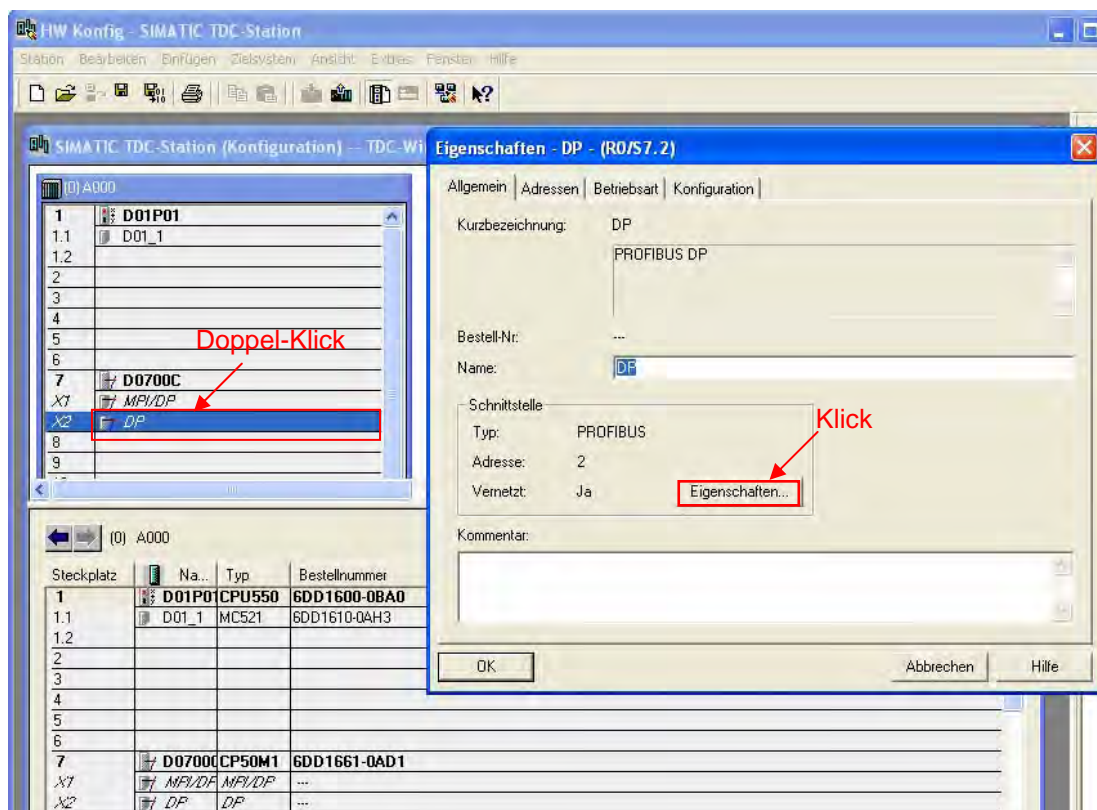
### 3.16.3 Kopplungsvarianten MPI und PROFIBUS DP

Im folgenden Kapitel sind die zu berücksichtigenden Abweichungen einer MPI- oder DP -Kopplung im Vergleich zur TCP-Kopplung beschrieben.

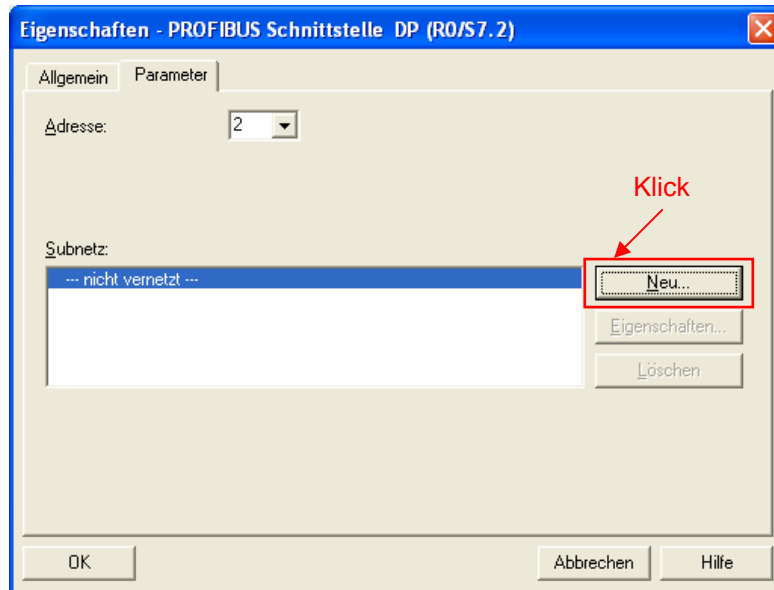
#### 3.16.3.1 Hardwarekonfiguration

Für eine PROFIBUS DP- oder MPI-Kopplung ist wie im Folgenden beispielsweise gezeigt eine CP50M1 mit der entsprechenden Kopplungsart zu projektieren.

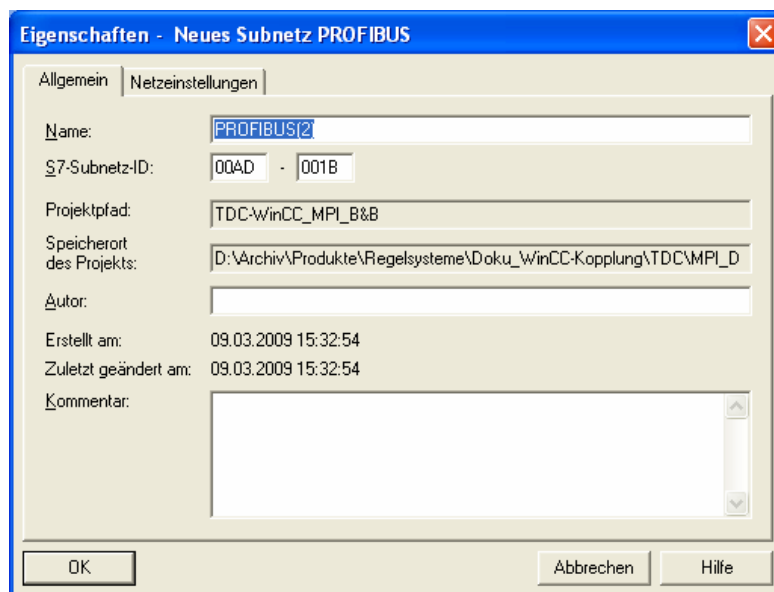
1. Doppel-Klick auf die gewünschte Schnittstelle der CP50M1.
2. Unter der Lasche „Allgemein“ Klick auf „Eigenschaften“.



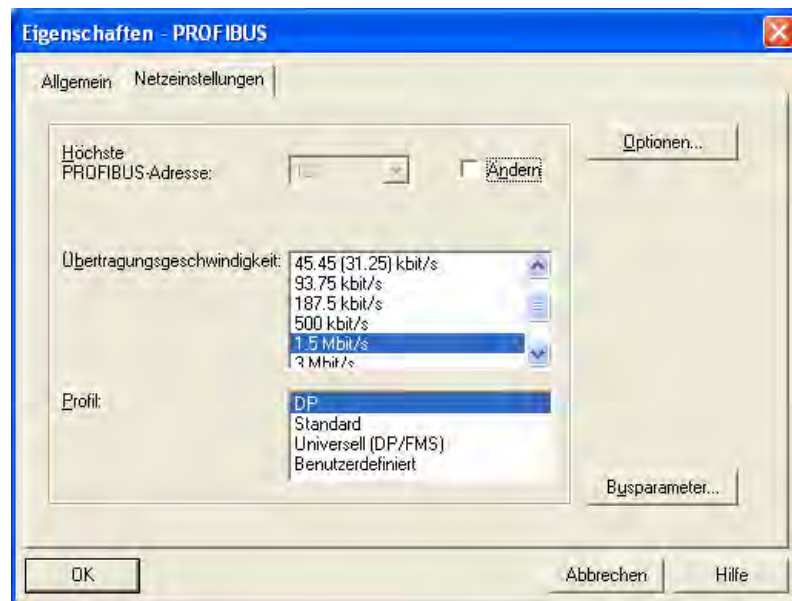
3. Im nächsten Fenster durch einen Klick auf „Neu“ ein neues „Subnetz“ einfügen.



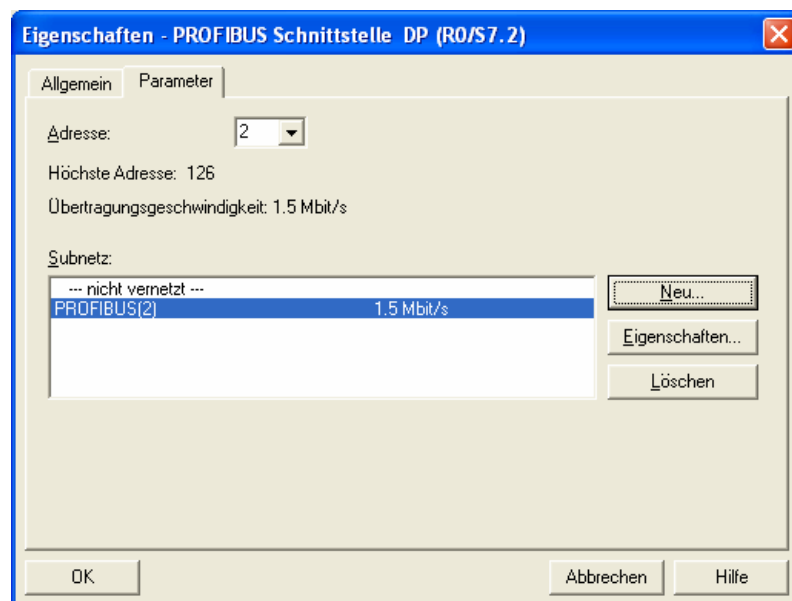
4. Namen ggf. ändern und Lasche „Netzeinstellungen öffnen...“



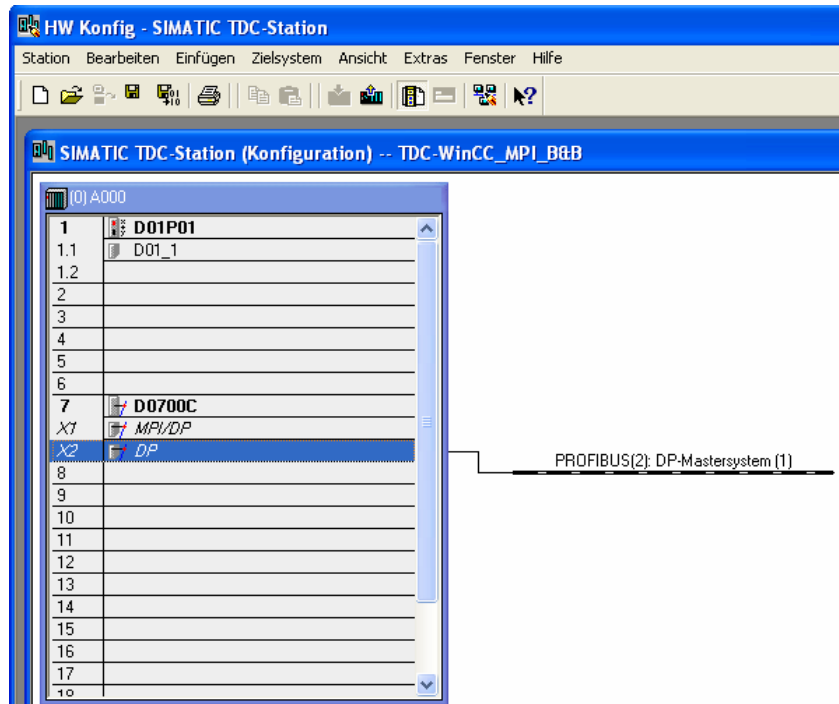
5. Im nächsten Fenster die gewünschte Baudrate einstellen.



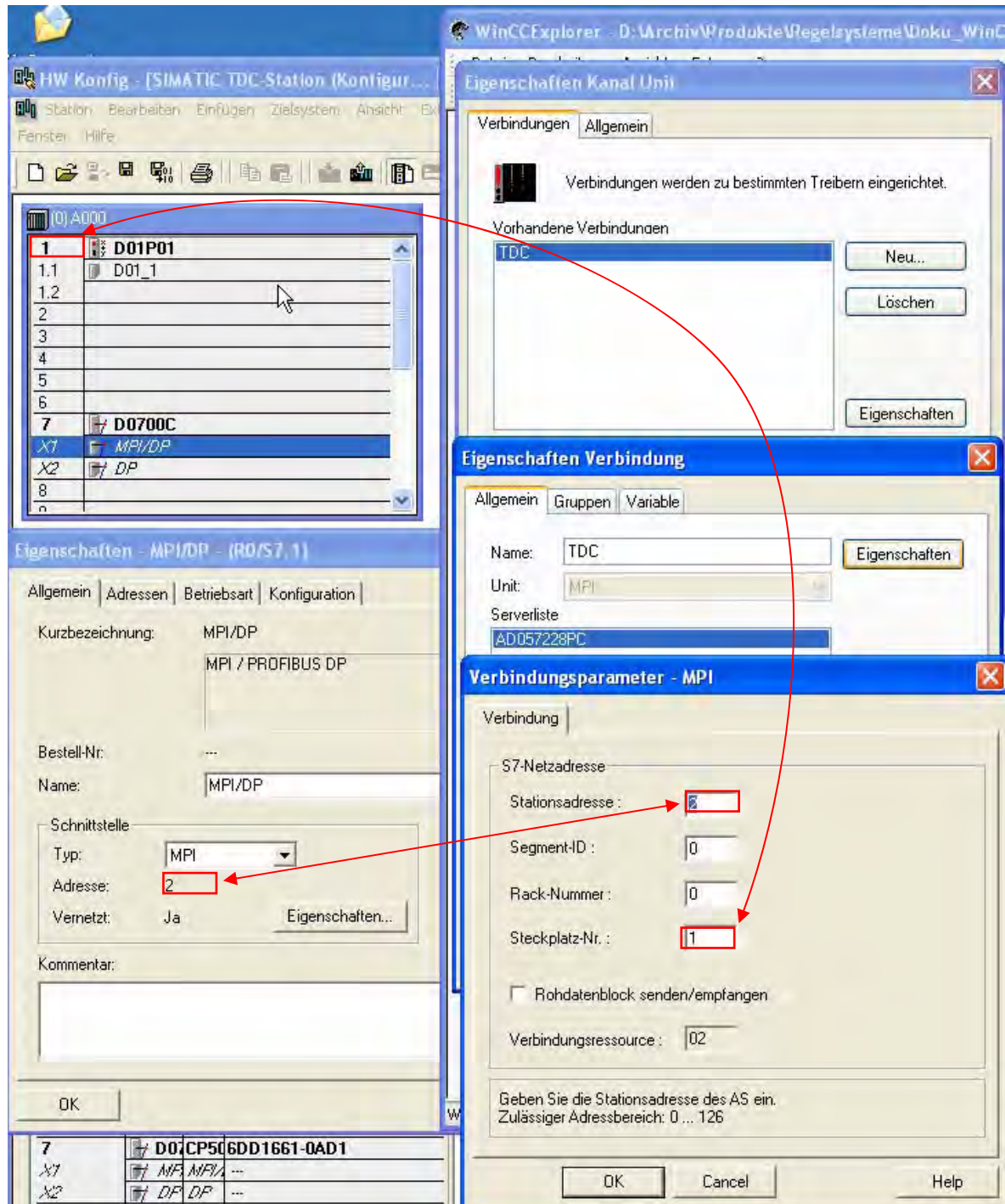
6. Dieses Fenster schliessen und im nächsten Fenster die Adresse einstellen. Anschliessend alle anderen Fenster mit „OK“ schliessen.



7. Damit ist der PROFIBUS DP- oder MPI-Strang, wie im nächsten Bild gezeigt, projektiert.

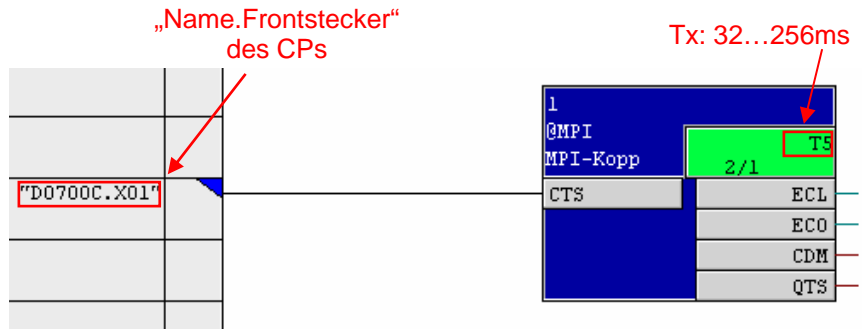


Zwischen der TDC-Station und WinCC sind wie folgt Stationsadresse und Steckplatz abzugleichen:

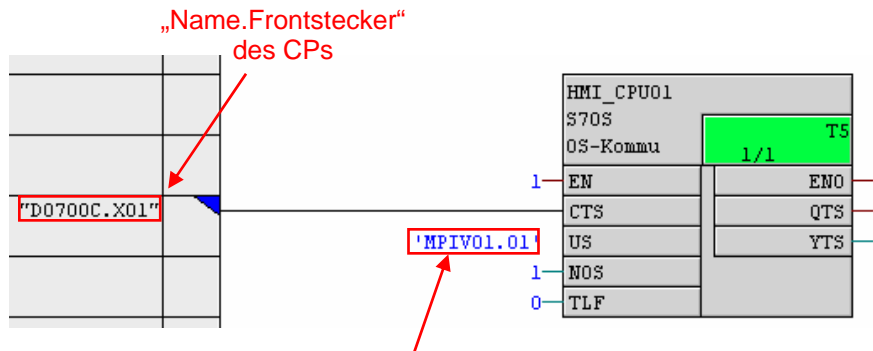


### 3.16.3.2 CFC-Projektierung

Für die Initialisierung der MPI- oder DP-Kopplung ist der entsprechende Zentralbaustein „@MPI“ oder @PRODP“ in einer Task zwischen 32 und 256 ms zu projektieren, dessen CTS-Konnektor auf den Steckplatz der CP50M1 und deren Frontstecker zu verschalten ist.



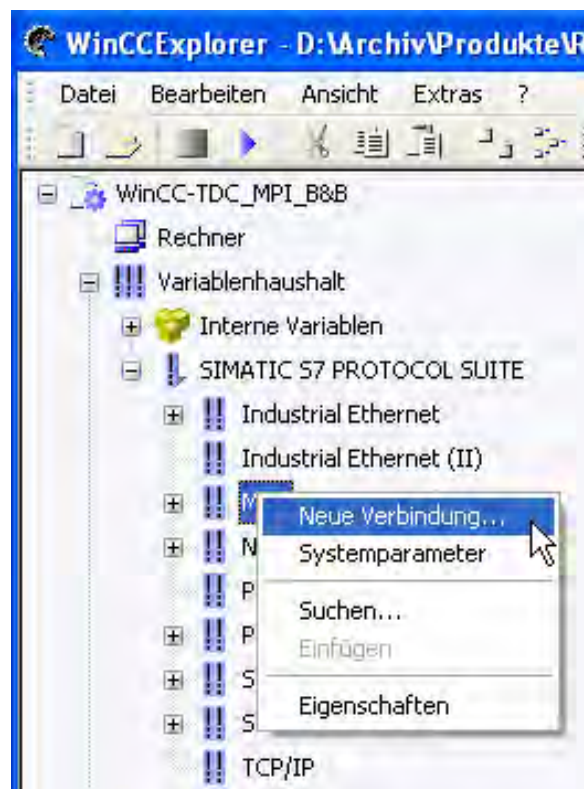
Zusätzlich ist der Funktionsbaustein „S7OS“ ebenfalls in einer Task zwischen 32 und 256ms zu projektieren und wie abgebildet zu verschalten:



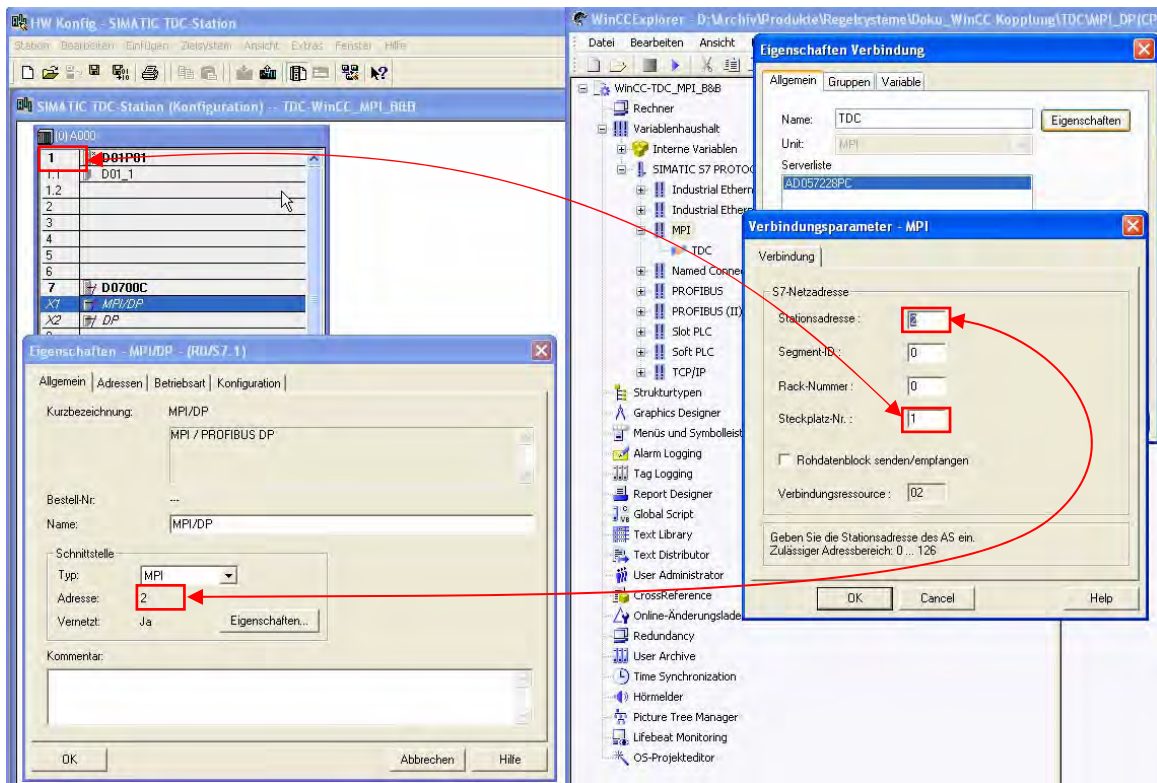
Beliebiger, eindeutiger, 6-stelliger Name.CPU-Steckplatznummer

### 3.16.3.3 WinCC-Projektierung

1. Durch Anwählen von **Variablenhaushalt** → **rechte Maustaste** → **Neuen Treiber hinzufügen** → **SIMATIC S7 Protocol Suite.CHN** → **Öffnen** einen neuen Treiber anlegen.  
Ist dieser bereits vorhanden, dann mit dem nächsten Schritt fortfahren.
2. Durch Anwählen von **MPI** → **rechte Maustaste** → **Neue Verbindung** eine neue Verbindung anlegen.  
Soll eine PROFIBUS DP-Kopplung verwendet werden, so ist an dieser Stelle in gleicher Weise **PROFIBUS DP** → **rechte Maustaste** → **Neue Verbindung** anzulegen.



3. Dazu der Verbindung im Dialog einen Namen geben, auf den Button **Eigenschaften** drücken und die Parameter für die Verbindung eingeben
4. In die WinCC-Eingabemaske „Verbindungsparameter“ sind die Stationsadresse und die Steckplatznummer wie im folgenden Bild gezeigt, aus der HW-Konfiguration zu übernehmen.



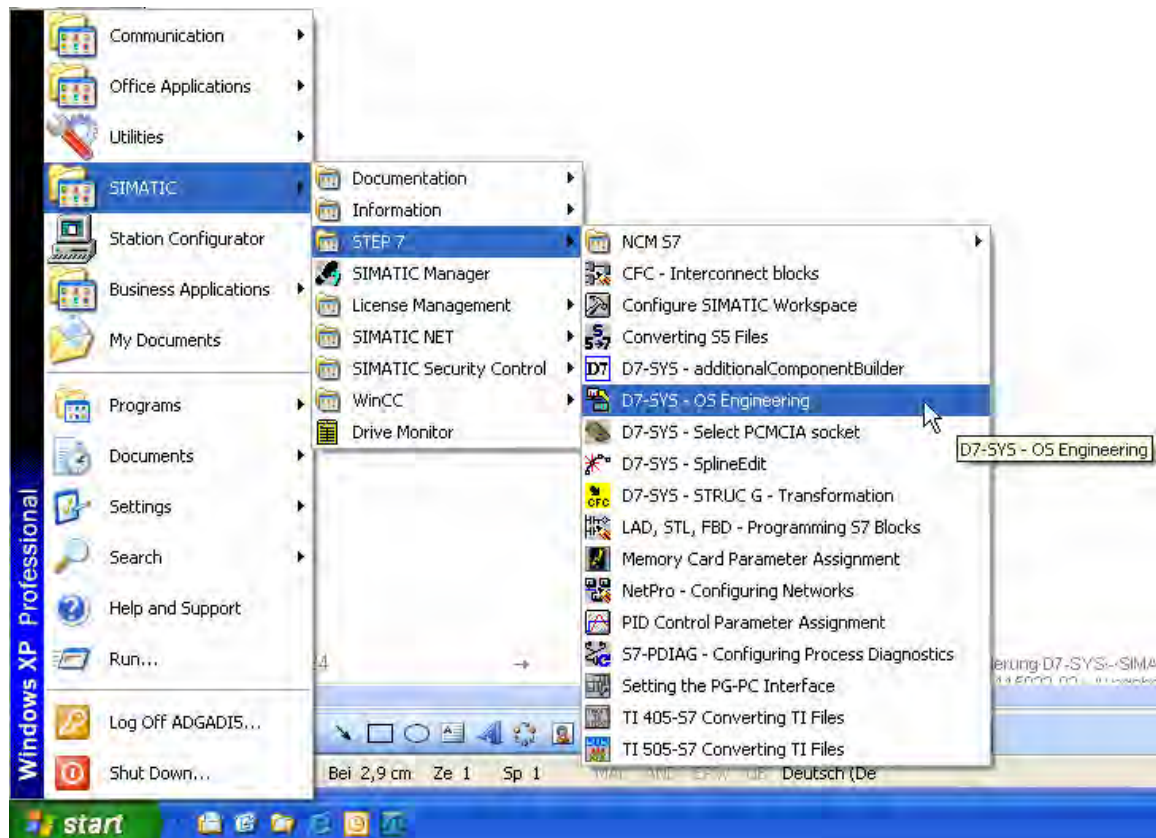
Die weiteren Projektierungsschritte unterscheiden sich nicht von der einer TCP/IP-Kopplung und können somit aus dem vorangestellten Kapitel übernommen werden.



### 3.16.4 Projektierung mit dem „D7-SYS-OS-Engineering-Tool“

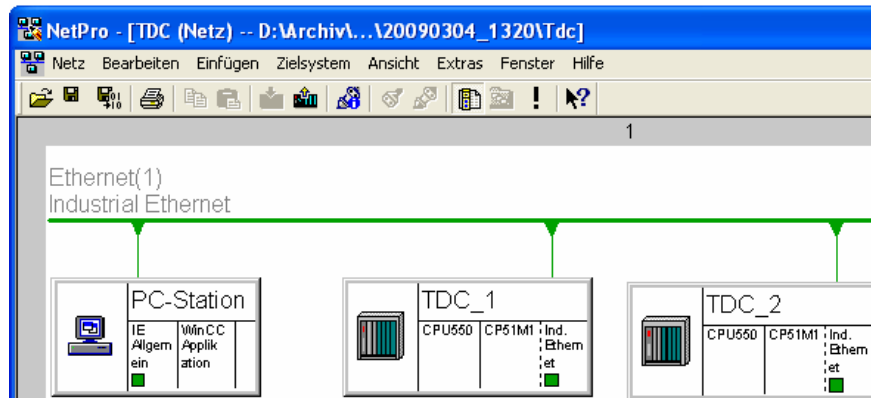
Das „D7-SYS-Engineering-Tool“ im Weiteren auch „Mapper“ genannt, legt für die ausgewählten Konnektoren der CFC-Funktionsbausteine Tags an, die von WinCC weiterverarbeitet werden können. Das folgende Kapitel beschreibt die Handhabung dieses Tools.

Aufruf des Mappers:



Voraussetzungen für den Mapp-Vorgang:

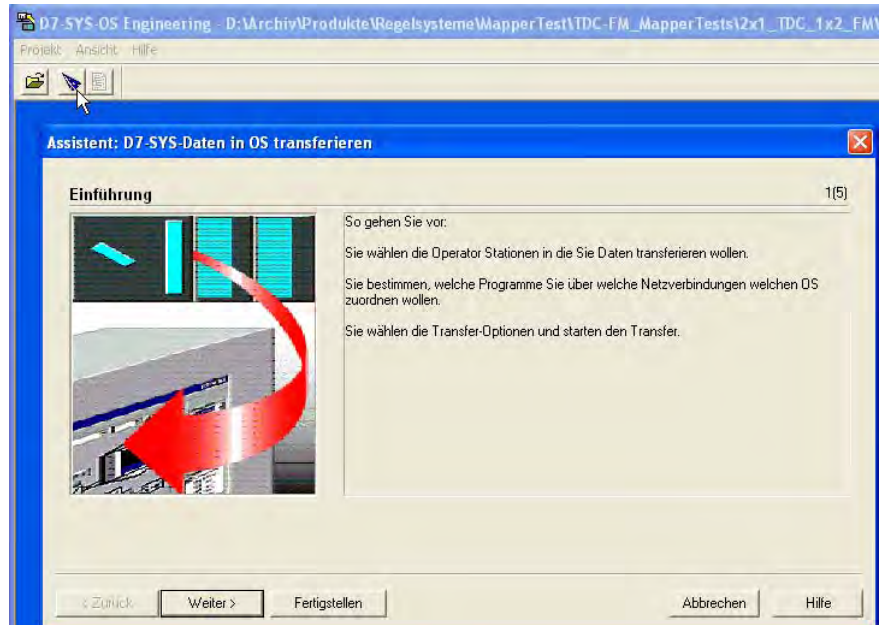
- Selektion der CFC-Funktionsbaustein-Konnektoren und abschließende Kompilierung mit aktivierter Option „Adressbuch erzeugen“, wie es im vorangestellten Kapitel beschrieben wurde.
- Eingefügte PC-Station mit Kommunikationsbaugruppe und WinCC-Applikation:
- Die „NetPro“-Konfiguration sollte dahingehend überprüft werden, ob alle Stationen über die gewünschten Kopplungsarten miteinander verbunden sind, wie beispielsweise hier über TCP/IP. In gleicher Weise ist mit MPI oder DP-Kopplungen zu verfahren.
- Es kann nur nach WinCC transferiert werden, wenn der WinCC-Explorer gestartet und das entsprechende WinCC-Projekt geladen wurde.



Nach dem Aufruf ist über das linke Icon „Öffnen“ das Zielprojekt auszuwählen.

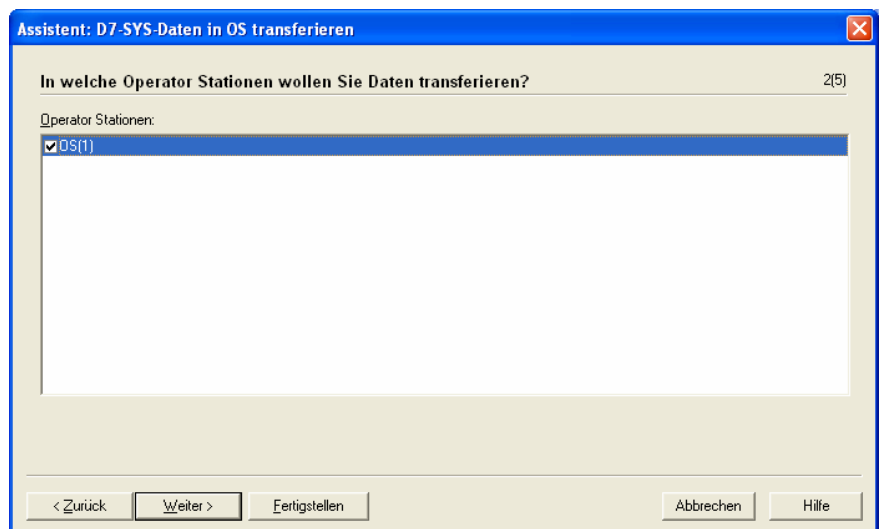


Durch Klicken auf das 2-te Ikon (Zauberhut) wird ein Assistent gestartet:



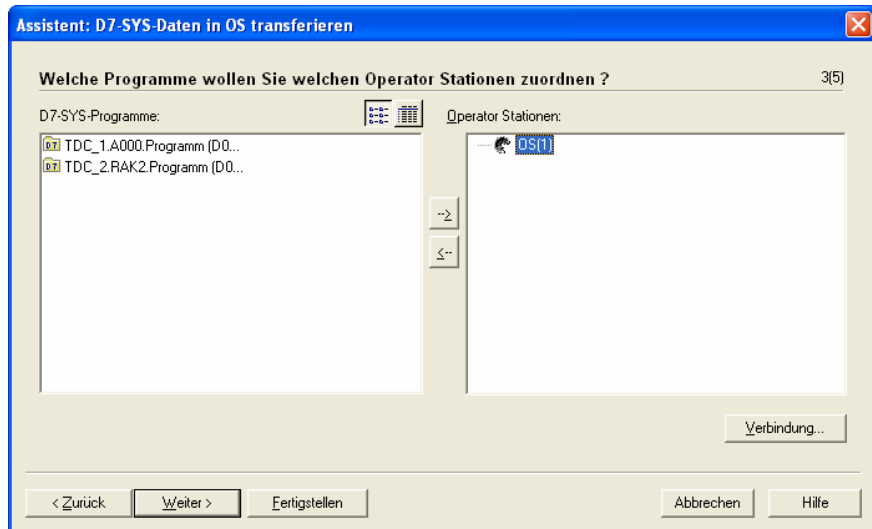
Anschließend auf „Weiter“ klicken:

Auswahl der Operator-Station erfolgt im nächsten Bild:

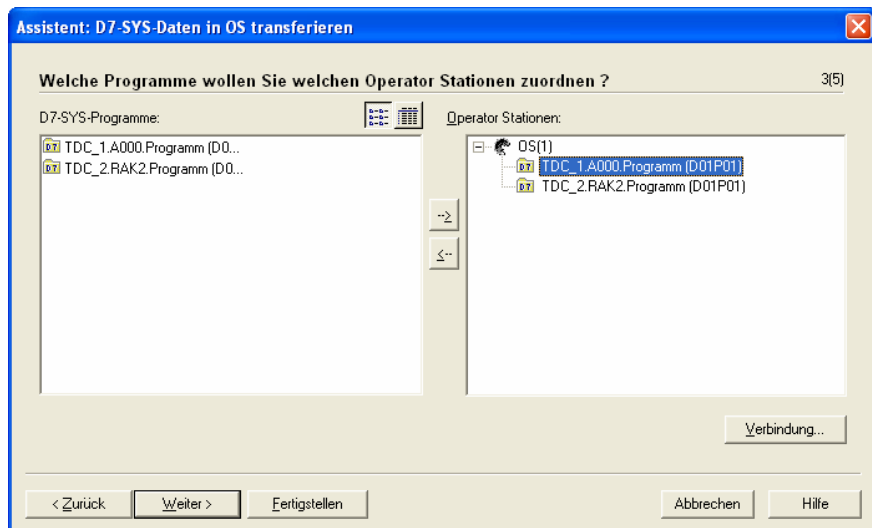


Anschließend auf „Weiter“ klicken:

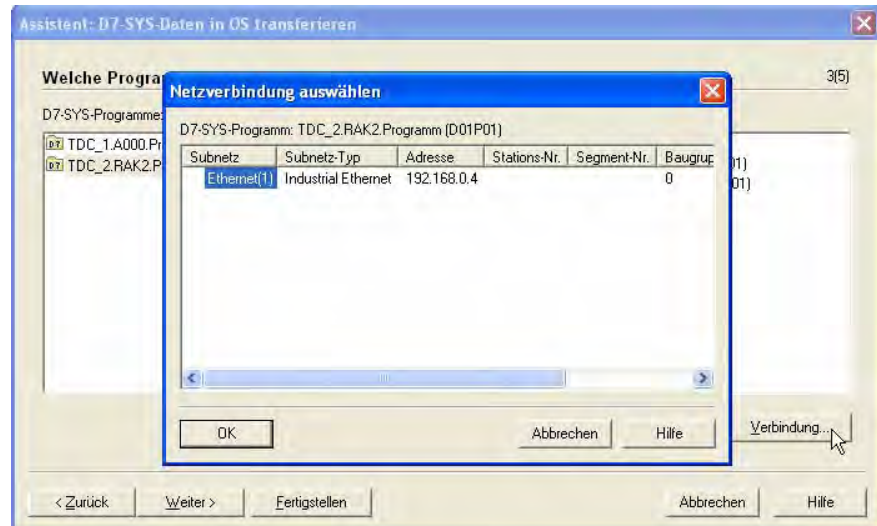
Die Auswahl und Zuordnung der Programme zu den Operator-Stationen erfolgt im nächsten Schritt:



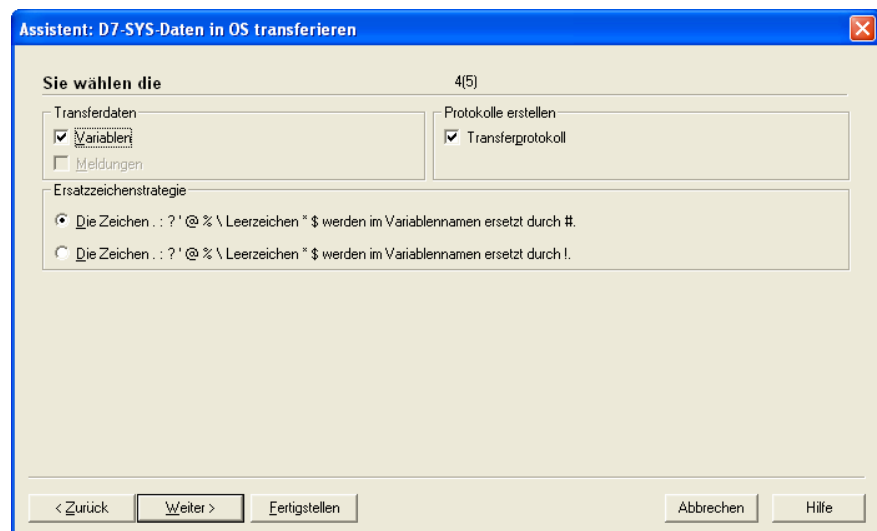
Anzeigen der ausgewählten Programme:



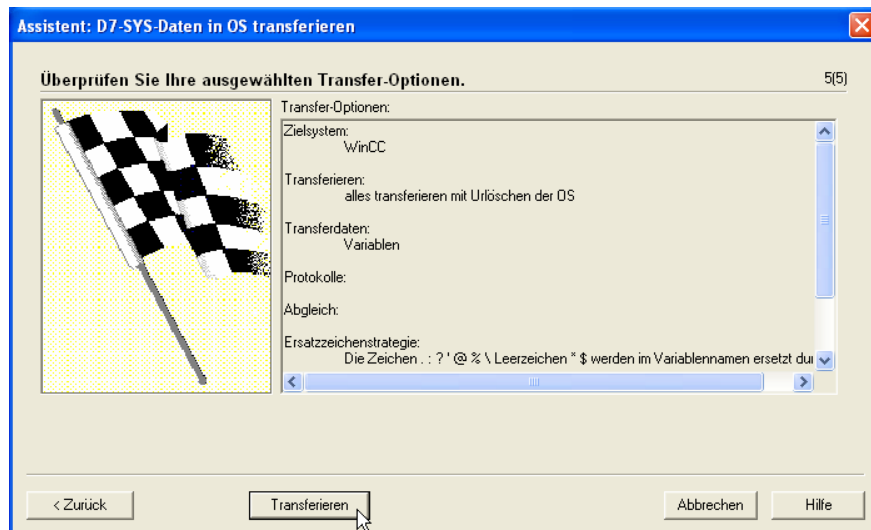
Sollen die Verbindungsparameter überprüft werden, ist auf „Verbindung“ zu klicken. Ist die „NetPro“-Konfiguration korrekt durchgeführt worden, ist dies nicht zwingend erforderlich. Sind jedoch verschiedene Kopplungen (TCP/IP, MPI oder DP) verwendet, so ist hier die gewünschte Verbindung auszuwählen.



Anschließend auf „Weiter“ klicken:

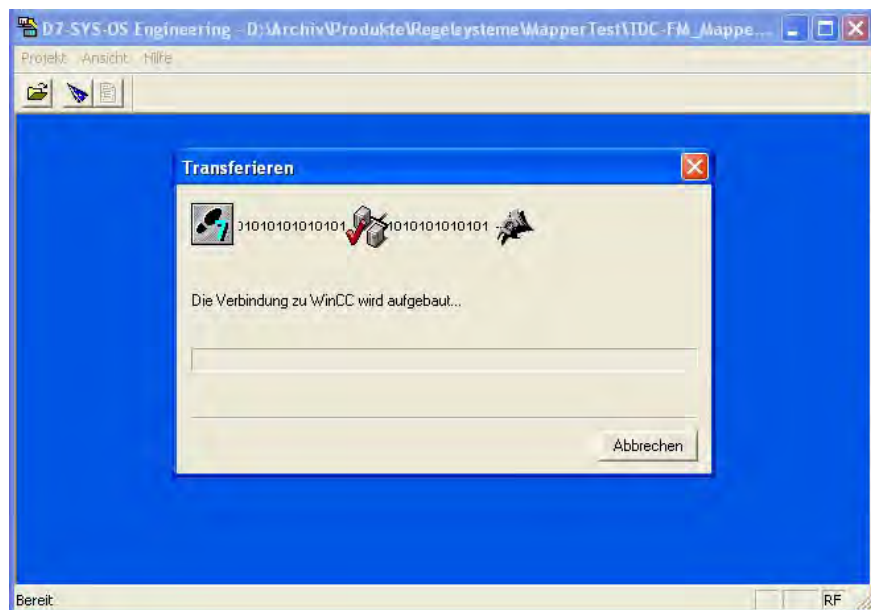


Anschließend auf „Weiter“ klicken:



Abschließend durch klicken auf „Transferieren“ den eigentlichen Mapp-Vorgang starten:

Transferieren (Mappen) läuft:

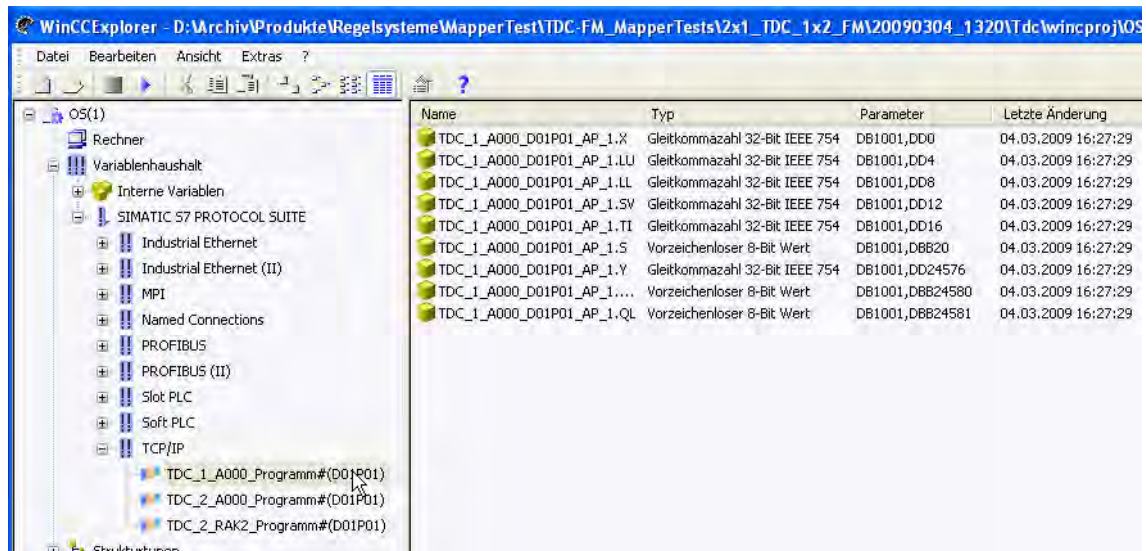


Damit ist die Übertragung abgeschlossen:

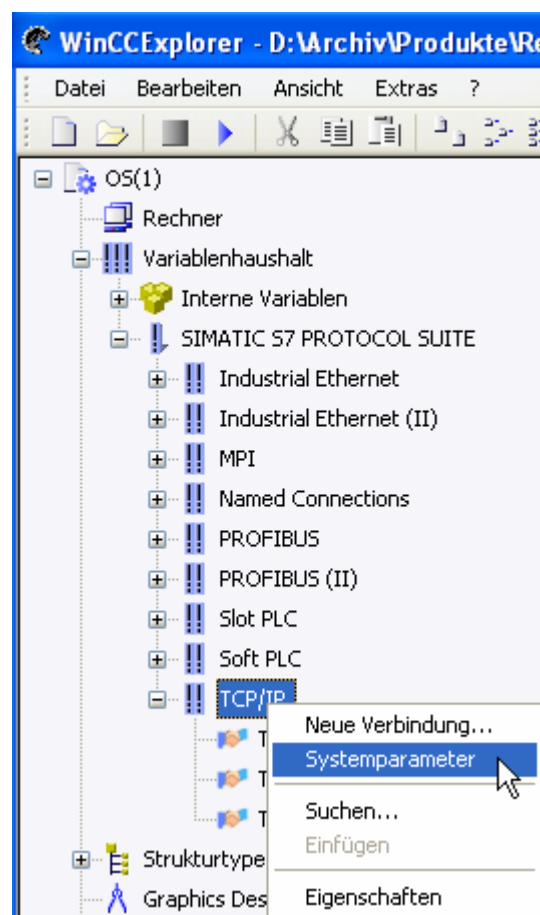


Mit „OK“ klicken beenden.

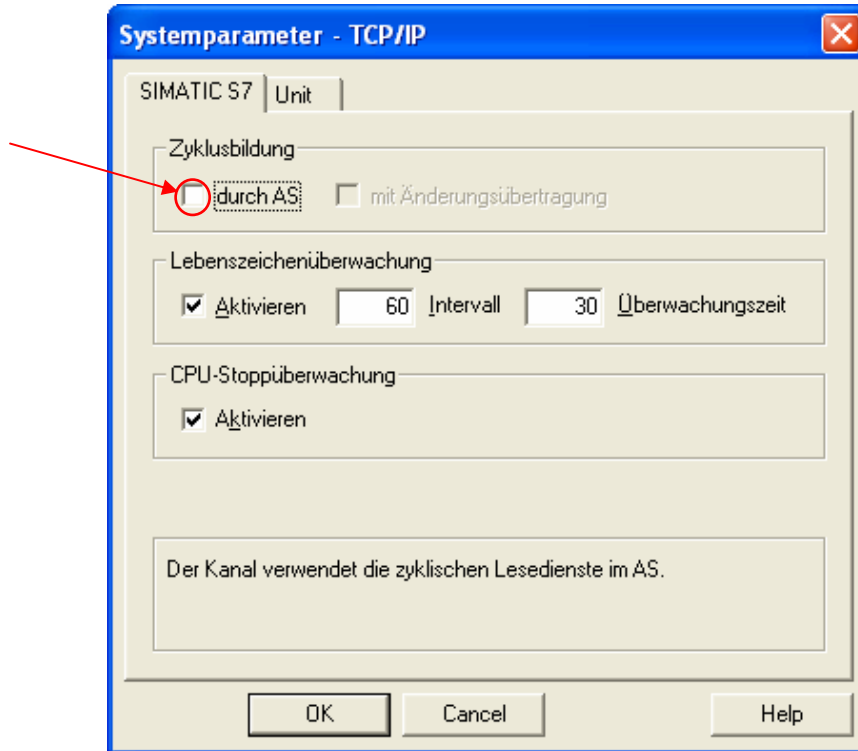
Im WinCC-Explorer sind die erzeugten Tags angelegt:



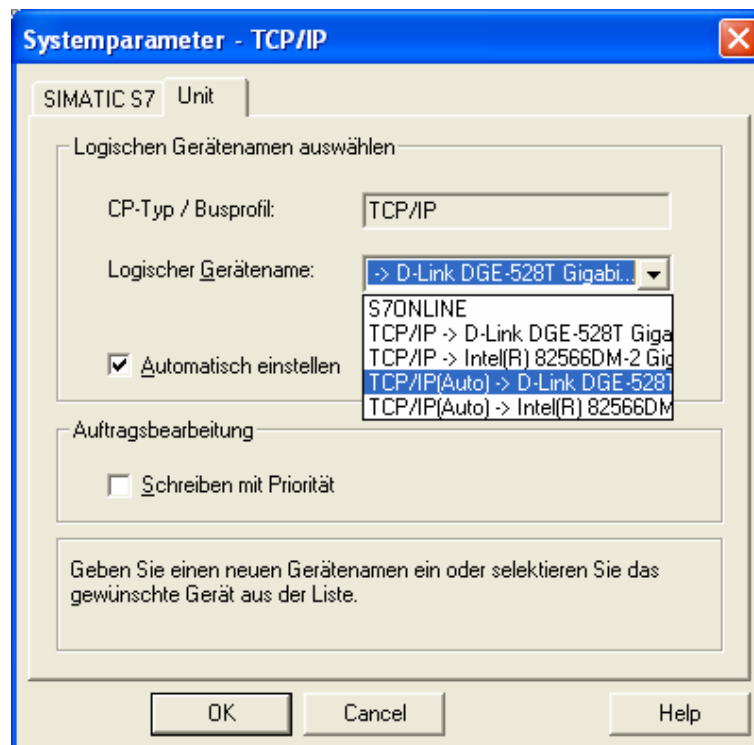
Zu überprüfen sind nun noch die in den folgenden Bildern gezeigten Systemparameter der Kopplung:



Zyklusbildung durch Automatisierungssystem deaktivieren



Gerätenamen auswählen:



Nach Änderung der Systemparameter ist ein WinCC-Neustart erforderlich. Anschliessend kann auf die projektierten Variablen zugegriffen werden.



## 3.17 Kommunikation mit WinCC (TCP/IP)

**Einleitung** Dieses Benutzerhandbuch stellt Ihnen an Hand eines einfachen Projektierungsbeispiels die mögliche Anbindung von WinCC an SIMATIC TDC über eine TCP/IP-Ankopplung vor. Es werden alle notwendigen Projektierungsschritte (incl. Hard- und Softwarevoraussetzungen) beschrieben. Auf die Bedienung der dafür benötigten Softwaretools wird hierbei nicht eingegangen, sondern auf die entsprechenden Benutzerhandbücher verwiesen.

### 3.17.1 Voraussetzungen

#### Software

<b>TDC PMC TCP Kanal-DLL</b>	<p><b>Software-Voraussetzungen</b></p> <ul style="list-style-type: none"> <li>WinCC-Systemsoftware: ab Version 5.0 für Windows NT 4.0</li> <li>SIMATIC TDC: PCS7 ab Version 5.0 mit D7-SYS V5.1</li> </ul> <p><b>Bestellnummer (SIMATIC TDC PMC TCP/IP)</b> 2XV9450-1WC43-0KX0</p> <p><b>Weitere Informationen</b> Siemens AG Industrial Solutions and Services IT Plant Solutions I&amp;S IT PS 3 Werner-von-Siemens-Straße, 60 91052 Erlangen</p> <p>Kontakt: Ihr IT4Industry Team E-Mail: <a href="mailto:info@it4industry.de">info@it4industry.de</a> WWW: <a href="http://www.it4industry.de">http://www.it4industry.de</a></p>
<b>Tools</b>	<p><b>PROBI:</b> Das Projektierungspaket PROBI ist Bestandteil jeder SIMATIC TDC-PMC-Lizenz.</p>

Tabelle 3-52 Software-Voraussetzungen

---

**HINWEIS** Die TDC PMC TCP D-PMC Kanal-Dll kann nur in Verbindung mit WinCC ab V5.0 verwendet werden. Die Installation erfolgt über ein Setup, welches sich auf der Diskette der Produktsoftware befindet.

---

**Hardware** Projektierungsplatz PC:

Netzwerkkarte für TCP/IP, z.B. 3COM
-------------------------------------

Tabelle 3-53 Hardware-Voraussetzungen

## **SIMATIC TDC Hardwareaufbau**

System:	D7-SYS ab V5.1	
Baugruppenträger:	UR5213	21 Steckplätze mit Lüfter
Steckplatz 1:	CPU550	CPU (mit lokaler Serviceschnittstelle)
Steckplatz 1.1:	MC500	Speichermodul 4 Mbyte Flash
Steckplatz 2:	CP50M1 / CP50M0	Koppelspeicher
Steckplatz18:	CP51M1/CP5100	Anschaltung für TCP/IP

*Tabelle 3-54 Hardwareaufbau für die Beispielprojektierung*

### **3.17.2 Prozessvariablen**

Mit dem Projektierungstool CFC muss eine SIMATIC TDC Station konfiguriert und parametrierung und ein beliebiger Testplan erstellt werden. Die Hardwarekonfiguration ist unter SIMATIC TDC - Hardwareaufbau beschrieben. Wie mit dem CFC eine Projektierung zu erstellen ist, wird an dieser Stelle nicht im Detail eingegangen. Lesen sie bitte hierzu die Projektierungsanleitung [4].

#### **3.17.2.1 SIMATIC TDC-Projektierung**

Der CFC-Plan für die WinCC-Ankopplung muss nicht in einem separaten Plan erfolgen, wird aber wegen der Übersichtlichkeit empfohlen. Für die Kopplung zwischen SIMATIC TDC und WinCC für Prozessvariablen werden folgende Funktionsbausteine benötigt :

- LI - LAN-Interface Baustein
- VM - Visualisierungs Baustein
- VI - Interface Baustein
- VC - Konzentrador Baustein
- CI - Interface Baustein
- SER02 - Kommunikations Baustein

Die Bausteine werden wie folgt verdrahtet:

(es werden nur die relevanten Anschlüsse beschrieben)

**FB LI**

Anschlussname	Bedeutung	Beispiel
<b>CTS</b>	Name der verwendeten Anschaltung (CP51M1/CP5100)	D1800C
<b>AT</b>		
<b>AR</b>	[Kanalname].[Protokoll]-[Lokale Port-Nr.]	ARC01.T-19000 mit T für TCP/IP
<b>NA</b>	maximale parallele Aufträge von WinCC	20
<b>NC</b>	WinCC ID	0
<b>COM</b>	Kommunikationsmedium TCP/IP	3
<b>CCV</b>	Verbindung mit VM, Anschluss CVP	<VM.CVP
<b>CCF</b>	keine Verbindung mit FM-Baustein	16#0
<b>CCB</b>	keine Verbindung mit MM-Baustein	16#0

**FB VM**

Anschlussname	Bedeutung	Beispiel
<b>NA</b>	Summe der für den VM reservierten Aufträge	20
<b>NL</b>	Anzahl der LI - Bausteine	1
<b>NV</b>	Anzahl der VI Bausteine	1
<b>MEM</b>	Default	0
<b>TGL</b>	Default	0
<b>CVP</b>	Verbindung mit LI.CCV,VI.CCV	>(LI.CCV,VI.CCV)

**FB VI**

Anschlussname	Bedeutung	Beispiel
<b>CTS</b>	Prozessorname	D01P01
<b>AT</b>	Sendekanalname zum VC	'CMDVCH,
<b>AR</b>	Empfangskanalname vom VC	ACKVCH'
<b>CCV</b>	Verbindung mit VM, Anschluss CVP	'VM.CVP'

**FB VC**

Anschlussname	Bedeutung	Beispiel
<b>CTS</b>	Prozessorname	D01P01
<b>AT</b>	Sendekanalname zum VI	'ACKVCH'
<b>AR</b>	Empfangskanalname vom VI	'CMDVCH'
<b>NC</b>	Anzahl angeschlossener CI's	1
<b>CVP</b>	Verbindung mit CI, Anschluss CCV	>(CI.CCV)

**FB CI**

Anschlussname	Bedeutung	Beispiel
CTS	Prozessorname	D01P01
AT	Sendekanalname zum SER02	'CMDH'
AR	Empfangskanalname vom SER02	'ACKH'
ADT	Datenkanalname vom SER02	'DATH'
CCV	Verbindung mit VC, Anschluss CVP	<VC.CVP

**FB SER02**

Anschlussname	Bedeutung	Beispiel
CTS	Prozessorname	D01P01
AT	Sendekanalname zum CI	'ACKH'
AR	Empfangskanalname vom CI	'CMDH'
ADT	Datenkanalname zum CI	'DATH'
CLT	Länge Sendekanal	116
CLR	Länge Empfangskanal	524
CLD	Länge Datenkanal	432
TPD	für B&B	0
NL	Maximale Anzahl MWL's	50
NV	Max. Anzahl Messwerte (Anschlüsse)	1000

**HINWEIS**

Es muss zusätzlich der Zentralbaustein @GLOB für den Koppelspeicher und der Kopplungszentralbaustein @TCPIP projiziert werden (in obigem Fall auch der @LOCAL, da alle Anschlüsse auf D01P01 verdrahtet).

**FB @TCPIP**

Anschlussname	Bedeutung	Beispiel
CTS	Name der verwendeten Anschaltung (CP51M1/CP5100)	'D1800C'

**FB @GLOB**

Anschlussname	Bedeutung	Beispiel
CTS	Koppelspeicher-Baugruppenname	'D0200A'
CDV	Speicherneuaufteilung (1)	0

**3.17.2.2 WinCC-Projektierung**

Für die Beispielprojektierung genügt eine einfache WinCC Projektierung mit einigen Ein/Ausgabefeldern.

Auf die WinCC Projektierung wird an dieser Stelle nicht eingegangen. Bitte lesen sie hierfür die umfangreichen WinCC - Projektierungshandbücher. Für den Einstieg in eine erste WinCC Projektierung wird das Handbuch SIMATIC WinCC Getting Started empfohlen.

### 3.17.3 Bitmeldeverfahren

#### **SIMATIC TDC-Projektierung**

Für das Bitmeldeverfahren mit WinCC muss keine zusätzliche Projektierung zur Prozesswertbeobachtung gemacht werden. Die Auswahl welches Bit einer Variablen welche Meldung auslöst erfolgt ausschließlich in WinCC. Die Projektierungsvorschriften für die Ausgabe der Prozessvariablen bleiben erhalten.

#### **WinCC-Projektierung**

Zusätzlich zur Projektierung für die Prozessvariablen muss eine ALARM-Logging-Projektierung erstellt werden. Auf die WinCC-Projektierung wird an dieser Stelle nicht eingegangen. Bitte lesen Sie hierfür die umfangreichen WinCC-Projektierungshandbücher. Für den Einstieg in eine erste WinCC-Projektierung mit Meldungsprojektierung wird das Handbuch SIMATIC WinCC Getting Started empfohlen.

### 3.17.4 SIMATIC TDC-Meldungen

#### 3.17.4.1 SIMATIC TDC-Projektierung

Für die Meldungsangaben von SIMATIC TDC nach WinCC wird zusätzlich zur Projektierung der Prozesswertausgabe der WinCC-Baustein MM benötigt :

MM - Meldungsmanager Die Bausteine werden wie folgt verdrahtet :

(es werden nur die relevanten Anschlüsse beschrieben)

**FB MM**

Anschlussname	Bedeutung	Beispiel
<b>CTS</b>	Prozessorname	D01P01
<b>AR</b>	Kanalname (identisch mit AT-Anschluss des MSI -Bausteins)	EMPFKANA
<b>NZ</b>	Anzahl der Zyklen pro Übertragung	5
<b>NL</b>	Anzahl der angeschlossenen LI-Bausteine	1
<b>MEM</b>	Diagnosetriple	0
<b>TGL</b>	Diagnosetriple	0
<b>CVP</b>	Verbindung mit LI, Anschluss CCV	>LI.CCV

**HINWEIS**

Es muss zusätzlich der Melde-Zentralbaustein @MSI, der Meldeausgabebaustein MSI und der Meldebaustein MERF0 oder ein anderer MERFxx projektiert werden.

**FB @MSI**

Anschlussname	Bedeutung	Beispiel
<b>CMS</b>	Name des Meldesystems	MYMELD
<b>CMT</b>	Meldetext (wird nicht ausgegeben)	“
<b>NOM</b>	Anzahl speicherbarer Meldungen	200
<b>SAV</b>	Meldepuffer um gepufferten RAM	0
<b>RP</b>	Präfix für Kommunikationsfehler	0
<b>MUN</b>	Freigabe für Meldeinträge	1

**FB MSI**

Anschlussname	Bedeutung	Beispiel
<b>CMS</b>	Meldesystemname	MYMELD
<b>CTS</b>	Kopplungsbaugruppenname	D01P01
<b>AT</b>	Adressparameter	EMPFKANA
<b>RP</b>	Präfix für Überlaufmeldungen	0
<b>SNV</b>	Ausgabe Meldungsnummer	1
<b>STM</b>	Ausgabe Meldetext	0
<b>STC</b>	Ausgabe Meldetext konstanter Länge	1
<b>SSF</b>	Ausgabeformat	1
<b>EN</b>	Freigabe	1
<b>MUN</b>	Freigabe für Meldeinträge	1

**FB MERFO**

Anschlussname	Bedeutung	Beispiel
<b>CMS</b>	Meldesystemname	MYMELD
<b>MT</b>	Meldetyp	1
<b>RP</b>	Präfix	0
<b>RS1</b>	Suffix kommende Meldung	10001
<b>RS2</b>	Suffix gehende Meldung	00005
<b>EN</b>	Meldungsfreigabe	1
<b>IS1</b>	Meldetrigger	16#0
<b>SM</b>	Meldung speichern	0

**3.17.4.2 WinCC-Projektierung**

Zusätzlich zur Projektierung für die Prozessvariablen muss eine ALARM - Logging - Projektierung erstellt werden. Auf die WinCC-Projektierung wird an dieser Stelle nicht eingegangen. Bitte lesen Sie hierfür die umfangreichen WinCC-Projektierungshandbücher. Für den Einstieg in eine erste WinCC-Projektierung mit Meldungen wird das Handbuch „SIMATIC WinCC Getting Started“ empfohlen. Die Zuordnung der SIMATIC TDC Meldungsnummern an den Meldebausteinen (RS\*-Anschlüsse) zu denen von WinCC erzeugten Meldungsnummern wird nur durch die „PMC-message no“ ersichtlich, welche aus den Meldungsnummern der Signalliste erzeugt wird.

**3.17.5 Adressbucherzeugung mit CFC-Editor**

Zur Erzeugung der Signalliste für WinCC benötigt ADRIMP die Symbolinformationen der SIMATIC TDC Prozessoren. SIMATIC TDC erzeugt für jede CPU eine ASCII-Datei, welche diese Informationen enthält. Der Dateiname besteht aus dem Baugruppenträgernamen und der CPU-Nummer, getrennt durch „\_“. Als Extension wird „.ADR“ benutzt.

Das Adressbuch wird erzeugt, indem man den gewünschten Plan des Projektes aufruft und die Menüpunkte Extras - Einstellen für Übersetzen... selektiert. Danach die Option Adressbuch erzeugen markieren und mit OK abschließen. Die Menüpunkte Plan und Übersetzen aufrufen. Das Adressbuch wird nun beim Übersetzen erzeugt. Den Pfad des erzeugten Adressbuchs findet man über die Menüpunkte Extras - Protokolle.

**HINWEIS**

Alle Anschlüsse in SIMATIC TDC, auf welche von WinCC zugegriffen werden soll, müssen BuB aktiviert sein (siehe Handbuch „D7-SYS – STEP 7, CFC und SFC projektieren, Kapitel 2.1.6.1).

### 3.17.6 Adresslistenimportwerkzeug ADRIMP

Damit die Adressen der SIMATIC TDC-Pfadnamen von WinCC interpretiert werden können, benötigt man das Adresslistenwerkzeug ADRIMP. Das Adresslistenwerkzeug ADRIMP ermöglicht die Erfassung von Textadresslisten (TALI) in die WinCC-Datenbank. Eine genaue Beschreibung findet man im Benutzerhandbuch TDC PMC TCP Kanal-DLL für WinCC.

#### 3.17.6.1 Voraussetzungen

Es muss eine Variablendefinitionsdatei existieren und das SIMATIC TDC-Adressbuch muss vorher erzeugt werden. Die Variablendefinitionsdatei und das Adressbuch müssen im selben Pfad liegen. Der Generierpfad kann unterschiedlich sein, sollte aber wegen der besseren Übersichtlichkeit ebenfalls in diesem Pfad generiert werden.

##### 3.17.6.1.1 Erzeugen der Variablendefinitionsdatei

Die Variablendefinitionsdatei ist ein Textfile und muss vom Anwender erzeugt werden. Die Variablendefinitionsdatei besteht aus zwei syntaktisch festgelegten Kopfzeilen (1. und 2.), gefolgt von der Zuordnung von symbolischen Namen zu den SIMATIC TDC-Anschlusspfadnamen. Die symbolischen Namen sind frei wählbar, sollten aber wegen der Übersichtlichkeit dieselben sein, die man in den WinCC-Textfeldern verwendet.

##### **Auszug aus der Variablendefinitionsdatei**

z.B.: winccvar.txt

- 1.) [VDM:wincc]
- 2.) [PN:A000\_1,C:\wincc\vardatei]
- 3.) MOTOR\_EIN,ANBIND.CI.CCV  
MOTOR\_AUS,ANBIND.CI.YTS

##### 3.17.6.1.2 Erzeugen und Importieren einer neuen Signalliste

**Voraussetzungen** Vor dem Erzeugen und Importieren der Signalliste muss in der WinCC-Projektierung der TDC PMC TCP-Treiber installiert werden.

- WinCC Projektierung aufrufen
- Variablenhaushalt anklicken
- Menüpunkt „Neuen Treiber hinzufügen“ anklicken
- TDC PMC TCP Ethernet.chn auswählen



Wird vor dem Start von ADRIMP keine WinCC-Projektierung gestartet, wird beim Importieren der Signalliste immer die zuletzt benutzte Projektierung verwendet.

**Durchführung**

- ADRIMP aufrufen
- Menüpunkt „File“ selektieren
- Menüpunkt „Probi“ selektieren
- Variablendefinitionsdatei aussuchen ( z.Bsp.: winvar.txt)
- Generierpfad festlegen
- Signalliste erzeugen (z.Bsp.: wincc.txt)
- Probi beenden

**Hinweis**

**Die Signalliste wird von ADRIMP automatisch in den Variablenhaushalt der entsprechenden WinCC Datenhaltung importiert.**

**3.17.6.1.3 Importieren einer vorhandenen Signalliste**

- WinCC mit gewünschtem Projekt starten
- ADRIMP aufrufen
- Menüpunkt „File“ selektieren
- Menüpunkt „Open“ selektieren
- Signalliste auswählen (z.Bsp.: wincc.txt)
- ADRIMP beenden

**HINWEIS**

---

**Die Signalliste wird von ADRIMP automatisch in den Variablenhaushalt der entsprechenden WinCC Datenhaltung importiert.**

---

**3.17.6.2 Überprüfen des erzeugten Datenhaushaltes in WinCC**

Überprüfen der importierten Daten, deren symbolischer Namen, Datenformate und Pfadnamen:

- WinCC Projektierung aufrufen
- Variablenhaushalt selektieren
- Logische Verbindung (entspricht dem VDM-Namen) anklicken

- TDC PMC TCP selektieren
- logischen Verbindungsnamen selektieren

Es erscheinen die in der Variablendatei definierten logischen Namen und SIMATIC TDC-Pfadnamen. Zusätzlich werden die Datenformate angezeigt. Auf diese Variablen kann nun mit WinCC zugegriffen werden.

### **3.17.7 Verbindungsaufbau SIMATIC TDC-WinCC**

#### **3.17.7.1 WinCC aktivieren**

Um die Verbindung zwischen SIMATIC TDC und WinCC aufzubauen, müssen die importierten Daten der WinCC-Datenbank den Ein/Ausgabefeldern der graphischen Projektierung zugewiesen werden. Dies geschieht durch selektieren der entsprechenden Felder im Graphics Designer und anklicken des Konfigurationsdialogs. Jedem Feld kann eine der importierten Variablen zugewiesen werden. Nach Durchführung der Zuweisungen wird im Hauptmenü der Menüpunkt Datei selektiert und die Daten werden gespeichert. Vor dem Start von Runtime müssen die Verbindungseigenschaften eingestellt werden.

##### **Im Control Center**

- Variablenhaushalt anklicken
- TDC PMC TCP anklicken
- mit rechter Maustaste TCP/IP Unit 1 anklicken
- Eigenschaften anklicken
- Eigenschaften Kanal Unit anklicken (PMC-Parameter projektieren)  
  
Verbindung anklicken TCP/IP Adressen (Local/Remote) eintragen  
Port-Nr. (Local/Remote)
- mit OK bestätigen
- Im Control Center „Datei“ selektieren
- Aktiveren anklicken

WinCC ist nun bereit Daten zwischen SIMATIC TDC und WinCC auszutauschen

#### **3.17.7.2 SIMATIC TDC aktivieren**

Projektierten Baugruppenrahmen eingeschalten. Nach dem Hochlauf steht die Verbindung zwischen SIMATIC TDC und WinCC. Die Daten werden nun zyklisch zwischen SIMATIC TDC und WinCC ausgetauscht.

## 3.18 Kommunikations-Dienst Trace

Für Tracewert-Aufzeichnungen gibt es drei unterschiedliche Dienste:

- **Einfach-Trace**  
er besitzt nur eine Ausgabeschnittstelle.
- **Anlagen-Trace**  
er besitzt eine interaktive Schnittstelle (Auftrags- und Antwortschnittstelle). Um den Anlagen-Trace einsetzen zu können, muss der Anwender ein *eigenes* Tool entwickeln, das die Auftrags-/Antwortsprache des Anlagen-Trace beherrscht. Aus diesem Grund wird der Anlagen-Trace hier auch nicht weiter beschrieben.
- **Analog-Trace**  
er besitzt keine Datenschnittstelle. Dieser Dienst wird allein mit den Funktionsbausteinen TRCC, TRCC\_D und TRCC\_I abgearbeitet. Eine Beschreibung dazu findet sich nicht in der vorliegenden Projektierungsanleitung.

### 3.18.1 Einfach-Trace

Mit Hilfe des Einfach-Trace können Werte von prozessorlokalen Konnektoren aufgezeichnet, gespeichert und ausgegeben werden. Ein Einfach-Trace setzt sich aus einem Steuerbaustein @TCP und einem oder mehreren Erfassungsbausteinen TRP, TRP\_B, TRP\_D und TRP\_I oder TRHI zusammen. Es können beliebig viele Erfassungsbausteine in beliebigen Abtastzeiten projiziert werden.

#### 3.18.1.1 Arbeitsweise des @TCP

Der @TCP-Baustein besitzt im Wesentlichen folgende Aufgaben:

- Steuerung der Tracewert-Erfassungsbausteine TRP, TRP\_B, TRP\_D und TRP\_I
- Auslesen von aufgezeichneten Werten
- Fehlerbehandlung

Diese Aufgaben werden im Folgenden näher erläutert.

#### Steuerung der Tracewert-Erfassung

Die Aufzeichnung und Ausgabe von Tracewerten wird durch die Eingangskonnektoren STA und TBR des @TCP gesteuert.

### Aufzeichnung von Tracewerten

Eine Aufzeichnung durch die Erfassungsbausteine wird folgendermaßen eingeleitet:

- R-Konnektor: 0
- TBR-Konnektor: 0
- STA-Konnektor: Übergang von 0 nach 1

Die Aufzeichnung ist solange aktiv, bis ein Übergang von 1 nach 0 am STA-Konnektor erkannt wird (ein vorzeitiger Abbruch der Tracewert-Aufzeichnung durch ein RESET soll an dieser Stelle nicht betrachtet werden). Während einer aktiven Aufzeichnung befindet sich der @TCP im *Aufzeichnungsmodus*. Die Aufzeichnung von Tracewerten wird beendet, wenn ein Übergang von 1 nach 0 am STA-Konnektor erkannt wird. Ist am Eingangskonnektor TDC der Wert 0 angegeben, so endet die Tracewert-Aufzeichnung unmittelbar. Anderenfalls wartet der @TCP-FB noch um die am TDC angegebene Anzahl von Zyklen, bis er die Tracewert-Aufzeichnung beendet.

### Ausgabe von Tracewerten

Eine Ausgabe von Tracewerten durch den @TCP kann nach dem Beenden einer Tracewert-Aufzeichnung wie folgt eingeleitet werden:

- R-Konnektor: 0
- TBR-Konnektor: Übergang von 0 nach 1
- STA-Konnektor: 0
- OUT-Konnektor: Sammeltelegrammausgabe (nein/ja)
- CID-Konnektor: Falls keine Sammeltelegrammausgabe: Kennzahl des Konnektors, dessen aufgezeichneten Werte ausgegeben werden sollen.

Während einer aktivierten Ausgabe befindet sich der @TCP im sogenannten *Ausgabezustand* (ein vorübergehender Zustandswechsel in einen Wartezustand ist möglich, sofern die Datenschnittstelle temporär nicht ansprechbar ist).

Bei der Ausgabe von Tracewerten gibt es prinzipiell 2 Möglichkeiten:

- Alle Tracewerte von allen Erfassungsbausteinen werden komplett in einem Telegramm übermittelt. Dies ist der Fall, wenn der Konnektor OUT den Wert 1 hat.
- Es wird selektiv über den CID-Konnektor ein Erfassungsbaustein ausgewählt, dessen Tracewerte ausgegeben werden sollen. Die aufgezeichneten Tracewerte werden in Form eines oder mehrerer Antworttelegramme über die Datenschnittstelle ausgegeben. Können die aufgezeichneten Werte für einen Konnektor nicht komplett mit einem Antworttelegramm übertragen werden, werden automatisch weitere (Folge-) Antworttelegramme gesendet. Die logische Reihenfolge der Folge-Telegramme wird durch zwei Zähler innerhalb des Telegramms bekanntgegeben: Zähler *Anzahl Pakete* beschreibt die Anzahl der Pakete, die für die Ausgabe aller Werte für den

betreffenden Konnektor notwendig sind, Zähler Paket-Nr. beschreibt die Nummer des aktuellen Pakets (die zugrundeliegenden Nummerierungskonvention für die beiden Zähler wird im nachfolgenden Kapitel erläutert). Eine Ausgabe endet, wenn entweder alle vorhandenen Tracewerte ausgegeben wurden oder ein Übergang von 1 nach 0 am TBR-Konnektor erfolgt.

<b>Rücksetzen des Einfach-Trace</b>	Der Einfach-Trace kann durch Setzen des Eingangskonnektors R auf 1 zurückgesetzt werden. Durch diese Aktion wird eine eventuell aktivierte Aufzeichnung oder Ausgabe von Tracewerten unmittelbar abgebrochen. Außerdem wird der Tracepuffer neu formatiert, d.h. eventuelle Werte im Puffer gehen verloren. Ein Rücksetzen bleibt solange aktiv, bis der R-Konnektor wieder auf 0 gesetzt wird.
<b>Fehlerbehandlung</b>	Beim Auftreten von Kommunikationsfehlern wird ein Eintrag in Kommunikationsfehlerfeld gemacht und der @TCP abgeschaltet (QTS-Konnektor= 0). Am YTS-Konnektor wird zusätzlich eine entsprechende Kommunikationsfehlernummer gesetzt.
<b>HINWEIS</b>	Die Ursache des Fehlers sowie die Anleitung zur Fehlerbehebung sind dem Dokument „D7-SYS Fehlercodes für schnellen Zugriff“ zu entnehmen.

### 3.18.1.2 Arbeitsweise der Erfassungsbausteine

Die Erfassungsbausteine TRP, TRP\_B, TRP\_D und TRP\_I haben folgende Aufgaben:

- Aufzeichnung der zu erfassenden Eingangsgröße (Trace-Aufzeichnung)
- Fehlerbehandlung

Die Arbeitsweise aller Erfassungsbausteine ist identisch. Sie unterscheiden sich nur hinsichtlich des Formats mit dem die zu erfassende Eingangsgröße aufgezeichnet wird (TRP = Real, TRP\_B = BOOL, TRP\_D = Double Integer, TRP\_I = Integer).

<b>Trace-Aufzeichnung</b>	<p>Jeder Erfassungsbaustein zeichnet die Werte seines Eingangskonnektors auf und speichert sie in einen Umlaufpuffer (Ringpuffer). Jeder Erfassungsbaustein bekommt einen Umlaufpuffer zugewiesen. Die Umlaufpuffer werden zu Beginn der zyklischen Phase eingeteilt. Dabei wird der gesamte Speicher des Tracepuffers, dessen Größe durch den Konnektor TBL des @TCP-FB bestimmt ist, auf die Umlaufpuffer aufgeteilt. Die Umlaufpuffer werden so bestimmt, dass jeder Umlaufpuffer dieselbe Anzahl von Werten aufnehmen kann.</p> <p>Pro Zyklus zeichnet der Erfassungsbaustein genau einen Wert auf, sofern die Tracewerterfassung durch den Zentralbaustein @TCP aktiviert worden ist. Werden Konnektoren durch Erfassungsbausteine aufgezeichnet, die in Alarmebenen (azyklische Aufzeichnung) projiziert sind, wird zusätzlich zu jedem Tracewert der Erfassungszeitpunkt, bestehend aus Datum und Uhrzeit, mit in den Tracepuffer eingetragen.</p>
---------------------------	---

Bei deaktivierter Tracewernerfassung durch den @TCP werden von den Erfassungsbausteinen keine Werte aufgezeichnet. Ist der SSF-Konnektor des @TCP auf 1 gesetzt, werden die aufgezeichnete Tracewerte in standardisierter Form als Floating-Point-Zahlen (Real-Werte) von den Erfassungsbausteinen aufgezeichnet.

**Fehlerbehandlung** Tritt ein Kommunikationsfehler auf, wird ein Eintrag ins Kommunikationsfehlerfeld gemacht und der entsprechende Erfassungsbaustein abgeschaltet (QTS-Konnektor = 0). Am YTS-Konnektor wird zusätzlich die Kommunikationsfehlernummer bekanntgegeben.

---

**HINWEIS** Die Abschaltung eines TRP, TRP\_B, TRP\_D oder TRP\_I - Erfassungsbausteins nach Auftreten eines Kommunikationsfehlers hat keinen Einfluss auf die anderen Bausteine des Einfach-Trace, sofern es noch mindestens einen anderen projektierten funktionsfähigen Erfassungsbaustein gibt. Die Ursache des Fehlers sowie die Anleitung zur Fehlerbehebung sind dem Dokument „D7-SYS Fehlercodes für schnellen Zugriff“ zu entnehmen.

---

### 3.18.1.3 Arbeitweise des Header-Bausteins TRHI

Der Header-Baustein stellt Parameter für ein Sammeltelegramm bereit. Ein Sammeltelegramm wird generiert, wenn der Konnektor OUT des @TCP-FB den Wert 1 hat.

**Der Header-Baustein TRHI hat nur Wirkung auf ein Tracesystem, wenn eine Sammeltelegrammausgabe projektiert ist.**

Ist keine Sammeltelegrammausgabe projektiert, zeigt die Projektierung eines TRHI-FB keine Wirkung.

---

**HINWEIS** Die Projektierung eines TRHI-FB hat nur Auswirkungen auf den Aufbau eines Sammeltelegramms. Siehe dazu Abschnitt „Antworttelegramme“.

---

### 3.18.1.4 Einfach-Trace Projektierung

Der Einfach-Trace kann sowohl einfach (Anzahl @TCP=1) als auch mehrfach (Anzahl @TCP>1) projiziert werden.

#### **Projektierung genau eines Einfach-Trace**

Für genau einen Einfach-Trace sind genau ein Steuerbaustein @TCP und mindestens ein Erfassungsbaustein TRP, TRP\_B, TRP\_D oder TRP\_I zu projektieren. Folgende Projektierungsvorschriften sind verbindlich:

- @TCP und Erfassungsbausteine eines Einfach-Trace besitzen gleiche Kennungen am TRC-Konnektor.
- Der Wert am TBL-Konnektor des @TCP muss größer null sein.
- Alle Erfassungsbausteine besitzen eindeutige, d.h. unterschiedliche Kennzahlen am CID-Konnektor.
- Die Erfassungsbausteine können in unterschiedlichen Abtastzeiten projiziert werden.
- Es muss mindestens ein, können aber beliebig viele Erfassungsbausteine projiziert werden.
- Wird Sammeltelegrammausgabe gewünscht (Konnektor OUT des @TCP-FB = 1), so können maximal 255 Erfassungsbausteine projiziert werden.

**Beispiel:** Belegung der Initialisierungskonnektoren für die Projektierung genau eines Einfach-Trace mit vier Erfassungsbausteinen und Sammeltelegrammausgabe.

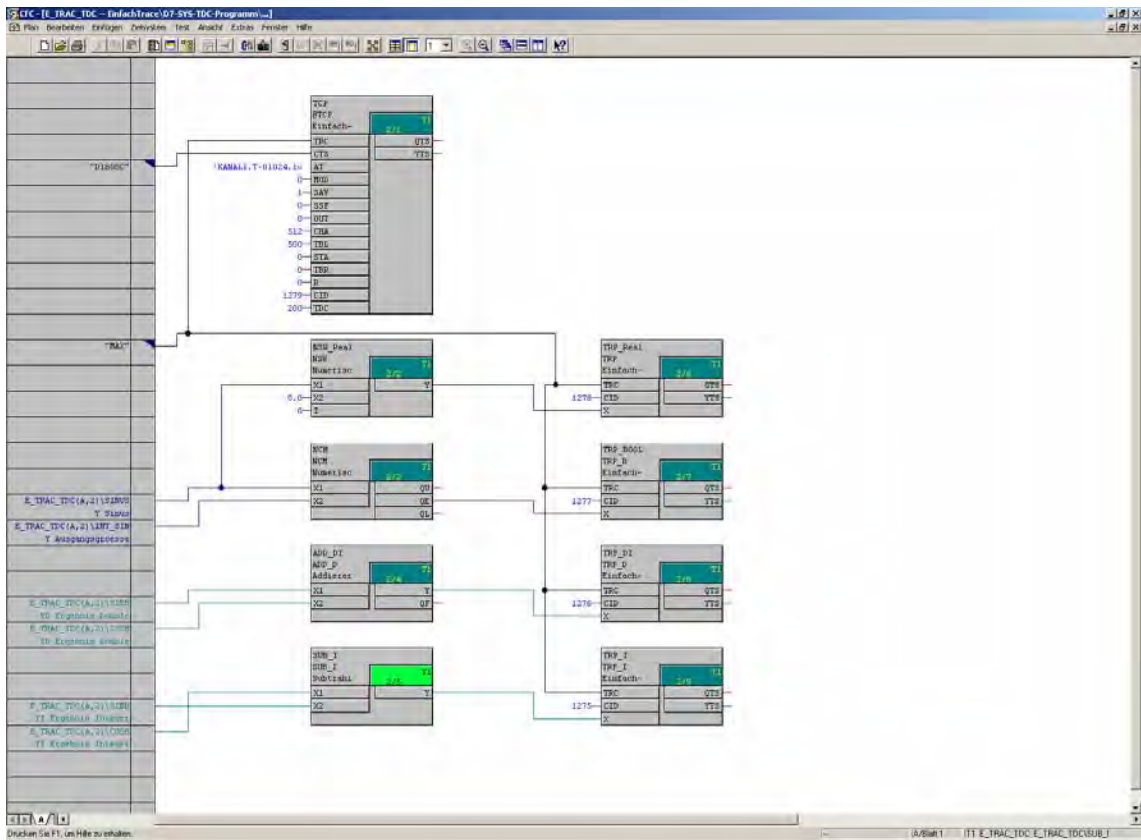


Bild 3-94 Beispiel für die Projektierung eines Einfach-Trace



**Projektierung mehrerer Einfach-Trace**

Neben der Einfach-Projektierung eines Einfach-Trace ist es möglich, mehrere Einfach-Trace parallel zu projektieren.

Ein Einfach-Trace wird dabei genau durch die Bausteine gebildet, die identische Kennungen am TRC-Konnektor besitzen. Neben den oben genannten Projektierungsvorschriften gilt folgendes zusätzlich:

**Alle projektierten @TCP-Bausteine tragen eindeutige, d.h. unterschiedliche TRC-Kennungen.**

**Beispiel:** Belegung der Initialisierungskonnektoren für die Projektierung von zwei Einfach-Trace.

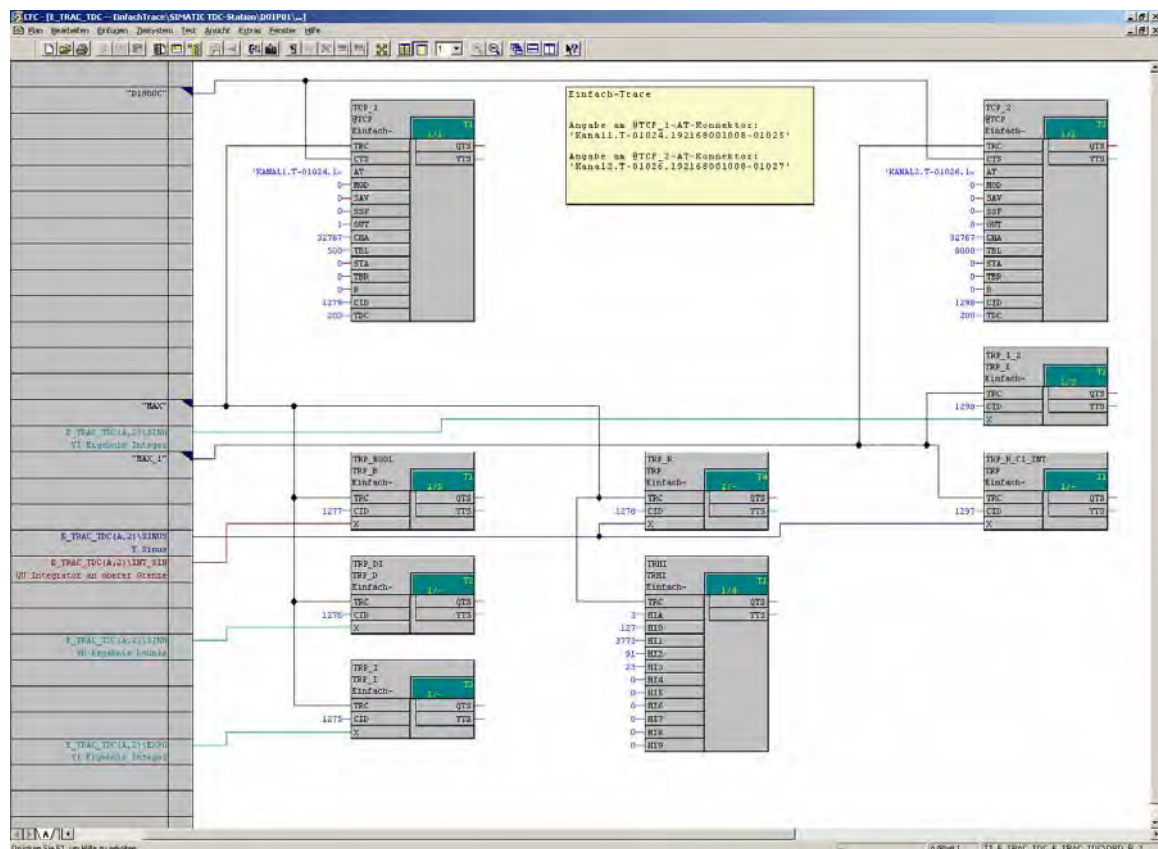


Bild 3-95 Beispiel für die Projektierung von zwei Einfach-Trace

### 3.18.1.5 Antworttelegramme

Das vom @TCP-FB generierte Antworttelegramm ist abhängig vom Wert des Konnektors OUT am @TCP-FB. Hat der Konnektor den Wert 0, so werden Einzeltelegramme generiert, hier liegt das gleiche Ausgabeformat wie beim Anlagen-Trace vor. Bei den Einzeltelegrammen wird selektiv durch den CID-Konnektor des @TCP-FB ein Erfassungsbaustein ausgewählt, dessen Tracewerte in ein oder mehreren Telegrammen ausgegeben werden. Hat der OUT Konnektor den Wert 1, so wird ein Sammeltelegramm generiert. Ein Sammeltelegramm ist dadurch charakterisiert, das es alle Tracewerte aller Erfassungsbausteine enthält. Eine Angabe am CID-Konnektor des @TRP-FB ist in diesem Fall nicht notwendig.

**Einzeltelegramme** Ein Einzeltelegramm wird vom Einfach-Trace generiert, wenn der Initialisierungskonnektor OUT des @TCP 0 enthält. Das Telegramm hat dann folgenden Aufbau:

Byte-Nr.	Inhalt	Datenformat	Bedeutung
1,2	Konnektor-Kennzahl	unsigned int16	
3	Ausgabeformat	char	0=binär, 1=standardisiert
4	Konnektor-Interpretation	char	1 - 12
5 bis 10	letzter Erfassungszeitpunkt	6 chars	
11, 12	Paket-Nr.	unsigned int16	
13, 14	Anzahl-Pakete	unsigned int16	
15	Aufzeichnungsmodus	char	0=zykl., 1=Alarmaufgabe
16	Konnektorformat	char	
17 bis 20	Abtastzeit	unsigned int32	Angabe in 1/10 ms
21, 22	Aufgabenkennung	unsigned int16	
23, 24	Anz. Blöcke im Tracepuffer	unsigned int16	
25, 26	Anz. Blöcke im Telegramm	unsigned int16	
27,28	leer	unsigned int16	

---

**HINWEIS** Telegrammrumpf: ab 29 Tracewert-Blöcke siehe nachfolgende Beschreibung

---

Die **Konnektor-Kennzahl** identifiziert den Erfassungsbaustein, dessen Tracewerte durch das Telegramm ausgegeben werden. Diese Angabe ist mit der Angabe am Eingangskonnektor CID des @TCP-FB identisch.

Das **Ausgabeformat** gibt das Format der Tracewerte in den Tracewert-Blöcken an. Beim binären Format werden die Konnektoren entsprechend ihren Datentypen ausgegeben, im standardisierten Format immer als 4 Bytes großer Fließkommawert.

Die **Konnektorinterpretation** gibt Auskunft über den Typ des aufgezeichneten Konnektors und kann folgende Werte annehmen:

Konstante	Konnektor-Interpretation
1	N = Standard-Interpretation
3	I = ganzzahlige Zahl
8	B = boolescher Wert

Der **letzte Erfassungszeitpunkt** bezieht sich auf den zuletzt aufgezeichneten Tracewert, sofern die Aufzeichnung zyklisch durchgeführt wurde (bei azyklischer Aufzeichnung hat diese Angabe keine Bedeutung). Dieser Tracewert ist der letzte Tracewert bzw. Tracewert-Block im Telegrammrumpf mit der Paket-Nr. 0. Der Erfassungszeitpunkt wird im SIMADYN D-Format *Time and Date* übertragen (Time and Date s. Beschreibung „D7-SYS – SIMADYN D System- und Kommunikationsprojektierung“, Kapitel „Kompatible Nutzdatenstrukturen“).

Die Felder **Paket-Nr.** und **Anzahl-Pakete** nummerieren die Telegramme durch. Jedes Antworttelegramm zu einem (ausgeführten) Ausgabe-Kommando hat zwei Zähler, die die aktuelle Nummer des Pakets und die Gesamtanzahl der notwendigen Pakete zur Übertragung aller aufgezeichneten Werte beinhalten. Die Nummerierungskonvention ist dahingehend festgelegt, dass das letzte Paket die Paket-Nr. 0 trägt. Sind beispielsweise drei Antworttelegramme zur Übertragung der Tracewerte notwendig, ergibt sich folgende Nummerierung:

**Beispiel:**

	1. Antworttelegramm	2. Antworttelegramm	3. Antworttelegramm
<b>Paket-Nr.</b>	2	1	0
<b>Anzahl der Pakete</b>	2	2	2

Durch den **Aufzeichnungsmodus** wird festgehalten, ob der zugehörige Erfassungsbaustein in einer zyklischen Ebene oder einer Alarmebene projiziert wurde.

Das **Konnektorformat** gibt die Größe der erfassten Tracewerte an und kann folgende Werte annehmen:

Konstante	Bedeutung
1	1-Byte (aufgezeichnet durch TRP_B-Baustein)
2	2. Byte (aufgezeichnet durch TRP_I-Baustein)
4	4-Byte (aufgezeichnet durch TRP_D-Baustein)
5	4-Byte standardisiert (aufgezeichnet durch TRP-Baustein)

Die **Abtastzeit** gibt die Abtastzeit des zugehörigen Erfassungsbausteins in 1/10 ms an.

Die **Aufgabenkennung** gibt Auskunft über die Ebene, in welcher der zugehörige Erfassungsbaustein projiziert wurde:

<u>Abtastzeit</u>	<u>Konstante</u>
T1	0
T2	1
T3	2
T4	3
T5	4

<u>Alarmbene</u>	<u>Konstante</u>
I1	5
I2	6
I3	7
I4	8
I5	9

Die **Anzahl Tracewert-Blöcke im Tracepuffer** gibt an, wie viele erfasste Werte sich für diesen Erfassungsbaustein im Tracepuffer befinden.

Die **Anzahl Tracewert-Blöcke im Telegramm** gibt an, wieviele Tracewert-Blöcke im aktuellen Telegramm enthalten sind.

Die **Tracewert-Blöcke** beinhalten die Daten, die für einen Konnektor zu einem bestimmten Zeitpunkt aufgezeichnet worden sind (gegebenenfalls durch Dummy-Bytes ergänzt). Die Struktur eines Tracewert-Blocks ist vom Konnektorformat und dem Aufzeichnungsmodus des Erfassungsbausteins, der diesen Konnektor aufzeichnet, und dem projizierten Ausgabeformat des @TCP abhängig. Alle möglichen Tracewert-Blöcke werden durch die nachfolgende Tabelle beschrieben (alle Zahlenangaben in Bytes).

Aufzeichnungsmodus	Ausgabeformat	Konnektorformat	Konnektorzwert	Dummy-Bytes	Zeitpkt. der Erfassung	Dummy-Bytes	Blocklänge
zyklisch	0	1	1	-	-	-	1
zyklisch	0	2	2	-	-	-	2
zyklisch	0	4	4	-	-	-	4
zyklisch	0	5	4	-	-	-	4
azyklisch*	0	1	1	1	6	-	8
azyklisch*	0	2	2	-	6	-	8
azyklisch*	0	4	4	-	6	2	12
azyklisch*	0	5	4	-	6	2	12
zyklisch	0	1,2,4,5	4	-	-	-	4
azyklisch*	0	1,2,4,5	4	-	6	2	12

Ein Tracewert-Block setzt sich aus dem Konnektorwert, einem Dummy-Byte (optional), dem Zeitpunkt der Erfassung (nur bei azyklischer Erfassung, wird im SIMADYN D-Format *Time and Date* ausgegeben) und weiteren Dummy-Bytes zusammen (optional). Das Ausgabeformat ist durch SSF-Konnektor des @TCP-Bausteins bestimmt. Das Konnektorformat durch den Typ des Erfassungsbausteins.

\* Der Konnektorwert wurde von einem Erfassungsbaustein in einer Alarmebene (I1 ... I5) aufgezeichnet.

#### **Beispiel:**

Strukturierung der Werte im Telegramm-Rumpf für einen 1 Byte-Konnektor, der durch einen Erfassungsbaustein azyklisch (Alarmebene) aufgezeichnet wurde. Die Ausgabe erfolgt in standardisierter Form als Fließkommazahl.

Byte	Bedeutung
29 ... 32	Tracewert zum Zeitpunkt t1
33 ... 38	Erfassungszeitpunkt t1
39, 40	Dummies
41 ... 44	Tracewert zum Zeitpunkt t2
45 ... 50	Erfassungszeitpunkt t2
51, 52	Dummies
...	...

#### **Sammeltelegramme**

Ein Sammeltelegramm wird vom @TCP-FB generiert, wenn am Initialisierungskonnektor OUT des @TCP-FB 1 anliegt.

Ein Sammeltelegramm besteht prinzipiell nur aus **einem** Telegramm. Dies hat zur Folge, dass der Ausgabekanal entsprechend dimensioniert sein muss, um alle Tracewerte auf einmal aufnehmen zu können (CHA-Konnektor des @TCP-FB). Ist der Kanal zu klein dimensioniert, geht der @TCP-FB zu Beginn der zyklischen Phase in den Zustand AUS. Dabei generiert er eine Kommunikationsfehlermeldung, wobei der Zusatz der Fehlermeldung die Mindestgröße des Kanals angibt. Die Mindestgröße des Ausgabekanal lässt sich nach folgender Formel berechnen:

$$CHA \geq 12 + 4 * HIA + 12 * TRPx + 4 * TBL$$

Dabei bedeuten:

- CHA -Mindestkanalgröße in Bytes ( Angabe am Konnektor CHA des @TCP )
- HIA -Angabe am Konnektor HIA des TRHI-FB ( 0 falls TRHI-FB nicht projektiert )
- TRPx -Anzahl projektiertes ( und korrekt initialisierter ) Erfassungsbausteine
- TBL -Größe des Tracepuffers ( Angabe am Konnektor TBL des @TCP )

Der Aufbau eines Sammeltelegramms gestaltet sich wie folgt:

Telegramm-Header:

Byte-Nr.	Inhalt	Datenformat	Bedeutung
1	Anzahl aktiver TRPx	char	
2	Ausgabeformat	char	siehe Einzeltelegramm
3 bis 8	letzter Erfassungszeitpunkt	6 chars	
9 bis 12	Parameter 0	unsigned int32	erster Parameter TRHI-FB
...			
$9 + (n-1) * 4$	Parameter n-1	unsigned int32	n-ter Parameter TRHI-FB

Telegrammrumpf:

Byte-Nr.	Inhalt	Datenformat	Bedeutung
$9 + n * 4$	Konnektorkennzahl	unsigned int16	siehe Einzeltelegramm
$11 + n * 4$	Konnektorinterpretation	char	siehe Einzeltelegramm
$12 + n * 4$	Konnektorformat	char	siehe Einzeltelegramm
$13 + n * 4$	Abtastzeit	unsigned int32	siehe Einzeltelegramm
$17 + n * 4$	Aufgabenkennung	unsigned int16	siehe Einzeltelegramm
$19 + n * 4$	Anzahl Tracewert-Blöcke	unsigned int16	siehe Einzeltelegramm
$21 + n * 4$	Tracewert-Blöcke		siehe Einzeltelegramm

Für jeden Erfassungsbaustein ist ein Telegrammrumpf vorhanden. Dabei sind die Telegrammrümpfe nach der Anmeldeihenfolge der zugehörigen Erfassungsbausteine geordnet.

Die Bedeutung der Daten innerhalb der Sammeltelegramme sind weitgehend mit der Bedeutung bei Einzeltelegrammen identisch.

Die **Anzahl der aktiven TRPx** entspricht der Anzahl der Telegrammrümpfe im Sammeltelegramm. Eine Zuordnung der Telegrammrümpfe zu den Erfassungsbausteinen ist dann über die Konnektorkennzahl, die mit der CID-Angabe der Erfassungsbausteine übereinstimmt, möglich.

Beim **letzten Erfassungszeitpunkt** im Telegramm-Header wird beim Sammeltelegramm der Zeitpunkt der Erkennung des Stop-Signals eingetragen. Bei Einzeltelegrammen entspricht der letzte Erfassungszeitpunkt dem Zeitpunkt der Aufnahme des letzten Tracewertes in den Tracewertpuffer und kann somit bei Erfassungsbausteinen differieren.

Die **Parameter 1- n** des TRHI-FB werden nur eingetragen, falls dieser auch projektiert ist. Wie viele Parameter in den Header eingetragen werden, ist durch den Konnektor HIA des TRHI-FB erkennbar.

Achtung: **Im Sammeltelegramm ist keine Angabe vorhanden, ob und wie viele Parameter des TRHI-FB übernommen wurden.** Die Interpretation des Sammeltelegramms ist somit von der Projektierung abhängig. Die Parameter stellen Zusatzinformationen dar, deren Anzahl und Interpretation vom Anwender frei wählbar ist.

**Jeder Telegrammrumpf beginnt auf einem durch 4 teilbaren Offset bzgl. des Telegrammanfangs. Deswegen werden Telegrammrümpfe u.U. mit Dummybytes aufgefüllt.**

Beispiel: Es wird ein Sammeltelegramm generiert, in dem drei Tracewerte eines TRP\_B-FB vorhanden sind. Der TRP\_B-FB wird im zyklischen Aufzeichnungsmodus und mit binärem Ausgabeformat betrieben, so dass der zugehörige Tracewert-Block die Größe von einem Byte hat. Das Feld *Anzahl Tracewert-Blöcke* im Telegrammrumpf hat dann den Wert 3, dahinter sind drei Bytes Tracewert-Blöcke im Telegramm vorhanden. Nun wird ein Dummybyte eingefügt, dahinter beginnt der nächste Telegrammrumpf. Hinter dem letzten Telegrammrumpf werden keine Dummybytes eingefügt.





# Index

## \$

\$-Signale .....	2-19
------------------	------

## A

Alarmgesteuerte Bearbeitung.....	2-37
Applikationsbeispiel PROFIBUS DP .....	3-86
Konfiguration Slaves .....	3-102
Aufgabenverwalter .....	2-20, 2-32

## B

Baugruppenträger-Kopplung CP52M0	
Anwendungsgebiete.....	3-24
Ein- und Ausschaltverhalten .....	3-25
Leistungsdaten .....	3-29
Projektierung .....	3-26
Baugruppenträger-Kopplung CP53M0	
Abschaltung eines Koppelpartners .....	3-34
Allgemeines.....	3-30
Einschränkungen.....	3-37
Hardware-Aufbau .....	3-33
Initialisierung und Überwachung .....	3-30
Leistungsumfang .....	3-33
Projektierung .....	3-36
Quittierung.....	3-34
Wiederanlauffähigkeit.....	3-36
Zuschaltung eines Koppelpartners.....	3-34
Baugruppenträgerlokale Kopplungen.....	3-21
Betriebssystemkomponenten .....	2-36
Bibliotheken .....	2-4

## C

CFC-Editor.....	2-7
Parametrierdialoge .....	2-8
Randleisten .....	2-12
CFC-Plan (Continuous Function Chart) .....	2-7
COM PROFIBUS.....	3-80
CPU-Grundtakt .....	2-20
CPU-lokale Kopplung .....	3-21
CPU-Synchronisation .....	2-26
Projektierung der Alarmtask.....	2-29
Projektierung des CPU-Grundtaktes.....	2-28
Reaktion bei Ausfall.....	2-28
Uhrzeitsynchronisation .....	2-27

**D**

Datenkonsistenz .....	2-17, 2-22
Dienst Service	
Funktionsbaustein SER .....	2-39
Systembelastung, Antwortzeiten .....	2-40
Dienstprogramme .....	2-39
Direkte CPU-CPU-Kopplung .....	3-22
Download im Betriebszustand RUN .....	2-16

**F**

Fast-\$-Signal .....	2-20
Fehlerdiagnose	
Hintergrund-Verarbeitung .....	2-45
Fehlerdifferenzierung .....	2-38
Funktionsbaustein .....	2-8
Auswählen .....	2-8
Kommentare .....	2-13
Parametrieren und Verschalten .....	2-8
Zuordnung zu Alarm-Tasks .....	2-10
Zuordnung zu zyklischen Tasks .....	2-10
Funktionsbausteine Diagnose .....	3-64

**G**

Grenzzahl von Verschaltungen .....	2-21
Grundlagen Kommunikation .....	3-2
Grundtakt .....	2-27

**H**

Hardwareadresse .....	2-13
Hardwarebaugruppen .....	2-5
Auswählen .....	2-5
Parametrieren .....	2-6
Hardware-Timer .....	2-28
HWKonfig .....	2-6
Parametrierdialog .....	2-6

**I**

Initialisierung .....	2-36
-----------------------	------

**K**

Kommandos SYNC/FREEZE .....	3-58
Kommunikation	
SIMATIC Operator Panels .....	3-183
WinCC über SINEC H1 .....	3-225
Kommunikationsbausteine	
Adreßanschlüsse AT, AR, US .....	3-8
Anschluß MOD .....	3-10
Firmwarestatus Anschluß ECL, ECO .....	3-14
Initialisierungsanschluß CTS .....	3-7

Kopplungs-Zentralbausteine .....	3-16
Sender und Empfänger .....	3-17
Zustandsanzeige Ausgang YTS.....	3-14
Kommunikations-Dienste	
Übersicht .....	3-7
Konsistenzprüfung.....	2-7
Koppelspeicher-Kopplung .....	3-23
Kopplung PROFIBUS DP	
Angaben am Adressanschluss AT, AR .....	3-57
Eigenschaften.....	3-55
Funktionsbausteine .....	3-56
Kommunikations-Dienst .....	3-56
Kopplungs-Zentralbaustein .....	3-56
Übertragungsmodus.....	3-56
Kopplungen	
Datenschnittstelle .....	3-20
Funktionsweise.....	3-15
Nutzdatenstrukturen.....	3-18
Übersicht .....	3-2
Kopplungsbaugruppen	
Anzahl im Baugruppenträger .....	3-19
 <b>L</b>	
Laden des Anwenderprogramms .....	2-14
Offline-Laden.....	2-15
Online-Laden.....	2-15
Leistungsmerkmale .....	2-34
Alarmtasks.....	2-35
Rechenzeiten des Betriebssystems .....	2-35
Speicherbedarf des Betriebssystems.....	2-35
zyklische Tasks .....	2-34
 <b>M</b>	
Meldesystem .....	3-144
Ausgabeformat .....	3-157
Eintragslogik.....	3-144
Fehler- oder Warnmeldung .....	3-149
Kommunikationsfehlermeldung.....	3-151
Meldeeintragsbausteine .....	3-144
Meldetypbeschreibung .....	3-149
Meldungen.....	3-146, 3-149
Meldungsformate.....	3-149, 3-152
Systemfehler .....	3-152
Systemfehlermeldung.....	3-152
Überlaufmeldung.....	3-151
MPI-Kopplung.....	3-105
Projektierung .....	3-105
 <b>N</b>	
Namensvergabe .....	2-3
Normalmode.....	2-24, 2-25

**O**

Operator Panels (SIMATIC) ..... 3-183

**P**

PROFIBUS DP ..... 3-68

- Adressanschluß ..... 3-57, 3-70
- Applikationsbeispiel ..... 3-95
- COM PROFIBUS ..... 3-95
- COM-Datenbasis ..... 3-95
- Diagnose ..... 3-76
- Download COM-Datenbasis ..... 3-103
- Error-Class ..... 3-67, 3-84
- Hard- und Software ..... 3-89
- Kommunikationsmodul SS52 ..... 3-95
- Leuchtdiode ..... 3-83
- Musterkonfiguration ..... 3-86
- Parametrierung ..... 3-95, 3-99
- Projektierung ..... 3-56, 3-69, 3-92
- Projektierung CFC ..... 3-89
- Sende- und Empfangsbausteine ..... 3-91
- SIEMENS DP-Slaves ..... 3-65, 3-78
- Speicher SS52 ..... 3-103
- SYNC/FREEZE ..... 3-58, 3-72

Prozeßabbild ..... 2-22

- Alarmtasks ..... 2-26
- Realisierung ..... 2-23
- zyklische Tasks ..... 2-24

Prozeßdaten ..... 3-159

- Bausteine CRV, CTV ..... 3-163
- Beispiel-Projektierung ..... 3-161
- Diagnose ..... 3-166
- Funktionsbausteine ..... 3-159
- Kanäle ..... 3-167
- Kanal-Rangierbausteine ..... 3-163
- Verteilerbaustein ..... 3-165
- virtuelle Verbindungen ..... 3-159

Pseudokommentare ..... 2-13

**S**

Service ..... 3-178

- Funktionsbaustein SER ..... 3-179
- Systembelastung ..... 3-180

Siebensegmentanzeige ..... 2-47

- Fehler quittieren ..... 2-47

Signaltransport ..... 2-16

- Task ..... 2-17

SIMADYN D

- Station konfigurieren ..... 2-4

SIMATIC Operator Panel

- Bausteinanschlüsse ..... 3-186
- Betriebsmeldung ..... 3-187
- Funktionstastatur ..... 3-188

Initialisierung .....	3-185
Konfiguration HWKonfig .....	3-184
Musterkonfiguration .....	3-184
Projektierung CFC .....	3-185
ProTool/Lite-Projektierung .....	3-189
Rechenzeiten Funktionsbausteine .....	3-190
Schnittstellenbereich .....	3-188
Störmeldung .....	3-187
Symboltabelle .....	3-189
Voraussetzung .....	3-183
Steckplatznummer .....	2-6
Symboltabelle .....	3-189
SYNC/FREEZE-Projektierungsvarianten .....	3-59
Synchronisationskonstellation .....	2-30
Systemmode .....	2-23
Systemplan .....	2-7
Systemzustand Anwenderstop .....	2-16
<b>T</b>	
Tabellenfunktion .....	3-106
Taktsynchronität und Äqidistanz .....	3-58
Taskbearbeitung .....	2-21, 2-23
Totzeiten .....	2-20
<b>U</b>	
Übertragungsmodus	
Handshake .....	3-10
Image .....	3-13
Multiple .....	3-12
Refresh .....	3-11
Select .....	3-12
Übersicht .....	3-10
Uhrzeitsynchronisation .....	3-181
<b>V</b>	
Verhalten im Störfall .....	2-41
Verschalten .....	2-13
Virtuelle Verbindungen .....	3-159
<b>W</b>	
WinCC- Anbindung .....	3-191
Kopplung über TCP/IP mit BuB- Funktionen .....	3-192
Kopplungsvarianten MPI und PROFIBUS DP .....	3-209
Projektierung mit dem D7-SYS-OS-Engineering-Tool .....	3-217
Projektierungsvariante S7DB .....	3-207
WinCC über SINEC H1 .....	3-225
<b>Z</b>	
Zeiger-basierte Kommunikationsbausteine	
Anwendungen .....	3-169

Beispiele .....	3-173
Einführung .....	3-168
Merkmale.....	3-169
Prinzipielle Funktionsweise .....	3-169
Projektierungshinweise .....	3-171
Zeiger-Schnittstelle.....	3-171
Zugehörige Funktionsbausteine .....	3-170
Zyklusfehler .....	2-32
Beseitigung.....	2-33