

SIMATIC Programmieren mit STEP 7 Lite V3.0

Handbuch

Vorwort, Inhaltsverzeichnis	
Produkteinführung und Installation	1
Grundlagen zum Entwerfen eines Programms	2
Starten und Bedienen	3
Einrichten und Bearbeiten des Projekts	4
Konfigurieren der Hardware	5
Programmieren von Bausteinen	6
Aufbau der Online-Verbindung und CPU-Einstellung	7
Importieren, Exportieren, Speichern unter	8
Laden in CPU und in PG	9
Testen	10
Diagnose	11
Drucken einer Projektdokumentation	12
Tipps und Tricks	13
Anhänge	

Index

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährungsgrad folgendermaßen dargestellt:



Gefahr

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



Warnung

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



Vorsicht

bedeutet, dass eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht

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

Achtung

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, 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

SIMATIC®, SIMATIC HMI® und SIMATIC NET® sind Marken der Siemens AG.

Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Copyright © Siemens AG 2004 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

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, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

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

© Siemens AG 2004
Technische Änderungen bleiben vorbehalten

Vorwort

Zweck des Handbuchs

Dieses Handbuch gibt Ihnen einen vollständigen Überblick über das Programmieren mit STEP 7 Lite. Es unterstützt Sie bei der Installation und Inbetriebnahme der Software. Die Vorgehensweise bei der Programmerstellung, den Aufbau von Anwenderprogrammen und die einzelnen Sprachelemente werden erläutert.

Es richtet sich an Personen, die für die Realisierung von Steuerungsaufgaben mit STEP 7 Lite auf Basis der SIMATIC Automatisierungssysteme tätig sind.

Wir empfehlen Ihnen, sich mit den Beispielen aus dem Handbuch "Erste Schritte mit STEP 7 Lite" vertraut zu machen. Das Handbuch bietet einen leichten Einstieg in die Programmierung mit STEP 7 Lite.

Erforderliche Grundkenntnisse

Zum Verständnis des Handbuchs sind allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik erforderlich.

Außerdem werden Kenntnisse über die Verwendung von Computern oder PC-ähnlichen Arbeitsmitteln (z. B. Programmiergeräten) unter den Betriebssystemen MS Windows 2000 Professional, MS Windows XP Home und MS Windows XP Professional vorausgesetzt.

Gültigkeitsbereich des Handbuchs

Das Handbuch ist gültig für das Softwarepaket STEP 7 Lite V3.0.

Einordnung in die Informationslandschaft

Dieses Handbuch ist Bestandteil des Softwarepaketes STEP 7 Lite.

Die folgende Tabelle zeigt die Dokumentation zu STEP 7 Lite im Überblick:

Handbücher	Zweck	Bestellnummer
Programmieren mit STEP 7 Lite	Vermittelt das Hintergrundwissen zur Realisierung von Steuerungsaufgaben mit STEP 7 Lite.	Bestandteil des Softwarepaketes STEP 7 Lite
Erste Schritte mit STEP 7 Lite	Beschreibt die wichtigsten Bedienabläufe anhand von praktischen Übungen.	Bestandteil des Softwarepaketes STEP 7 Lite

Online-Hilfen	Zweck	Bestellnummer
Hilfe zu STEP 7 Lite	Vermittelt das Hintergrundwissen zur Realisierung von Steuerungsaufgaben mit STEP 7 Lite.	Bestandteil des Softwarepaketes STEP 7 Lite
Referenzhilfen zu KOP, FUP, AWL Referenzhilfen zu Bausteinbibliotheken	Kontextsensitives Referenzwissen	Bestandteil des Softwarepaketes STEP 7 Lite
Kurzhilfe und Direkthilfe	Bieten Information zum aktuellen Kontext, z. B. zu Menübefehlen, Oberflächenelementen und Dialogfeldern	Bestandteil des Softwarepaketes STEP 7 Lite

Weitere Unterstützung

Bei Fragen zur Nutzung der im Handbuch beschriebenen Produkte, die Sie hier nicht beantwortet finden, wenden Sie sich bitte an Ihren Siemens-Ansprechpartner in den für Sie zuständigen Vertretungen und Geschäftsstellen.

<http://www.ad.siemens.de/partner>

Den Wegweiser zum Angebot an technischen Dokumentationen für die einzelnen SIMATIC Produkte und Systeme finden Sie unter:

<http://www.siemens.de/simatic-tech-doku-portal>

Trainingscenter

Um Ihnen den Einstieg in das Automatisierungssystem SIMATIC S7 zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in D 90327 Nürnberg.
Telefon: +49 (911) 895-3200.

<http://www.sitrain.com>

Automation and Drives, Service & Support

Weltweit erreichbar zu jeder Tageszeit:



<p>Weltweit (Nürnberg) Technical Support</p> <p>Ortszeit: 0:00 bis 24:00 / 365 Tage Telefon: +49 (180) 5050-222 Fax: +49 (180) 5050-223 mailto:adsupport@siemens.com GMT: +1:00</p>		
<p>Europa / Afrika (Nürnberg) Authorization</p> <p>Ortszeit: Mo.-Fr. 8:00 bis 17:00 Telefon: +49 (180) 5050-222 Fax: +49 (180) 5050-223 mailto:adsupport@siemens.com GMT: +1:00</p>	<p>United States (Johnson City) Technical Support and Authorization</p> <p>Ortszeit: Mo.-Fr. 8:00 bis 17:00 Telefon: +1 (423) 262 2522 Fax: +1 (423) 262 2289 mailto:simatic.hotline@sea.siemens.com GMT: -5:00</p>	<p>Asien / Australien (Peking) Technical Support and Authorization</p> <p>Ortszeit: Mo.-Fr. 8:00 bis 17:00 Telefon: +86 10 64 75 75 75 Fax: +86 10 64 74 74 74 mailto:adsupport.asia@siemens.com GMT: +8:00</p>
<p>Technical Support und Authorization sprechen generell Deutsch und Englisch.</p>		

Service & Support im Internet

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet unser komplettes Wissen online an.

<http://www.siemens.com/automation/service&support>

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.
- Informationen über Vor-Ort Service, Reparaturen, Ersatzteile. Vieles mehr steht für Sie unter dem Begriff "Leistungen" bereit.

Inhaltsverzeichnis

1	Produkteinführung und Installation	1-1
1.1	Leitfaden durch STEP 7 Lite	1-1
1.2	Projektfenster und Sichten in STEP 7 Lite.....	1-5
1.3	Hilfe und Dokumentation zu STEP 7 Lite.....	1-10
1.4	Installation	1-11
1.4.1	Automation License Manager	1-11
1.4.1.1	Nutzungsberechtigung durch den Automation License Manager.....	1-11
1.4.1.2	Installieren des Automation License Managers	1-13
1.4.1.3	Regeln für den Umgang mit License Keys.....	1-15
1.4.2	Installieren von STEP 7 Lite.....	1-16
1.4.2.1	Vorgehen beim Installieren	1-17
1.4.2.2	Einstellen der PG/PC-Schnittstelle	1-19
1.4.3	Deinstallieren von STEP 7 Lite	1-21
2	Grundlagen zum Entwerfen eines Programms	2-1
2.1	Programme in einer CPU	2-1
2.2	Bausteine im Anwenderprogramm.....	2-2
2.2.1	Organisationsbausteine und Programmstruktur	2-3
2.2.2	Aufrufhierarchie im Anwenderprogramm	2-8
2.2.3	Zyklische Programmbearbeitung und CPU-Einstellungen.....	2-10
2.2.3.1	Organisationsbaustein für zyklische Programmbearbeitung (OB 1).....	2-10
2.2.3.2	Kommunikationslast	2-13
2.2.4	Alarmgesteuerte Programmbearbeitung.....	2-14
2.2.4.1	Organisationsbausteine für alarmgesteuerte Programmbearbeitung	2-14
2.2.4.2	Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17).....	2-15
2.2.4.3	Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23).....	2-18
2.2.4.4	Organisationsbausteine für den Weckalarm (OB 30 bis OB 38)	2-19
2.2.4.5	Organisationsbausteine für den Prozessalarm (OB 40 bis OB 47)	2-20
2.2.4.6	Organisationsbausteine für den Anlauf (OB 100/OB 102).....	2-21
2.2.4.7	Organisationsbaustein für die Programmbearbeitung im Hintergrund (OB 90).....	2-23
2.2.4.8	Organisationsbausteine für die Fehlerbearbeitung (OB 80 bis OB 87 / OB 121 bis OB 122)	2-24
2.2.5	Bausteinarten für die strukturierte Programmierung.....	2-26
2.2.5.1	Funktionen (FC)	2-26
2.2.5.2	Funktionsbausteine (FB)	2-26
2.2.5.3	Instanz-Datenbausteine	2-29
2.2.6	Globale Datenbausteine (DB)	2-31
2.2.6.1	Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC)	2-32

3	Starten und Bedienen	3-1
3.1	Starten von STEP 7 Lite.....	3-1
3.2	Aufrufen der Hilfefunktionen.....	3-2
3.3	Benutzungsoberfläche und Bedienung.....	3-3
3.3.1	Aufbau der Benutzungsoberfläche.....	3-3
3.3.2	Symbole im Projektfenster.....	3-5
3.3.3	Elemente in Fenstern und Dialogfeldern.....	3-6
3.3.4	Sitzungsgedächtnis.....	3-9
3.3.5	Ändern der Fensteranordnung.....	3-9
3.3.6	Speichern und Wiederherstellen der Fensteranordnung.....	3-10
3.3.7	Suchen bzw. Ersetzen von Begriffen.....	3-11
3.3.8	Schritte zum Manipulieren von Objekten.....	3-13
3.3.8.1	Umbenennen von Objekten.....	3-13
3.3.8.2	Verschieben von Objekten.....	3-13
3.3.8.3	Löschen von Objekten.....	3-13
3.4	Bedienung über Tastatureingabe.....	3-14
3.4.1	Tastenkombinationen für Menübefehle.....	3-14
3.4.2	Tastenkombinationen für das Bewegen des Mauszeigers.....	3-17
3.4.3	Tastenkombinationen für das Markieren von Texten.....	3-18
3.4.4	Tastenkombinationen für den Zugriff auf die Online-Hilfe.....	3-18
3.4.5	Tastenkombinationen für das Umschalten zwischen Fenstertypen.....	3-19
3.5	Bedienung über Teleservice.....	3-20
4	Einrichten und Bearbeiten des Projekts	4-1
4.1	Was ist ein STEP 7 Lite Projekt.....	4-1
4.2	Einrichten eines Projekts.....	4-4
4.2.1	Anlegen eines Projekts.....	4-4
4.2.2	Einfügen eines Programms.....	4-4
4.3	Bearbeiten eines Projekts.....	4-6
4.3.1	Änderungen übernehmen und speichern.....	4-7
4.3.2	Schritte zum Bearbeiten von Projekten.....	4-9
4.3.2.1	Kopieren eines Projekts.....	4-9
4.3.2.2	Kopieren eines Projektteils.....	4-9
4.3.2.3	Konfigurieren der Hardware (prinzipiell).....	4-10
4.3.2.4	Erstellen der Software im Projekt (prinzipiell).....	4-10
4.4	Löschen und Umbenennen eines Projekts.....	4-10
5	Konfigurieren der Hardware	5-1
5.1	Grundlagen zum Hardware-Konfigurieren mit STEP 7 Lite.....	5-1
5.1.1	Einführung zum Konfigurieren der Hardware.....	5-1
5.1.2	Grundsätzliche Bedienung beim Hardware konfigurieren.....	5-2
5.1.2.1	Grundsätzliche Schritte beim Konfigurieren einer Station.....	5-3
5.1.2.2	Aufbau der Sicht 'HW-Konfiguration'.....	5-4
5.1.2.3	Konfigurationstabelle als Abbild eines Baugruppenträgers.....	5-6
5.1.2.4	Festlegen der Eigenschaften von Komponenten.....	5-7
5.1.2.5	Wissenswertes zu Steckplatz- und sonstigen Regeln.....	5-7
5.2	Konfigurieren der Baugruppen.....	5-8
5.2.1	Regeln für die Anordnung von Baugruppen (SIMATIC 300).....	5-8
5.2.1.1	Besondere Regeln zur Platzhalterbaugruppe (DM 370 Dummy).....	5-9
5.2.1.2	Besondere Regeln zur Digital-Simulationsbaugruppe SIM 374 IN/OUT 16.....	5-9
5.2.2	Regeln für die Anordnung von Modulen (ET 200S und ET 200X).....	5-10
5.2.2.1	Regeln für die Anordnung von Modulen bei ET 200S.....	5-10
5.2.2.2	Regeln für die Anordnung von Modulen bei ET 200X.....	5-10

5.2.3	Schritte zum Konfigurieren von Baugruppen	5-11
5.2.3.1	Überblick: Vorgehensweise beim Konfigurieren und Parametrieren einer Station.....	5-11
5.2.3.2	Auswählen eines Stationstyps	5-11
5.2.3.3	Anordnen von Baugruppen im Baugruppenträger	5-12
5.2.3.4	Anzeige der CPU-Betriebssystem-Version in der Baugruppenliste.....	5-13
5.2.3.5	Anordnen von C7-Komplettsystemen (Besonderheiten)	5-13
5.2.3.6	Festlegen der Eigenschaften von Baugruppen/Schnittstellen	5-14
5.2.3.7	Adressen zuweisen	5-15
5.2.3.8	Ein-/Ausgangsadressen zuweisen.....	5-15
5.2.3.9	Tipps zum Bearbeiten von Stationskonfigurationen	5-16
5.2.4	Wissenswertes zu ET 200S-Motorstartern (High Feature)	5-18
5.2.4.1	Anlagenzustände erkennen durch Stromwerte.....	5-18
5.2.4.2	Blockierstrom	5-18
5.2.4.3	Blockierzeit	5-18
5.2.4.4	Verhalten bei Nullstromerkennung.....	5-19
5.2.4.5	Unsymmetrie	5-19
5.2.4.6	Thermisches Motormodell.....	5-19
5.2.4.7	Wiederbereitschaftszeit.....	5-19
5.2.4.8	Übersicht: Mögliche Aktionen des Motorstarters	5-20
5.2.4.9	Belegung des Motorstarters im Prozessabbild	5-21
5.3	Speichern einer Konfiguration und Konsistenzprüfung.....	5-22
6	Programmieren von Bausteinen	6-1
6.1	Festlegen von Symbolen.....	6-1
6.1.1	Absolute und symbolische Adressierung	6-1
6.1.2	Globale und lokale Symbole	6-3
6.1.3	Darstellung von globalen oder lokalen Symbolen.....	6-4
6.1.4	Einstellen des Operandenvorrangs (absolut/symbolisch).....	6-4
6.1.5	Symboltabelle für globale Symbole.....	6-6
6.1.5.1	Struktur und Bestandteile der Symboltabelle.....	6-6
6.1.5.2	Zulässige Adressen und Datentypen in der Symboltabelle	6-8
6.1.5.3	Unvollständige und mehrdeutige Symbole in der Symboltabelle.....	6-9
6.1.6	Eingabemöglichkeiten von globalen Symbolen	6-10
6.1.6.1	Allgemeine Hinweise zur Eingabe von Symbolen	6-11
6.1.6.2	Eingeben einzelner globaler Symbole im Dialog	6-12
6.1.6.3	Eingeben mehrerer globaler Symbole in der Symboltabelle.....	6-13
6.1.6.4	Exportieren und Importieren von Symboltabellen	6-14
6.1.7	Schritte zum Bearbeiten der Symboltabelle.....	6-14
6.1.7.1	Öffnen einer Symboltabelle.....	6-14
6.1.7.2	Definieren einzelner Symbole	6-14
6.1.7.3	Einfügen von Symbolzeilen.....	6-15
6.1.7.4	Löschen von Symbolzeilen	6-15
6.1.7.5	Filtern der Symboltabelle	6-16
6.1.7.6	Nicht verwendete Symbole	6-16
6.1.7.7	Operanden ohne Symbol	6-17
6.1.7.8	Sortieren der Symboltabelle.....	6-17
6.1.7.9	Markieren von Symbolzeilen.....	6-17
6.1.7.10	Kopieren von Symbolzeilen in die Zwischenablage.....	6-17
6.1.7.11	Speichern einer Symboltabelle	6-18
6.1.8	Schritte zum Ändern der Fenster-Einstellungen	6-18
6.1.8.1	Ein-/Ausschalten der Symbolleiste	6-18
6.1.8.2	Ein-/Ausschalten der Statusleiste	6-18
6.1.8.3	Anordnen der Symbolleiste.....	6-18
6.1.8.4	Einstellen der Anzeigegröße eines Fensters	6-19

6.2	Arbeiten mit Bausteinen	6-20
6.2.1	Bausteineditor	6-20
6.2.2	Auswahl der Programmiersprache.....	6-21
6.2.2.1	Programmiersprachen des Bausteineditors.....	6-21
6.2.2.2	Programmiersprache KOP (Kontaktplan)	6-22
6.2.2.3	Programmiersprache FUP (Funktionsplan)	6-23
6.2.2.4	Programmiersprache AWL (Anweisungsliste)	6-23
6.2.3	Anlegen von Bausteinen	6-24
6.2.3.1	Anwenderdefinierte Datentypen (UDT).....	6-24
6.2.3.2	Bausteineigenschaften.....	6-25
6.2.3.3	Bausteinschutz setzen	6-27
6.2.3.4	Zulässige Bausteineigenschaften je Bausteinart	6-28
6.2.3.5	Anzeige von Baustein-Längen	6-29
6.2.3.6	Bausteine vergleichen.....	6-30
6.2.4	Arbeiten mit Bibliotheken	6-32
6.2.4.1	Übersicht der Bausteinbibliotheken	6-32
6.3	Erstellen von Codebausteinen	6-33
6.3.1	Grundlagen zum Erstellen von Codebausteinen	6-33
6.3.1.1	Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen.....	6-33
6.3.1.2	Voreinstellungen für Bausteineditor KOP/FUP/AWL	6-34
6.3.1.3	Anweisungen aus den Befehlsbibliotheken	6-34
6.3.1.4	Festlegen der Ansicht des Bausteineditors.....	6-35
6.3.2	Editieren der Variablendeklarationstabelle	6-37
6.3.2.1	Verwendung der Variablendeklaration in Codebausteinen.....	6-37
6.3.2.2	Zusammenspiel zwischen Variablendeklarationstabelle und Anweisungsteil.....	6-38
6.3.2.3	Aufbau der Variablendeklarationstabelle	6-39
6.3.2.4	Allgemeine Hinweise zu Variablendeklarationstabellen	6-40
6.3.2.5	Schritte zum Arbeiten mit der Variablendeklarationstabelle	6-41
6.3.3	Multiinstanzen in der Variablendeklarationstabelle.....	6-45
6.3.3.1	Verwendung von Multiinstanzen	6-45
6.3.3.2	Regeln für die Bildung von Multiinstanzen.....	6-46
6.3.3.3	Eingeben der Multiinstanz in der Variablendeklarationstabelle	6-46
6.3.4	Allgemeine Hinweise zum Editieren von Anweisungen und Kommentaren ...	6-47
6.3.4.1	Aufbau des Anweisungsteils	6-47
6.3.4.2	Vorgehensweise beim Eingeben von Anweisungen.....	6-48
6.3.4.3	Eingeben von globalen Symbolen in ein Programm	6-49
6.3.4.4	Titel und Kommentare zu Bausteinen und Netzwerken.....	6-49
6.3.4.5	Eingeben von Baustein-/Netzwerkcommentaren.....	6-50
6.3.4.6	Suchfunktion für Fehler im Anweisungsteil.....	6-51
6.3.4.7	Umverdrahten	6-51
6.3.5	Editieren von KOP-Anweisungen im Anweisungsteil.....	6-52
6.3.5.1	Einstellungen für die Programmiersprache KOP	6-52
6.3.5.2	Regeln für die Eingabe von KOP-Anweisungen	6-53
6.3.5.3	Unzulässige Verschaltungen in KOP	6-55
6.3.5.4	Schritte zum Eingeben von KOP-Elementen	6-56
6.3.6	Editieren von FUP-Anweisungen im Anweisungsteil	6-62
6.3.6.1	Einstellungen für die Programmiersprache FUP.....	6-62
6.3.6.2	Regeln für die Eingabe von FUP-Anweisungen.....	6-63
6.3.6.3	Schritte zum Eingeben von FUP-Elementen	6-65
6.3.7	Editieren von AWL-Anweisungen im Anweisungsteil	6-69
6.3.7.1	Einstellungen für die Programmiersprache AWL.....	6-69
6.3.7.2	Regeln für die Eingabe von AWL-Anweisungen	6-69
6.3.7.3	Schritte zum Eingeben von AWL-Anweisungen	6-70

6.3.8	Aktualisieren von Bausteinaufrufen	6-72
6.4	Erstellen von Datenbausteinen	6-73
6.4.1	Grundlagen zum Erstellen von Datenbausteinen	6-73
6.4.2	Deklarationssicht von Datenbausteinen.....	6-74
6.4.3	Datensicht von Datenbausteinen	6-75
6.4.4	Eingeben in Datenbausteine und Speichern.....	6-76
6.4.4.1	Eingeben der Datenstruktur von globalen Datenbausteinen	6-76
6.4.4.2	Eingeben/Anzeige der Datenstruktur von Datenbausteinen mit zugeordnetem FB (Instanz-DBs)	6-76
6.4.4.3	Eingeben der Struktur von anwenderdefinierten Datentypen (UDT)	6-78
6.4.4.4	Eingeben/Anzeige der Struktur von Datenbausteinen mit zugeordnetem UDT	6-78
6.4.4.5	Ändern von Datenwerten in der Datensicht	6-79
6.4.4.6	Rücksetzen von Datenwerten auf die Anfangswerte	6-80
6.5	Anzeigen von Referenzen.....	6-81
6.5.1	Übersicht der möglichen Referenzen.....	6-81
6.5.2	Adressübersicht	6-82
6.5.3	Querverweisliste.....	6-82
6.5.4	Verwendete Operanden	6-84
6.5.5	Programmstruktur	6-86
6.5.6	Arbeiten mit Referenzdaten	6-91
6.5.6.1	Schnelles Positionieren auf Verwendungsstellen im Programm	6-91
6.5.6.2	Beispiel zum Arbeiten mit Verwendungsstellen	6-92
6.5.6.3	Schritte zum Arbeiten mit Referenzdaten	6-94
6.6	Programmkonsistenz herstellen und Zeitstempel als Bausteineigenschaft....	6-96
6.6.1	Programmkonsistenz herstellen.....	6-96
6.6.2	Zeitstempel und Zeitstempelkonflikte.....	6-97
6.6.3	Zeitstempel in Codebausteinen	6-98
6.6.4	Zeitstempel bei globalen Datenbausteinen.....	6-99
6.6.5	Zeitstempel bei Instanz-Datenbausteinen.....	6-99
6.6.6	Zeitstempel bei UDTs und von UDTs abgeleiteten DBs.....	6-100
6.6.7	Vermeiden von Fehlern beim Aufrufen von Bausteinen	6-101
6.6.8	Hinweise zur Änderung von Register-Inhalten.....	6-103
7	Aufbau der Online-Verbindung und CPU-Einstellung	7-1
7.1	Aufbau von Online-Verbindungen	7-1
7.1.1	Passwortschutz für Zugriff auf Zielsysteme	7-2
7.2	Anzeigen und Ändern des Betriebszustands	7-3
7.3	Anzeigen und Einstellen von Uhrzeit und Datum.....	7-3
8	Importieren, Exportieren, Speichern unter	8-1
8.1	Importieren, Exportieren, Speichern unter	8-1
8.2	Projekte auf Datenträger sichern	8-2
8.3	Projektdaten auf Micro Memory Card (MMC) ablegen	8-2
8.4	Micro Memory Card als Datenträger verwenden	8-5
8.5	Projektdaten zwischen STEP 7 Lite und STEP 7 austauschen.....	8-6
8.6	Projektdaten für externe Editoren exportieren	8-8
8.6.1	Datenformat für den Import/Export einer Symboltabelle.....	8-8
8.6.2	Texte mehrsprachig verwalten	8-8
8.6.2.1	Texttypen mehrsprachig verwalteter Texte.....	8-10
8.6.2.2	Aufbau der Export-Datei.....	8-10
8.6.2.3	Schritte zum Verwalten mehrsprachiger Texte	8-12
8.6.2.4	Tipps zum Übersetzen	8-14

9	Laden in CPU und in PG	9-1
9.1	Laden aus dem PG in die CPU	9-1
9.1.1	Voraussetzungen für das Laden	9-1
9.1.2	Was wird wann geladen?	9-2
9.1.3	Unterschied zwischen Speichern und Laden von Bausteinen	9-3
9.1.4	Lade- und Arbeitsspeicher in der CPU	9-3
9.1.5	Lademöglichkeiten abhängig vom Ladespeicher	9-4
9.1.6	Bausteine und Konfiguration in CPU laden und auf Memory Card speichern ..	9-5
9.1.6.1	Nachladen von Bausteinen in die CPU	9-5
9.1.6.2	Speichern geladener Bausteine auf integriertem EPROM bzw. auf S7-Memory Card in der CPU	9-6
9.1.6.3	Laden einer Konfiguration in ein Zielsystem	9-7
9.2	Laden aus der CPU in das PG	9-8
9.2.1	Was wird wann zurückgeladen?	9-8
9.2.2	Schritte zum Laden von Objekten aus der CPU in das PG/PC	9-10
9.2.3	Bearbeiten von geladenen Bausteinen im PG/PC	9-10
9.2.4	Bearbeiten einer geladenen Hardware-Konfiguration im PG/PC	9-11
9.3	Löschen in der CPU	9-12
9.3.1	Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU	9-12
9.3.2	Löschen einzelner Bausteine in der CPU	9-13
9.3.3	Löschen der Memory Card in der CPU	9-14
9.4	Komprimieren des Anwenderspeichers (RAM)	9-15
9.4.1	Entstehung von Lücken im Anwenderspeicher (RAM)	9-15
9.4.2	Komprimieren des Speicherinhalts einer CPU	9-16
10	Testen	10-1
10.1	Übersicht der verschiedenen Testarten	10-1
10.2	Testen mit Variablen Tabellen und Forcetafellen	10-1
10.2.1	Einführung zum Testen mit Variablen Tabellen und Forcetafellen	10-1
10.2.2	Prinzipielle Vorgehensweise beim Beobachten und Steuern mit Variablen Tabellen	10-2
10.2.3	Prinzipielle Vorgehensweise beim Beobachten und Forcen mit Forcetafellen	10-2
10.2.4	Bearbeiten und Speichern von Variablen Tabellen und Forcetafellen	10-3
10.2.4.1	Erstellen und Öffnen einer Variablen Tabelle	10-3
10.2.4.2	Erstellen und Öffnen einer Forcetafelle	10-3
10.2.4.3	Kopieren / Duplizieren von Variablen Tabellen	10-4
10.2.4.4	Kopieren / Duplizieren von Forcetafellen	10-5
10.2.4.5	Speichern einer Variablen Tabelle	10-5
10.2.4.6	Speichern einer Forcetafelle	10-6
10.2.5	Eingeben von Variablen in Variablen Tabellen und Forcetafellen	10-6
10.2.5.1	Eingabe von Operanden oder Symbolen in eine Variablen Tabelle	10-6
10.2.5.2	Eingabe von Operanden oder Symbolen in eine Forcetafelle	10-7
10.2.5.3	Einfügen eines zusammenhängenden Operandenbereichs in eine Variablen Tabelle	10-9
10.2.5.4	Einfügen eines zusammenhängenden Operandenbereichs in eine Forcetafelle	10-9
10.2.5.5	Obergrenzen für die Eingabe von Zeiten	10-10
10.2.5.6	Obergrenzen für die Eingabe von Zählern	10-11
10.2.5.7	Beispiele	10-12

10.2.6	Bearbeiten von Variablen in Variablentabellen und Forcetabellen.....	10-16
10.2.6.1	Festlegen des Anzeigeformats	10-16
10.2.6.2	Ausschneiden von markierten Bereichen in die Zwischenablage.....	10-16
10.2.6.3	Einfügen von Bereichen aus der Zwischenablage in die Variablentabelle bzw. Forcetabelle	10-16
10.2.6.4	Kopieren von markierten Bereichen in die Zwischenablage.....	10-16
10.2.7	Beobachten von Variablen	10-17
10.2.7.1	Einführung zum Beobachten von Variablen.....	10-17
10.2.7.2	Festlegen des Beobachtmodus	10-17
10.2.7.3	Beobachten von Variablen.....	10-18
10.2.7.4	Beobachten von Variablen einmalig und sofort	10-19
10.2.8	Steuern von Variablen.....	10-20
10.2.8.1	Einführung zum Steuern von Variablen	10-20
10.2.8.2	Festlegen des Steuermodus	10-20
10.2.8.3	Steuern von Variablen.....	10-22
10.2.8.4	Sofortiges Steuern von Variablen	10-22
10.2.8.5	Steuern: CPU im STOP mit eigenen Werten initialisieren	10-23
10.2.8.6	Steuern der Peripherieausgänge im STOP-Zustand der CPU	10-23
10.2.9	Forcen von Variablen	10-24
10.2.9.1	Einführung zum Forcen von Variablen.....	10-24
10.2.9.2	Sicherheitsvorkehrungen zum Forcen von Variablen	10-25
10.2.9.3	Anzeigen der Werte, die von der CPU geforct werden	10-25
10.2.9.4	Werte forcen.....	10-25
10.2.9.5	Werte entforcen.....	10-26
10.2.9.6	Unterschiede zwischen Forcen und Steuern von Variablen	10-26
10.3	Testen mit Programmstatus	10-27
10.3.1	Testen mit Programmstatus	10-27
10.3.2	Anzeigen im Programmstatus	10-28
10.3.3	Programmstatus von Datenbausteinen.....	10-29
10.3.4	Schritte zum Testen im Programmstatus.....	10-30
10.3.4.1	Festlegen der Anzeigen für den Programmstatus	10-30
10.3.4.2	Einstellen der Aufrufumgebung des Bausteins	10-31
10.3.4.3	Festlegen der Betriebsart für den Test	10-32
10.3.4.4	Steuern von Variablen im Programmstatus	10-33
10.3.4.5	Ein- und Ausschalten des Tests mit Programmstatus	10-33
11	Diagnose	11-1
11.1	Diagnosefunktionen	11-1
11.2	Diagnose der Hardware und Fehlersuche	11-2
11.3	Vergleich der Konfiguration 'Online/Offline/Physik'	11-2
11.4	Aufbau der Sicht 'HW-Vergleich'	11-4
11.5	Gestörte Baugruppen ermitteln	11-5
11.6	Aufbau der Sicht 'HW-Diagnose'	11-6
11.7	Baugruppenzustand	11-7
11.7.1	Aufruf des Baugruppenzustands.....	11-7
11.7.2	Auskunftsfunktionen des Baugruppenzustands.....	11-9
11.7.3	Baugruppentyp-abhängiger Umfang der Auskunftsfunktionen im Baugruppenzustand	11-11
11.8	Diagnose im Betriebszustand STOP	11-12
11.8.1	Prinzipielles Vorgehen zum Ermitteln einer STOP-Ursache.....	11-12
11.8.2	Stack-Inhalte im Betriebszustand STOP.....	11-12
11.8.3	Öffnen des Bausteins zu einem Diagnosepuffer- oder Stack-Eintrag	11-14
11.8.3.1	Öffnen des Bausteins zu einem Diagnosepuffereintrag	11-14
11.8.3.2	Öffnen des Bausteins aus der B-Stack-Liste	11-15
11.8.3.3	Öffnen des Bausteins aus der U-Stack-Liste	11-15

11.9	Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern	11-16
11.9.1	Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern	11-16
11.10	Übermittlung von Diagnoseinformationen	11-17
11.10.1	Übermittlung von Diagnoseinformationen	11-17
11.10.2	Systemzustandsliste SZL	11-18
11.10.3	Eigene Diagnosemeldungen senden	11-20
11.11	Maßnahmen im Programm zur Störungsbehandlung	11-21
11.11.1	Auswerten des Ausgangsparameter RET_VAL	11-22
11.11.2	Fehler-OBs als Reaktion auf die Erkennung eines Fehlers	11-23
11.11.3	Einfügen von Ersatzwerten bei Fehlererkennung	11-26
11.11.4	Zeitfehler (OB 80)	11-29
11.11.5	Stromversorgungsfehler (OB 81)	11-29
11.11.6	Diagnosealarm (OB 82)	11-30
11.11.7	CPU-Hardwarefehler (OB 84)	11-31
11.11.8	Programmablauffehler (OB 85)	11-32
11.11.9	Baugruppenträgerausfall (OB 86)	11-33
11.11.10	Kommunikationsfehler (OB 87)	11-34
11.11.11	Programmierfehler (OB 121)	11-35
11.11.12	Peripheriezugriffsfehler (OB 122)	11-36
12	Drucken einer Projektdokumentation	12-1
12.1	Übersicht zur Projektdokumentation	12-1
12.2	Projektdokumentation zusammenstellen	12-3
12.3	Druckobjekte	12-5
12.4	Optionen, Schriftart und Seitenlayout festlegen	12-7
12.5	Vorlagen definieren und verwenden	12-10
12.6	Projektdokumentation drucken	12-14
13	Tipps und Tricks	13-1
13.1	Baugruppen tauschen in der Hardware-Konfiguration	13-1
13.2	Testen mit der Variablen-tabelle	13-1
13.3	Arbeiten ohne Original-Projekt auf dem PG/PC	13-2
A	Anhang	A-1
A.1	Betriebszustände	A-1
A.1.1	Betriebszustände und Übergänge	A-1
A.1.2	Betriebszustand STOP	A-4
A.1.3	Betriebszustand ANLAUF	A-4
A.1.4	Betriebszustand RUN	A-9
A.1.5	Betriebszustand HALT	A-11
A.2	Speicherbereiche von S7-CPU's	A-12
A.2.1	Aufteilung der Speicherbereiche	A-12
A.2.2	Lade- und Arbeitsspeicher	A-13
A.2.3	Systemspeicher	A-15
A.2.3.1	Verwenden der Systemspeicherbereiche	A-15
A.2.3.2	Prozessabbild der Ein-/Ausgänge	A-17
A.2.3.3	Lokaldaten-Stack	A-18
A.2.3.4	Unterbrechungs-Stack	A-19
A.2.3.5	Baustein-Stack	A-20
A.2.3.6	Diagnosepuffer	A-21
A.2.3.7	Auswertung des Diagnosepuffers	A-21
A.2.3.8	Remanente Speicherbereiche in S7-300-CPU's	A-23

A.3	Daten- und Parametertypen.....	A-26
A.3.1	Einführung zu Daten- und Parametertypen	A-26
A.3.2	Elementare Datentypen	A-27
A.3.2.1	Format des Datentyps INT (16-Bit-Ganzzahlen).....	A-28
A.3.2.2	Format des Datentyps DINT (32-Bit-Ganzzahlen).....	A-28
A.3.2.3	Format des Datentyps REAL (Gleitpunktzahlen).....	A-29
A.3.2.4	Format des Datentyps WORD	A-33
A.3.2.5	Format des Datentyps DWORD.....	A-33
A.3.2.6	Format der Datentypen WORD und DWORD bei binär-codierten Dezimalzahlen	A-34
A.3.2.7	Format des Datentyps S5TIME (Zeitdauer)	A-35
A.3.2.8	Format des Datentyps TIME	A-36
A.3.3	Zusammengesetzte Datentypen	A-37
A.3.3.1	Format des Datentyps DATE_AND_TIME (Datum und Uhrzeit)	A-38
A.3.3.2	Format des Datentyps STRING	A-39
A.3.3.3	Format des Datentyps ARRAY	A-40
A.3.3.4	Format des Datentyps STRUCT	A-41
A.3.3.5	Verwenden von zusammengesetzten Datentypen	A-42
A.3.3.6	Verwenden von Feldern für den Datenzugriff	A-43
A.3.3.7	Verwenden von Strukturen für den Datenzugriff.....	A-46
A.3.3.8	Verwenden von anwenderdefinierten Datentypen für den Datenzugriff	A-48
A.3.4	Parametertypen.....	A-50
A.3.4.1	Format der Parametertypen BLOCK, COUNTER und TIMER	A-51
A.3.4.2	Format des Parametertyps POINTER	A-52
A.3.4.3	Verwenden des Parametertyps POINTER.....	A-53
A.3.4.4	Baustein zum Verändern des Pointers	A-54
A.3.4.5	Format des Parametertyps ANY	A-57
A.3.4.6	Verwenden des Parametertyps ANY	A-60
A.3.4.7	Zuordnen von Datentypen zu Lokaldaten von Codebausteinen.....	A-63
A.3.4.8	Zulässige Datentypen beim Übergeben von Parametern.....	A-65
A.3.4.9	Übergabe an In_out-Parameter eines FB.....	A-70
A.4	Beispielprogramme	A-70
A.4.1	Beispielprojekte und Beispielprogramme.....	A-70
A.4.2	Beispiel für das Maskieren und Demaskieren von Synchronfehlerereignissen.....	A-71
A.4.3	Beispiel zum Sperren und Freigeben von Alarm- und Asynchronereignissen (SFC 39 und 40)	A-75
A.4.4	Beispiel zur verzögerten Bearbeitung von Alarm- und Asynchronereignissen (SFC 41 und 42)	A-76
A.4.5	Beispielprogramm für einen industriellen Mischprozess.....	A-77
A.4.5.1	Beispielprogramm für einen industriellen Mischprozess.....	A-77
A.4.5.2	Definieren von Codebausteinen.....	A-80
A.4.5.3	Zuordnen von symbolischen Namen	A-81
A.4.5.4	Erstellen des FB für den Motor	A-83
A.4.5.5	Erstellen der FC für die Ventile	A-87
A.4.5.6	Erstellen des OB 1	A-89
A.4.6	Beispiel zur Hantierung mit Uhrzeitalarmen.....	A-95
A.4.6.1	Struktur des Anwenderprogramms Uhrzeitalarme.....	A-95
A.4.6.2	FC 12.....	A-97
A.4.6.3	OB 10	A-99
A.4.6.4	OB 1 und OB 80.....	A-101
A.4.7	Beispiel zur Hantierung mit Verzögerungsalarman	A-102
A.4.7.1	Struktur des Anwenderprogramms Verzögerungsalarman.....	A-102
A.4.7.2	OB 20	A-104
A.4.7.3	OB 1	A-105

A.5	Zugriff auf Prozess- und Peripheriedatenbereiche	A-107
A.5.1	Zugriff auf den Prozessdatenbereich	A-107
A.5.2	Zugriff auf den Peripheriedatenbereich.....	A-108
A.6	Einstellen des Betriebsverhaltens	A-110
A.6.1	Einstellen des Betriebsverhaltens	A-110
A.6.2	Ändern des Verhaltens und der Eigenschaften von Baugruppen.....	A-110
A.6.3	Nutzen der Uhrzeitfunktionen	A-112
A.6.4	Verwenden von Taktmerkern und Zeiten	A-113

Index

1 **Produkteinführung und Installation**

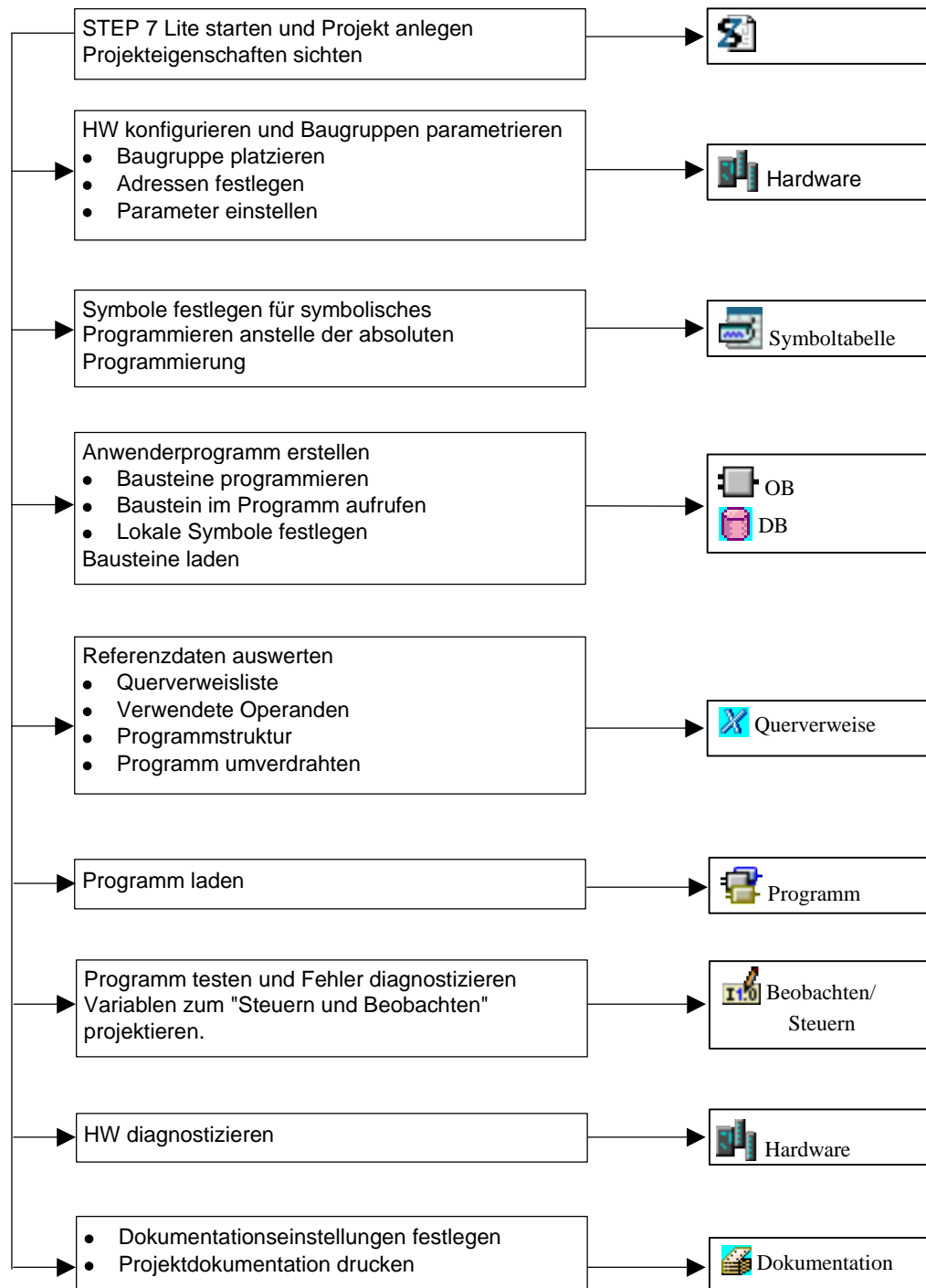
1.1 **Leitfaden durch STEP 7 Lite**

Unterstützte Hardware

STEP 7 Lite ist das Softwarepaket zur Konfigurierung und Programmierung von SIMATIC-Automatisierungssystemen der Familien S7-300 und C7 sowie der Familien ET 200X und ET 200S (stand alone).

Arbeitsschritte für eine Automatisierungslösung

Bei der Erstellung einer Automatisierungslösung mit STEP 7 Lite fallen verschiedene Aufgaben an. Das folgende Bild zeigt die Aufgaben, die in den meisten Projekten durchgeführt werden, und ordnet sie zu einer beispielhaften Vorgehensweise in Form eines Leitfadens an.



Kurzbeschreibung der einzelnen Arbeitsschritte:

- **Installieren und Autorisieren**
Bei der Erstanwendung installieren Sie STEP 7 Lite und übertragen die Autorisierung von Diskette auf die Festplatte (siehe auch Installieren von STEP 7 Lite und Autorisierung).
- **Programmstruktur entwerfen**
Die beim Steuerungsentwurf beschriebenen Aufgaben setzen Sie mit den von STEP 7 Lite zur Verfügung gestellten Bausteinen in eine Programmstruktur um (siehe auch Bausteine im Anwenderprogramm).
- **STEP 7 Lite starten**
Sie starten STEP 7 Lite von der Windows-Oberfläche aus (siehe auch Starten von STEP 7 Lite).
- **Projekt anlegen**
Ein Projekt ist wie ein Ordner, in dem alle Daten hierarchisch gegliedert abgelegt werden und jederzeit zur Verfügung stehen. Nachdem Sie ein Projekt angelegt haben, werden alle weiteren Arbeiten in diesem Projekt ausgeführt (siehe auch Projektstruktur).
- **Station einrichten**
Mit dem Einrichten der Station legen Sie die Steuerung fest: z. B. SIMATIC 300.
- **Hardware konfigurieren**
Beim Konfigurieren legen Sie fest, welche Baugruppen Sie für Ihre Automatisierungslösung einsetzen und über welche Adressen die Baugruppen aus dem Anwenderprogramm heraus angesprochen werden sollen. Darüber hinaus lassen sich noch Eigenschaften der Baugruppen über Parameter einstellen (siehe auch Grundsätzliche Bedienung beim Hardware konfigurieren).
- **Symbole festlegen**
Sie können in einer Symboltabelle anstelle von Adressen lokale oder globale Symbole mit einer aussagekräftigen Bezeichnung definieren, um sie dann in Ihrem Programm zu verwenden (siehe auch Öffnen einer Symboltabelle).
- **Programm erstellen**
Mit einer der zur Verfügung stehenden Programmiersprachen erstellen Sie ein Programm und hinterlegen dieses in Form von Bausteinen (siehe auch Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen).
- **Referenzdaten auswerten**
Sie können sich dieser Referenzdaten bedienen, um sich das Testen und Ändern Ihres Programms zu erleichtern (siehe auch Übersicht der möglichen Referenzdaten).
- **Programme in die CPU laden**
Nach Abschluss von Konfiguration, Parametrierung und Programmerstellung können Sie Ihr komplettes Programm oder einzelne Bausteine in die CPU übertragen (siehe auch Voraussetzungen für das Laden). Die CPU enthält bereits das Betriebssystem.
- **Programme testen**
Zum Testen haben Sie die Möglichkeit, sich Werte von Variablen aus Ihrem Anwenderprogramm oder einer CPU anzeigen zu lassen, diesen Werte zuzuweisen und eine Variablen-tabelle für die Variablen zu erstellen, die Sie anzeigen oder steuern wollen (siehe auch Einführung zum Testen mit Variablen-tabellen und Forcetabellen).

- Betrieb überwachen, Hardware diagnostizieren
Sie ermitteln die Ursache einer Baugruppenstörung, indem Sie sich Online-Informationen zu einer Baugruppe anzeigen lassen. Die Ursache für die Störung im Ablauf eines Programms ermitteln Sie mit Hilfe des Diagnosepuffers und der Stack-Inhalte. Darüber hinaus können Sie prüfen, OB ein Programm auf einer bestimmten CPU ablauffähig ist (siehe auch Diagnose der Hardware und Anzeigen des Baugruppenzustands).
- Drucken

Programmiersprachen

Die in STEP 7 Lite integrierten SIMATIC-Programmiersprachen erfüllen die Norm DIN EN 6.1131-3.

- KOP (Kontaktplan) ist eine grafische Programmiersprache. Die Syntax der Anweisungen ähnelt einem Stromlaufplan. KOP ermöglicht Ihnen eine einfache Verfolgung des Signalflusses zwischen Stromschienen über Kontakte, komplexe Elemente und Spulen.
- AWL (Anweisungsliste) ist eine textuelle, maschinennahe Programmiersprache. Wird ein Programm in AWL programmiert, so entsprechen die einzelnen Anweisungen weitgehend den Arbeitsschritten, mit denen die CPU das Programm bearbeitet. Zur Erleichterung der Programmierung wurde AWL um einige Hochsprachenkonstrukte (wie z. B. strukturierte Datenzugriffe und Bausteinparameter) erweitert.
- FUP (Funktionsplan) ist eine grafische Programmiersprache und benutzt zur Darstellung der Logik die von der Booleschen Algebra bekannten logischen Boxen. Außerdem können komplexe Funktionen (z. B. mathematische Funktionen) direkt in Verbindung mit den logischen Boxen dargestellt werden.

1.2 Projektfenster und Sichten in STEP 7 Lite

Projektfenster und Sichten

Wenn Sie STEP 7 Lite gestartet haben, dann ist im linken Bereich des Fensters das Projektfenster angeordnet.

Alle zentralen Objekte (z. B. das Programm) und Funktionen (z. B. Beobachten und Steuern) sind bei STEP 7 Lite über das Projektfenster erreichbar.

Wenn Sie auf das Register "Online-CPU" klicken, das sich am unteren Rand des Projektfensters neben dem Register "Projekt" befindet, sehen Sie die Objekte in der CPU (siehe Umschalten auf Online-CPU).

Ansicht des Projektfensters (Projekt)	Verweise zu Erläuterungen
	<ul style="list-style-type: none"> Projekt Hardware Symboltabelle Steuern und Beobachten Querverweise Projektdokumentation Programm Bausteine Symbole im Projektfenster Importieren, Exportieren, Speichern unter

Wenn Sie auf ein Element im Projektfenster doppelklicken, dann wird im Arbeitsbereich ein Fenster mit einer oder mehreren Sichten geöffnet. Wenn ein Element mehrere Sichten hat, dann sind sie über Register am unteren Rand des Fensters umschaltbar.

Beispiel

Das Element "Hardware" hat die Sichten "HW-Konfiguration", "HW-Vergleich" und "HW-Diagnose".

Projekt

Die vorgegebene Bezeichnung "Neues Projekt" können Sie mit Hilfe der Menübefehle **Datei > Speichern** bzw. **Speichern unter** ändern.

Mögliche Sichten	Bedeutung	Siehe auch
Projektübersicht	Hardware-Konfiguration sowie Übersicht aller Bausteine im Projekt mit Angabe der Eigenschaften wie z. B. Bausteinsymbol, Größe, Erstellungssprache etc.	Bausteine im Anwenderprogramm

Hardware

Die vorgegebene Bezeichnung im Projektfenster können Sie nicht ändern.

Mögliche Sichten	Bedeutung	Siehe auch
HW-Vergleich	Für die Gegenüberstellung zwischen projektierter Konfiguration und in die CPU geladener Konfiguration; Visualisierung von Unterschieden (z. B. unterschiedlicher Baugruppentyp oder fehlende Baugruppen)	Vergleich zwischen geladener und projektierter Konfiguration
HW-Konfiguration	Für die Anordnung von Baugruppen aus einem Katalog in Baugruppenträgern und Parametrierung von Baugruppen.	Einführung zum Konfigurieren der Hardware Festlegen der Eigenschaften von Komponenten
HW-Diagnose	Für die Visualisierung der Baugruppenzustände; aus dieser Sicht sind detailliertere Diagnoseinformationen abrufbar	Diagnose der Hardware und Fehlersuche

Symboltabelle

Die vorgegebene Bezeichnung im Projektfenster können Sie nicht ändern.

Mögliche Sichten	Bedeutung	Siehe auch
Symboltabelle	Für die Anzeige und Bearbeitung von globalen Symbolen für Bausteine und Operanden	Symboltabelle für globale Symbole

Beobachten/Steuern

Die vorgegebene Bezeichnung im Projektfenster können Sie nicht ändern.

Mögliche Sichten	Bedeutung	Siehe auch
Beobachten/Steuern	Für die Beobachtung des Status von Operanden in einer Variablen-tabelle sowie für die Steuerung von Operanden	Einführung zum Beobachten von Variablen Einführung zum Steuern von Variablen
Forcen	Für das Forcen von Operanden und für das Löschen von Forceaufträgen	Einführung zum Forcen von Variablen

Querverweise

Die vorgegebene Bezeichnung im Projektfenster können Sie nicht ändern.

Mögliche Sichten	Bedeutung	Siehe auch
Querverweisliste	Zeigt für alle verwendeten Operanden an, in welchen Bausteinen und an welcher Stelle sie verwendet werden	Querverweisliste Umverdrahten
Verwendete Operanden	Zeigt an, welche Operanden mit welchem Zugriff (bit- byte- wort- oder doppelwortweiser Zugriff)	Verwendete Operanden
Programmstruktur	Zeigt alle vorhandenen Bausteine und die Aufrufhierarchien zwischen den Bausteinen grafisch an	Programmstruktur

Projektdokumentation

Die vorgegebene Bezeichnung im Projektfenster können Sie nicht ändern.

Mögliche Sichten	Bedeutung	Siehe auch
Projekt-dokumentation	Für die Erstellung der Projektdokumentation; Druckobjekte wie z. B. Querverweis oder Symboltabelle können angeordnet werden, Dokumentationsvorlagen verwaltet werden, Layout kann für den Druck angepasst werden	Übersicht zur Projektdokumentation

Programm

Die vorgegebene Bezeichnung im Projektfenster können Sie nicht ändern.

Mögliche Sichten	Bedeutung	Siehe auch
Übersicht	Übersicht aller Bausteine im Projekt mit Angabe der Eigenschaften wie z. B. Bausteinsymbol, Größe, Erstellungssprache etc.	Bausteine im Anwenderprogramm Programmkonsistenz herstellen

Bausteine

Die Bezeichnung im Projektfenster ergibt sich aus dem Bausteintyp und Nummer, den Sie bei der Erzeugung eines neuen Bausteins vergeben.

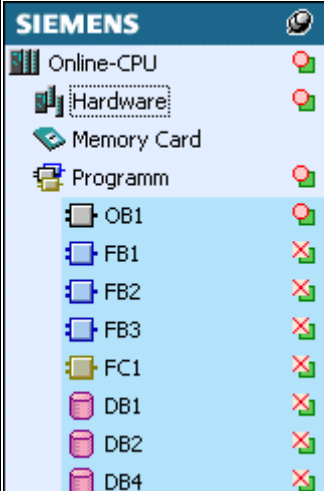
Mögliche Sichten	Bedeutung	Siehe auch
Bausteineditor	Editor mit Deklarations- und Anweisungsteil zum Erstellen des Programms innerhalb des markierten Bausteins	Bausteineditor Editieren von AWL-Anweisungen im Anweisungsteil Editieren von FUP-Anweisungen im Anweisungsteil Editieren von KOP-Anweisungen im Anweisungsteil Testen mit Programmstatus
Eigenschaften	Für die Anzeige von Bausteineigenschaften wie Name, Längen, Speicherbedarf etc. und für die Eingabe/Änderung des Symbols, verschiedenen Kommentaren und Bausteinattributen	Bausteineigenschaften Einstellen der Bausteineigenschaften

Umschalten auf "Online-CPU"

Das Projektfenster selbst kann umgeschaltet werden. Am unteren Rand des Projektfensters können Sie auf das Register "Online-CPU" umschalten, um auf die CPU-Inhalte (geladene Bausteine und geladene Hardware-Konfiguration) bzw. auf die Memory Card/Micro Memory Card (MMC) in der CPU zugreifen zu können. Diese Ansicht ist im Gegensatz zum "Offline"-Projektfenster farbig hinterlegt.

In der Ansicht "Online-CPU" können Sie keine Änderungen vornehmen.

Beispiel:

Ansicht "Online CPU"	Verweise zu Erläuterungen
	<p>Besonderheiten:</p> <p>Symbole am rechten Rand der Online-Sicht symbolisieren z. B. Gleichheiten oder Ungleichheiten zwischen Offline- und Online-Objekt (z. B. Programm oder HW-Konfiguration).</p> <p>Wenn im Projekt Bausteine gespeichert sind, die nicht in die CPU geladen wurden, erscheint das Symbol "Offline Bausteine".</p> <p>Hardware</p> <p>Wissenswertes zu Micro Memory Cards (MMC)</p> <p>Programm</p> <p>Bausteine</p> <p>Symbole im Projektfenster</p> <p>Voraussetzungen für das Laden</p> <p>Arbeiten ohne Original-Projekt auf dem PG/PC</p>

1.3 Hilfe und Dokumentation zu STEP 7 Lite

Die Dokumentation zu STEP 7 Lite steht Ihnen als Online-Hilfe zur Verfügung. Die Online-Hilfe ist zweigeteilt:

- Hinweise zum unmittelbaren Funktionskontext erhalten Sie über den Hilfe-Zeiger bzw. über Shift+F1.
- Die funktionsübergreifenden Inhalte der Online-Hilfe basieren auf dem HTML-Format; sie sind über den Menübefehl **Hilfe > STEP 7 Lite Hilfe** zu erreichen.

Zusätzlich zur Online-Hilfe stehen Ihnen elektronische Handbücher in PDF-Format zur Verfügung. Diese Handbücher finden Sie über die Taskleiste unter **Start > Simatic > Dokumentation**.

Über Ihre Siemens-Ansprechpartner können Sie darüber hinaus, wie gewohnt, die Dokumentation in Papierform beziehen.

Mehr Informationen zur Nutzung der Dokumentation entnehmen Sie bitte dem Kapitel "Hinweise zur Dokumentation" in der Datei LIESMICH.WRI auf Ihrer STEP 7 Lite-CD. Diese Datei enthält auch eventuelle Änderungen zur Online-Hilfe und zu den elektronischen Handbüchern, die bei Redaktionsschluss noch nicht feststanden.

1.4 Installation

1.4.1 Automation License Manager

1.4.1.1 Nutzungsberechtigung durch den Automation License Manager

Automation License Manager

Für die Nutzung der Programmiersoftware STEP 7 Lite wird eine produktspezifische Autorisierung (Nutzungsberechtigung) benötigt, dessen Installation ab der V3.0 von STEP 7 Lite mit dem Automation License Manager durchgeführt wird.

Der Automation License Manager ist ein Software-Produkt der Siemens AG. Er wird systemübergreifend zur Handhabung von License Keys (technische Repräsentanten von Lizenzen) eingesetzt.

Den Automation License Manager finden Sie:

- auf dem Installationsmedium des jeweiligen Software-Produktes, für den ein License Key benötigt wird, oder
- auf einem separaten Installationsmedium sowie
- auf den Internetseiten des A&D Customer Support der Siemens AG als WebDownload.

Im Automation License Manager ist eine Online-Hilfe integriert, die Sie nach der Installation kontextsensitiv über die F1-Taste oder über den Menübefehl **Hilfe > Hilfe zum License Manager** aufrufen können. In dieser Hilfe erhalten Sie detaillierte Informationen zur Funktionalität und Handhabung des Automation License Managers.

Lizenzen

Für die Nutzung von lizenzrechtlich geschützten Programmpaketen von STEP 7 Lite werden Lizenzen benötigt. Eine Lizenz wird als Recht zur Nutzung von Produkten vergeben. Die Repräsentanten dieses Rechtes sind:

- das CoL (**Certificate of License**) und
- der License Key.

Certificate of License (CoL)

Das im Lieferumfang der jeweiligen Produkte enthaltene "Certificate of License" ist der juristische Nachweis des Nutzungsrechtes. Das jeweilige Produkt darf nur durch den Besitzer des CoL oder beauftragte Personen genutzt werden.

License Keys

Der License Key ist der technische Repräsentant einer Lizenz (elektronischer Lizenzstempel).

Für jede Software, die lizenzrechtlich geschützt ist, wird von der SIEMENS AG ein License Key vergeben. Erst wenn nach dem Starten der Software auf einem Rechner das Vorhandensein eines gültigen License Keys festgestellt wurde, kann die jeweilige Software entsprechend der mit diesem License Key verbundenen Lizenz- und Nutzungsbedingungen genutzt werden.

Hinweise

- Sie können die Software von STEP 7 Lite zum kurzen kennen lernen von Bedienoberfläche und Funktionsumfang auch ohne License Key verwenden.
 - Die uneingeschränkte Nutzung unter Berücksichtigung der lizenzrechtlichen Vereinbarungen ist jedoch nur mit installiertem License Key zulässig und möglich.
 - Wenn Sie den License Key **nicht** installiert haben, werden Sie in regelmäßigen Abständen aufgefordert, die Installation vorzunehmen.
-

License Keys können wie folgt abgelegt sein und zwischen den einzelnen Speichermedien transferiert werden:

- auf License Key-Disketten,
- auf lokalen Festplattenspeichern und
- auf Festplattenspeichern von Rechnern im Netzwerk.

Wenn Software-Produkte installiert sind, für die kein License Key verfügbar ist, so können Sie den License Key-Bedarf ermitteln und anschließend die notwendigen Lizenzen bestellen.

Weiterführende Informationen über die Handhabung von License Keys entnehmen Sie bitte der Online-Hilfe zum Automation License Manager.

Lizenz-Typen

Für Software-Produkte der Siemens AG wird zwischen folgenden anwendungsorientierten Lizenz-Typen unterschieden. Das Verhalten der Software wird durch die für diese Lizenz-Typen unterschiedlichen License Keys gesteuert. Die Art der Nutzung ergibt sich aus dem jeweiligen Certificate of License.

Lizenz-Typ	Beschreibung
Single License	Die Nutzung der Software ist zeitlich unbegrenzt auf einem beliebigen Rechner zulässig.
Floating License	Zeitlich unbegrenzte, auf Nutzung über ein Netzwerk bezogenes Nutzungsrecht ("remote"Nutzung) einer Software.
Trial License	Die Nutzung der Software ist beschränkt auf: <ul style="list-style-type: none"> • eine Gültigkeit von maximal 14 Tagen, • eine bestimmte Anzahl von Tagen ab der Erstnutzung, • die Nutzung für Tests und zur Validierung (Haftungsausschluss).
Upgrade License	Für ein Upgrade können spezifische Anforderungen an den Systemzustand gefordert sein: <ul style="list-style-type: none"> • Mit einer Upgrade License kann eine Lizenz einer "Alt-"Version x auf eine Version >x+... umgestellt werden. • Ein Upgrade kann z. B. durch eine Erweiterung des Mengengerüsts notwendig sein.

1.4.1.2 Installieren des Automation License Managers

Der Automation License Manager wird über ein MSI-Setup installiert. Die Installations-Software für den Automation License Manager finden Sie auf der STEP 7 Lite-Produkt-CD.

Sie können den Automation License Manager im Zusammenhang mit STEP 7 Lite oder erst zu einem späteren Zeitpunkt installieren.

Hinweise

- Detaillierte Informationen zur Vorgehensweise beim Installieren des Automation License Managers entnehmen Sie bitte der aktuellen Liesmich.wri.
- In der Online-Hilfe zum Automation License Manager erhalten Sie alle benötigten Informationen zur Funktionalität und Handhabung von License Keys.

License Keys später installieren

Wenn Sie die STEP 7 Lite-Software starten und keine License Keys vorhanden sind, so erhalten Sie eine entsprechende Meldung.

Hinweise

- Sie können die Software von STEP 7 Lite zum kurzen kennen lernen von Bedienoberfläche und Funktionsumfang auch ohne License Key verwenden.
 - Die uneingeschränkte Nutzung unter Berücksichtigung der lizenzrechtlichen Vereinbarungen ist jedoch nur mit installiertem License Key zulässig und möglich.
 - Wenn Sie den License Key **nicht** installiert haben, werden Sie in regelmäßigen Abständen aufgefordert, die Installation vorzunehmen.
-

Zum nachträglichen Installieren von License Keys haben Sie folgende Möglichkeiten:

- Installieren der License Keys von Disketten
- Installieren der License Keys über WebDownLoad (vorherige Bestellung erforderlich)
- Nutzung von im Netzwerk vorhandenen Floating License Keys.

Detaillierte Informationen zur Vorgehensweise entnehmen Sie bitte der Online-Hilfe zum Automation License Manager, die Sie nach der Installation kontextsensitiv über die F1-Taste oder über den Menübefehl **Hilfe > Hilfe zum License Manager** aufrufen können.

Hinweise

- License Keys sind unter Windows 2000/XP nur dann funktionsfähig, wenn sie auf einem Festplattenlaufwerk liegen, auf dem schreibende Zugriffe zugelassen sind.
 - Floating Licenses können auch über ein Netzwerk, also "remote" genutzt werden.
-

1.4.1.3 Regeln für den Umgang mit License Keys



Vorsicht

Beachten Sie die Hinweise zum Umgang mit License Keys, die in der Online-Hilfe zum Automation License Manager beschrieben sind und in der STEP 7 Lite-Liesmich.wri auf der CD-ROM. Bei Nichtbeachtung besteht die Gefahr, dass License Keys unwiderruflich verloren gehen.

Die Online-Hilfe zum Automation License Manager können Sie kontextsensitiv über die F1-Taste oder über den Menübefehl **Hilfe > Hilfe zum Automation License Manager** aufrufen.

In dieser Hilfe erhalten Sie alle benötigten Informationen zur Funktionalität und Handhabung von License Keys.

1.4.2 Installieren von STEP 7 Lite

STEP 7 Lite enthält ein Setup-Programm, das die Installation automatisch durchführt. Eingabeaufforderungen auf dem Bildschirm führen Sie Schritt für Schritt durch den gesamten Installationsvorgang. Es wird mit der unter Windows üblichen Standardprozedur zur Installation von Software aufgerufen.

Die wesentlichen Phasen der Installation sind:

- das Kopieren der Daten auf Ihr Erstellsystem,
- das Eintragen der Ident-Nr.,
- das Einrichten der Treiber für die Kommunikation,
- das Installieren der License Keys (falls gewünscht).

Installationsvoraussetzungen

- Das Paket läuft unter dem Betriebssystem
 - Microsoft Windows XP Home
 - Microsoft Windows XP Professional
 - Microsoft Windows 2000
- Basishardware:
PC oder Programmiergerät mit den für das Betriebssystem empfohlenen Systemanforderungen. Die Systemanforderungen des Betriebssystems entnehmen Sie der Dokumentation des Betriebssystems bzw. den Microsoft Internetseiten.

Ein Programmiergerät (PG) ist ein Personal Computer in spezieller industrietauglicher und kompakter Ausführung. Es ist komplett ausgestattet für die Programmierung der SIMATIC-Automatisierungssysteme.

- Speicherkapazität
Erforderlicher Speicherplatz auf der Festplatte siehe "LIESMICH.WRI".
- MPI-Schnittstelle (optional)
Die MPI-Schnittstelle zwischen Erstellsystem (Programmiergerät oder PC) und Zielsystem ist nur erforderlich, wenn Sie unter STEP 7 Lite über MPI mit dem Zielsystem kommunizieren wollen.
Dazu verwenden Sie entweder
 - einen PC-Adapter und ein Null-Modem-Kabel (RS232), die an die Kommunikationsschnittstelle Ihres Geräts angeschlossen werden oder
 - eine MPI-Baugruppe (z. B. CP 5611), die in Ihrem Gerät installiert wird.

Bei Programmiergeräten ist die MPI-Schnittstelle bereits eingebaut.

Hinweise

Beachten Sie auch die Hinweise zum Installieren von STEP 7 Lite in der Datei LIESMICH.WRI.

Die Liesmich-Datei finden Sie in der Startleiste unter **Start > Simatic > Produkt-Hinweise**.

1.4.2.1 Vorgehen beim Installieren

Vorbereitungen

Bevor Sie mit der Installation beginnen können, muss das Windows-Betriebssystem (Windows 2000 oder XP) gestartet sein.

- Sie benötigen keine externen Datenträger, wenn sich die installierbare STEP 7 Lite-Software bereits auf der Festplatte des PG befindet.
- Um von CD-ROM zu installieren, legen Sie die CD-ROM in das CD-ROM-Laufwerk Ihres PC.

Starten des Installationsprogramms

Gehen Sie zur Installation folgendermaßen vor:

1. Legen Sie die CD-ROM ein und starten Sie das Setup durch Doppelklick auf die Datei "setup.exe".
2. Befolgen Sie Schritt für Schritt die Anweisungen, die Ihnen das Installationsprogramm anzeigt.

Das Programm führt Sie schrittweise durch den Installationsprozess. Sie können jeweils zum nachfolgenden oder vorhergehenden Schritt weiterschalten.

Während des Installationsvorgangs werden Ihnen in Dialogfeldern Fragen angezeigt oder Optionen zur Auswahl angeboten. Lesen Sie bitte die folgenden Hinweise, um die Dialoge schneller und leichter beantworten zu können.

Falls schon eine STEP 7 Lite-Version installiert ist ...

Wenn das Installationsprogramm feststellt, dass sich bereits eine STEP 7 Lite-Installation auf dem Erstellsystem befindet, wird eine entsprechende Meldung angezeigt, und Sie haben folgenden Wahlmöglichkeiten:

- Installation abbrechen (um danach alte STEP 7 Lite-Version unter Windows zu deinstallieren und anschließend die Installation erneut zu starten) oder
- Installation fortsetzen und damit alte Version durch neue Version überschreiben.

Hinsichtlich einer sauberen Softwarepflege sollten Sie vor einer Installation eine eventuell vorhandene ältere Version deinstallieren. Das einfache Überschreiben einer älteren Version hat außerdem den Nachteil, dass bei einem anschließenden Deinstallieren die eventuell noch vorhandenen Teile aus einer älteren Installation nicht entfernt werden.

Zum Installationsumfang

Zur Festlegung des Installationsumfangs haben Sie drei Auswahlmöglichkeiten:

- Maximalkonfiguration: alle Sprachen der Benutzungsoberfläche. Den dazu benötigten Speicherplatz entnehmen Sie bitte der aktuellen Produktinformation.
- Minimalkonfiguration: nur eine Sprache. Den dazu benötigten Speicherplatz entnehmen Sie bitte der aktuellen Produktinformation.
- Benutzerdefiniert: Sie können den Installationsumfang wählen hinsichtlich Programme, Datenbasis, etc.

Zur Ident-Nummer

Sie werden bei der Installation nach einer Ident-Nummer gefragt. Tragen Sie die Ident-Nummer ein. Sie finden diese auf dem Software-Produktschein oder auf der zugehörigen Autorisierungsdiskette.

Zum Installieren von License Keys

Bei der Installation wird überprüft, OB ein entsprechender License Key auf der Festplatte vorhanden ist. Wird kein gültiger License Key erkannt, so erscheint ein Hinweis, dass die Software nur mit vorhandenem License Key benutzt werden kann. Wenn Sie es wünschen, können Sie gleich die License Keys installieren oder aber die Installation von STEP 7 Lite fortsetzen und die License Keys zu einem späteren Zeitpunkt nachinstallieren. Im erstgenannten Fall legen Sie die mitgelieferte License Key-Diskette ein, wenn Sie dazu aufgefordert werden.

Zur PG/PC-Schnittstelleneinstellung

Während des Installationsvorgangs wird ein Dialog zur Einstellung der PG/PC-Schnittstelle angezeigt. Lesen Sie dazu "Einstellen der PG/PC-Schnittstelle".

Fehler während der Installation

Folgende Fehler führen zum Abbruch der Installation:

- Wenn sofort nach dem Start des Setups ein Initialisierungsfehler auftritt, so wurde höchstwahrscheinlich *setup* nicht unter Windows gestartet.
- Speicherplatz reicht nicht aus: Sie benötigen, abhängig vom Installationsumfang, ca. 100 MB freien Speicherplatz auf Ihrer Festplatte für die Basissoftware.
- Defekte CD: Wenn Sie feststellen, dass eine CD defekt ist, wenden Sie sich bitte an Ihre Siemens-Vertretung.
- Bedienungsfehler: Beginnen Sie die Installation erneut und beachten Sie die Anweisungen sorgfältig.

Zum Abschluss der Installation ...

Wenn die Installation erfolgreich war, wird dies durch eine entsprechende Meldung auf dem Bildschirm angezeigt.

Falls bei der Installation Systemdateien aktualisiert wurden, werden Sie aufgefordert, Windows neu zu starten. Nach dem Neustart (Warmstart) können Sie STEP 7 Lite starten.

Nach einer erfolgreichen Installation ist eine Programmgruppe für STEP 7 Lite eingerichtet.

1.4.2.2 Einstellen der PG/PC-Schnittstelle

Mit den hier gemachten Einstellungen legen Sie die Kommunikation zwischen PG/PC und dem Automatisierungssystem fest. Während des Installationsvorgangs wird ein Dialog zur Einstellung der PG/PC-Schnittstelle angezeigt. Sie können sich den Dialog auch nach der Installation anzeigen lassen, indem Sie das Programm "PG/PC-Schnittstelle einstellen" in der STEP 7 Lite-Programmgruppe oder der Systemsteuerung aufrufen. Dadurch ist es möglich, die Schnittstellenparameter auch später unabhängig von einer Installation zu ändern.

Prinzipielles Vorgehen

Zum Betrieb einer Schnittstelle sind erforderlich:

- Einstellungen im Betriebssystem
- eine geeignete Schnittstellenparametrierung

Wenn Sie ein PG über eine MPI-Verbindung einsetzen, so sind keine weiteren betriebssystemspezifischen Anpassungen notwendig.

Wenn Sie einen PC mit MPI-Karte oder Kommunikationsprozessoren (CP) einsetzen, so sollten Sie über die "Systemsteuerung" von Windows die Interrupt- und Adressbelegung prüfen, um sicherzustellen, dass es zu keinen Interrupt-Konflikten oder Adressbereichsüberschneidungen kommt.

Um die Parametrierung der PG/PC-Schnittstelle zu vereinfachen, werden Ihnen in Dialogfeldern vordefinierte Sätze von Grundparametern (Schnittstellenparametrierungen) zur Auswahl angeboten.

Parametrieren der PG/PC-Schnittstelle

Gehen Sie folgendermaßen vor (ausführliche Beschreibung in der Online-Hilfe):

1. Doppelklicken Sie in der "Systemsteuerung" auf "PG/PC-Schnittstelle einstellen".
2. Stellen Sie den "Zugangspunkt der Applikation" auf "S7ONLINE".
3. Wählen Sie in der Liste "Benutzte Schnittstellenparametrierung" die gewünschte Schnittstellenparametrierung aus. Wird die von Ihnen gewünschte Schnittstellenparametrierung nicht angezeigt, müssen Sie zunächst über die Schaltfläche "Auswählen" eine Baugruppe bzw. ein Protokoll installieren. Die Schnittstellenparametrierung wird dann automatisch erstellt.

- Wenn Sie eine Schnittstelle **mit automatischer Erkennung der Busparameter** wählen (z. B. CP 5611 (Auto)), dann können Sie das PG bzw. den PC an MPI bzw. PROFIBUS anschließen, ohne Busparameter einstellen zu müssen. Bei Übertragungsgeschwindigkeiten kleiner 187,5 kbit/s können allerdings Wartezeiten bis zu einer Minute entstehen. **Voraussetzung für die automatische Erkennung:** Am Bus sind Master angeschlossen, die zyklisch Busparameter verteilen. Alle neuen MPI-Komponenten tun das; bei PROFIBUS-Subnetzen darf das zyklische Verteilen der Busparameter nicht ausgeschaltet sein (voreingestellte PROFIBUS-Netzeinstellung).
- Wenn Sie eine Schnittstelle **ohne automatische Erkennung der Busparameter** wählen, dann lassen Sie sich die Eigenschaften anzeigen und passen sie an das Subnetz an.

Änderungen sind auch dann erforderlich, wenn sich Konflikte mit anderen Einstellungen ergeben (z. B. Interrupt- oder Adressbelegungen). Nehmen Sie in diesem Fall die entsprechenden Änderungen mit der Hardwareerkennung und der Systemsteuerung von Windows vor (siehe unten).



Vorsicht

Eine eventuell angezeigte Schnittstellenparametrierung "TCP/IP" **nicht** entfernen!
Dies könnte den Ablauf von anderen Anwendungen beeinträchtigen.

Prüfen der Interrupt- und Adressbelegung

Wenn Sie einen PC mit MPI-Karte einsetzen, so sollten Sie in jedem Fall prüfen, OB der voreingestellte Interrupt und der voreingestellte Adressbereich frei sind und ggf. einen freien Interrupt und/oder Adressbereich wählen.

Windows 2000

Unter Windows 2000 können Sie sich die Ressourcen unter **Systemsteuerung > Verwaltung > Computerverwaltung > System > Systeminformationen > Hardwareressourcen** anzeigen lassen.

Windows XP

Unter Windows XP können Sie sich die Ressourcen unter **START > Alle Programme > Zubehör > System > Systemprogramme > Systeminformationen > Hardwareressourcen** anzeigen lassen.

1.4.3 Deinstallieren von STEP 7 Lite

Benutzen Sie das unter Windows übliche Verfahren zur Deinstallation:

1. Starten Sie unter Windows den Dialog zu Installation von Software durch Doppelklick auf das Symbol "Software" in "Systemsteuerung".
2. Markieren Sie den STEP 7 Lite-Eintrag in der angezeigten Liste der installierten Software. Betätigen Sie dann die Schaltfläche zum "Entfernen" der Software.
3. Falls Dialogfelder "Freigegebene Datei entfernen" erscheinen, so klicken Sie im Zweifelsfall auf die Schaltfläche "Nein".

2 Grundlagen zum Entwerfen eines Programms

2.1 Programme in einer CPU

In einer CPU laufen grundsätzlich zwei verschiedene Programme ab:

- das Betriebssystem und
- das Anwenderprogramm.

Betriebssystem

Das Betriebssystem ist bereits in jeder CPU enthalten und organisiert alle Funktionen und Abläufe der CPU, die nicht mit einer spezifischen Steuerungsaufgabe verbunden sind. Zu seinen Aufgaben gehören:

- das Abwickeln des Anlaufs
- das Aktualisieren des Prozessabbildes der Eingänge und die Ausgabe des Prozessabbildes der Ausgänge
- das Aufrufen des Anwenderprogramms
- das Erfassen von Alarmen und das Aufrufen der Alarm-OBs
- das Erkennen und Behandeln von Fehlern
- das Verwalten von Speicherbereichen
- das Kommunizieren mit Programmiergeräten und anderen Kommunikationspartnern

Durch Ändern der Betriebssystemparameter (Voreinstellungen des Betriebssystems) können Sie das Verhalten der CPU in bestimmten Bereichen beeinflussen

Anwenderprogramm

Das Anwenderprogramm müssen Sie erstellen und in die CPU laden. Es enthält alle Funktionen, die zur Bearbeitung Ihrer spezifischen Automatisierungsaufgabe erforderlich sind. Zu den Aufgaben des Anwenderprogramms gehören:

- das Festlegen der Voraussetzungen für den Anlauf der CPU (z. B. Signale mit einem bestimmten Wert vorbesetzen)
- das Bearbeiten von Prozessdaten (z. B. Binärsignale verknüpfen, Analogwerte einlesen und auswerten, Binärsignale für die Ausgabe festlegen, Analogwerte ausgeben)
- das Reagieren auf Alarme
- das Bearbeiten von Störungen im normalen Programmablauf.

2.2 Bausteine im Anwenderprogramm

Die Programmiersoftware STEP 7 Lite bietet Ihnen die Möglichkeit, das Anwenderprogramm zu strukturieren, d. h. in einzelne, in sich geschlossene Programmabschnitte aufzuteilen. Daraus ergeben sich die folgenden Vorteile:

- umfangreiche Programme lassen sich übersichtlich programmieren
- einzelne Programmteile können standardisiert werden
- die Programmorganisation wird vereinfacht
- Änderungen des Programms lassen sich leichter durchführen
- der Programmtest wird vereinfacht, weil er abschnittsweise erfolgen kann
- die Inbetriebnahme wird erleichtert

Bausteinarten

Es gibt verschiedene Arten von Bausteinen, die Sie innerhalb eines Anwenderprogramms verwenden können:

Baustein	Kurzbeschreibung der Funktion	Näheres siehe
Organisationsbausteine (OB)	OBs legen die Struktur des Anwenderprogramms fest.	"Organisationsbausteine und Programmstruktur"
Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC)	SFBs und SFCs sind in die S7-CPU integriert und machen Ihnen einige wichtige Systemfunktionen zugänglich.	"Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC)"
Funktionsbausteine (FB)	FBs sind Bausteine mit "Gedächtnis", die Sie selbst programmieren.	"Funktionsbausteine (FB)"
Funktionen (FC)	FCs enthalten ProgrammROUTINEN für häufig verwendete Funktionen.	"Funktionen (FC)"
Instanz-Datenbausteine (Instanz-DB)	Instanz-DBs werden bei Aufruf eines FB/SFB dem Baustein zugeordnet. Beim Übersetzen werden sie automatisch generiert.	"Instanz-Datenbausteine"
Datenbausteine (DB)	DBs sind Datenbereiche zur Speicherung von Anwenderdaten. Zusätzlich zu den Daten, die jeweils einem Funktionsbaustein zugeordnet sind, können globale Daten definiert und von beliebigen Bausteinen genutzt werden.	"Globale Datenbausteine (DB)"

OBs, FBs, SFBs, FCs und SFCs enthalten Programmteile und werden deshalb auch als Codebausteine bezeichnet. Die erlaubte Anzahl der Bausteine pro Bausteintyp und die erlaubte Länge der Bausteine ist CPU-abhängig.

2.2.1 Organisationsbausteine und Programmstruktur

Organisationsbausteine (OBs) bilden die Schnittstelle zwischen dem Betriebssystem und dem Anwenderprogramm. Sie werden vom Betriebssystem aufgerufen und steuern die zyklische und alarmgesteuerte Programmbearbeitung, das Anlaufverhalten des Automatisierungssystems und die Behandlung von Fehlern. Sie können die Organisationsbausteine programmieren und so das Verhalten der CPU bestimmen.

Priorität von Organisationsbausteinen

Organisationsbausteine bestimmen die Reihenfolge (Startereignisse), in der die einzelnen Programmteile bearbeitet werden. Die Bearbeitung eines OB kann durch den Aufruf eines anderen OB unterbrochen werden. Welcher OB einen anderen OB unterbrechen darf, hängt von seiner Priorität ab. Höherprioritäre OBs unterbrechen niederprioritäre OBs. Die niedrigste Priorität hat der Hintergrund-OB.

Alarmarten und Prioritätsklassen

Die Startereignisse, die den Aufruf eines bestimmten OB veranlassen, werden auch Alarme genannt. Nachfolgende Tabelle zeigt die Alarmarten bei STEP 7 Lite und die Priorität der zugeordneten Organisationsbausteine. Nicht alle angegebenen Organisationsbausteine und deren Prioritätsklassen sind in allen S7-CPUs vorhanden (siehe z. B. Handbuch *Automatisierungssystem S7-300, Aufbauen, CPU-Daten*).

Alarmart	Organisationsbaustein	Prioritätsklasse (voreingestellt)	Näheres siehe
Freier Zyklus	OB 1	1	"Organisationsbaustein für zyklische Programmbearbeitung (OB 1)"
Uhrzeitalarme	OB 10 bis OB 17	2	"Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17)"
Verzögerungsalarme	OB 20	3	"Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23)"
	OB 21	4	
	OB 22	5	
	OB 23	6	
Weckalarme	OB 30	7	"Organisationsbausteine für den Weckalarm (OB 30 bis OB 38)"
	OB 31	8	
	OB 32	9	
	OB 33	10	
	OB 34	11	
	OB 35	12	
	OB 36	13	
	OB 37	14	
	OB 38	15	
Prozessalarme	OB 40	16	"Organisationsbausteine für den Prozessalarm (OB 40 bis OB 47)"
	OB 41	17	
	OB 42	18	
	OB 43	19	
	OB 44	20	
	OB 45	21	
	OB 46	22	
OB 47	23		

Alarmart	Organisationsbaustein	Prioritätsklasse (voreingestellt)	Näheres siehe
Asynchrone Fehler	OB 80 Zeitfehler OB 82 Diagnosealarm OB 84 CPU-Hardwarefehler OB 85 Programmablauffehler OB 86 Baugruppenträgerausfall OB 87 Kommunikationsfehler	26 (bzw. 28, wenn der asynchrone Fehler-OB im Anlaufprogramm vorkommt)	"Organisationsbausteine für die Fehlerbearbeitung (OB 80 bis OB 87 / OB 121 bis OB 122)"
Anlauf	OB 100 Neustart (Warmstart) OB 102 Kaltstart	27 27	"Organisationsbausteine für den Anlauf (OB 100/OB 102)"
Synchrone Fehler	OB 121 Programmierfehler OB 122 Zugriffsfehler	Prioritätsklasse des fehlerverursachenden OB	"Organisationsbausteine für die Fehlerbearbeitung (OB 80 bis OB 87 / OB 121 bis OB 122)"
1) Der Prioritätsklasse 29 entspricht die Priorität 0.29. Der Hintergrundzyklus hat also eine niedrigere Priorität als der freie Zyklus.			

Verändern der Priorität

Die Priorität der Organisationsbausteine ist nicht änderbar.

Fehler-OBs, die bei synchronen Fehlern gestartet werden, werden in derselben Prioritätsklasse bearbeitet wie der Baustein, der beim Erkennen des Fehlers bearbeitet wurde.

Lokaldaten

Beim Erstellen von Codebausteinen (OBs, FCs, FBs) können Sie temporäre Lokaldaten vereinbaren. Der in der CPU zur Verfügung stehende Lokaldatenbereich wird unter den Prioritätsklassen aufgeteilt.

Startinformation eines OB

Jeder Organisationsbaustein verfügt über eine Startinformation von 20 Byte Lokaldaten, die das Betriebssystem beim Starten eines OB übergibt. Die Startinformation gibt Auskunft über das Startereignis des OB, Datum und Uhrzeit des OB-Starts, aufgetretene Fehler und Diagnoseereignisse.

Ein Prozessalarm-OB 40 enthält z. B. in der Startinformation die Adresse der alarmerzeugenden Baugruppe.

Abgewählte Alarm-OBs

Wählen Sie die Prioritätsklassen 0 oder ordnen Sie einer Prioritätsklasse weniger als 20 Byte Lokaldaten zu, wird der entsprechende Alarm-OB abgewählt.

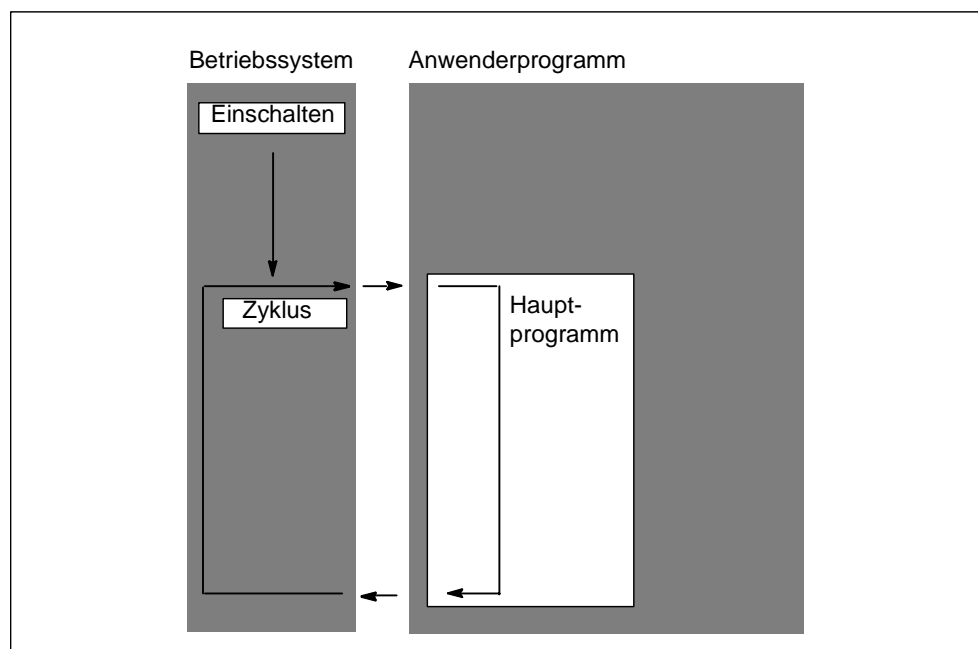
Abgewählte Alarm-OBs können:

- im Betriebszustand RUN nicht kopiert bzw. nicht in Ihr Anwenderprogramm eingebunden werden.
- im Betriebszustand STOP zwar kopiert bzw. in Ihr Anwenderprogramm eingebunden werden, führen jedoch beim Neustart (Warmstart) der CPU zum Abbruch des Anlaufs und erzeugen einen Eintrag in den Diagnosepuffer.

Durch das Abwählen nicht benötigter Alarm-OBs wird der zur Verfügung stehende freie Lokaldatenbereich größer und kann für die Speicherung temporärer Daten in anderen Prioritätsklassen genutzt werden.

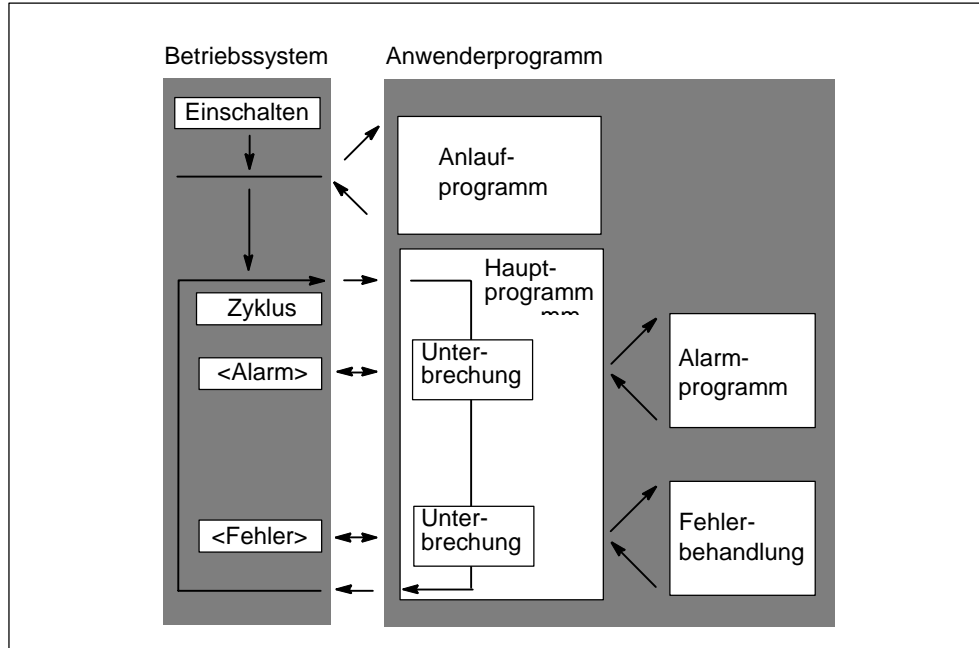
Zyklische Programmbearbeitung

Bei speicherprogrammierbaren Steuerungen herrscht typischerweise die zyklische Programmbearbeitung vor, d. h. das Betriebssystem läuft in einer Programmschleife (dem Zyklus) und ruft dabei in jeder Schleife einmal den Organisationsbaustein OB 1 im Hauptprogramm auf. Das Anwenderprogramm im OB 1 wird also zyklisch bearbeitet.



Ereignisgesteuerte Programmbearbeitung

Die zyklische Programmbearbeitung kann durch bestimmte Startereignisse (Alarme) unterbrochen werden. Tritt ein solches Startereignis ein, wird der gerade bearbeitete Baustein an einer Befehlsgränze unterbrochen und ein anderer Organisationsbaustein abgearbeitet, der dem Startereignis zugeordnet ist. Danach wird die Bearbeitung des zyklischen Programms an der Unterbrechungsstelle wieder fortgesetzt.

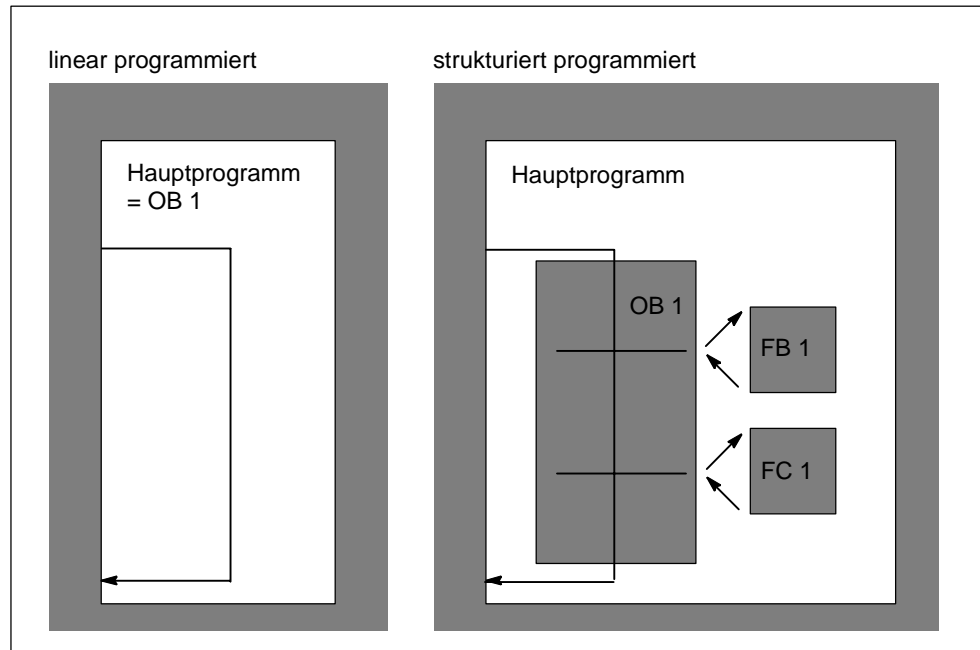


Damit besteht die Möglichkeit, Teile des Anwenderprogramms, die nicht zyklisch bearbeitet werden müssen, nur bei Bedarf zu bearbeiten. Das Anwenderprogramm kann in Teilprogramme zerlegt und auf verschiedene Organisationsbausteine aufgeteilt werden. Wenn das Anwenderprogramm z. B. auf ein wichtiges Signal reagieren soll, das relativ selten vorkommt (z. B. Grenzwertgeber zur Füllstandsmessung eines Tanks meldet Füllstand erreicht), dann sollte dieser Teil des Anwenderprogramms in einem OB stehen, der ereignisgesteuert bearbeitet wird.

Linear oder strukturiert programmieren

Sie können Ihr gesamtes Anwenderprogramm in den OB 1 schreiben (lineare Programmierung). Dies ist nur empfehlenswert bei einfachen Programmen, die auf S7-300-CPUs mit kleinem Speicherumfang laufen.

Komplexe Automatisierungsaufgaben können besser bearbeitet werden, wenn sie in kleinere Teilaufgaben untergliedert werden, die den technologischen Funktionen des Automatisierungsprozesses entsprechen oder mehrfach verwendet werden sollen. Im Anwenderprogramm werden diese Teilaufgaben durch entsprechende Programmteile, die Bausteine, repräsentiert (strukturierte Programmierung).



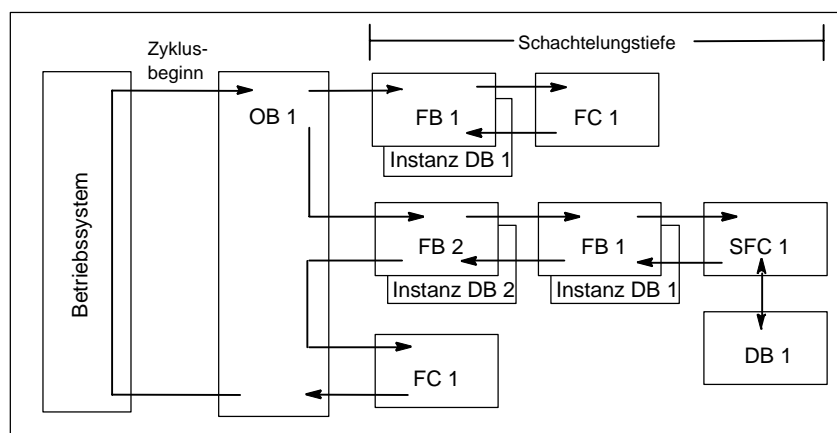
2.2.2 Aufrufhierarchie im Anwenderprogramm

Soll das Anwenderprogramm funktionieren, so müssen die Bausteine, aus denen es besteht, aufgerufen werden. Das geschieht durch spezielle STEP 7 Lite-Operationen, den Bausteinaufrufen, die nur innerhalb von Codebausteinen programmiert und gestartet werden können.

Reihenfolge und Schachtelungstiefe

Die Reihenfolge und Schachtelung der Bausteinaufrufe wird Aufrufhierarchie genannt. Die zulässige Schachtelungstiefe ist CPU-abhängig.

Nachfolgendes Bild zeigt anhand eines Beispiels die Reihenfolge und Schachtelung der Bausteinaufrufe innerhalb eines Bearbeitungszyklus.



Regeln für die Reihenfolge der Bausteinerstellung:

- Sie erstellen die Bausteine von oben nach unten, fangen also mit der obersten Bausteinreihe an.
- Jeder Baustein der aufgerufen wird, muss schon vorhanden sein, d.h. innerhalb einer Bausteinreihe ist die Erstellrichtung von rechts nach links.
- Als letztes erstellen Sie den OB 1.

Diese Regeln umgesetzt bedeuten für das Beispiel-Bild folgende Erstellreihenfolge:

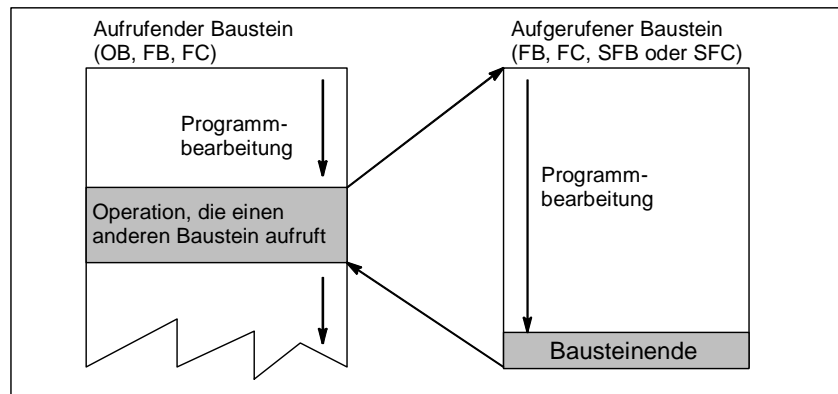
FC 1 > FB 1 + Instanz-DB 1 > DB 1 > SFC 1 > FB 2 + Instanz-DB 2 > OB 1

Achtung

Bei zu großer Schachtelungstiefe kann der Lokaldaten-Stack überlaufen (siehe auch Lokaldaten-Stack).

Bausteinaufrufe

Nachfolgendes Bild zeigt den Ablauf eines Bausteinaufrufs innerhalb eines Anwenderprogramms: das Programm ruft den zweiten Baustein auf, dessen Operationen dann vollständig bearbeitet werden. Ist die Bearbeitung des aufgerufenen Bausteins beendet, wird die Bearbeitung des aufrufenden Bausteins mit der dem Bausteinanruf folgenden Operation wiederaufgenommen.



Bevor Sie einen Baustein programmieren, müssen Sie erst festlegen, mit welchen Daten die Bearbeitung Ihres Programms erfolgen soll: Sie müssen die Variablen des Bausteins deklarieren.

Hinweis

OUT-Parameter müssen bei jedem Bausteinanruf beschrieben werden.

Achtung

Das Betriebssystem setzt die Instanzen des SFB 3 "TP" bei Kaltstart zurück. Falls Instanzen dieses SFB nach Neustart (Warmstart) initialisiert sein sollen, müssen Sie über den OB 100 die zu initialisierenden Instanzen des SFB mit PT = 0 ms aufrufen. Dies können Sie z. B. über eine Initialisierungsroutine in den Bausteinen erreichen, die Instanzen dieses SFB enthalten.

2.2.3 Zyklische Programmbearbeitung und CPU-Einstellungen

2.2.3.1 Organisationsbaustein für zyklische Programmbearbeitung (OB 1)

Die zyklische Programmbearbeitung ist die "normale" Programmbearbeitung bei speicherprogrammierbaren Steuerungen. Das Betriebssystem ruft zyklisch den OB 1 auf und startet damit die zyklische Bearbeitung des Anwenderprogramms.

Ablauf der zyklischen Programmbearbeitung

Nachfolgende Tabelle zeigt die Phasen der zyklischen Programmbearbeitung:

Schritt	Ablauf (bei CPUs ab 10/98)
1	Das Betriebssystem startet die Zyklusüberwachungszeit.
2	Die CPU schreibt die Werte aus dem Prozessabbild der Ausgänge in die Ausgabebaugruppen.
3	Die CPU liest den Zustand der Eingänge an den Eingabebaugruppen aus und aktualisiert das Prozessabbild der Eingänge.
4	Die CPU bearbeitet das Anwenderprogramm und führt die im Programm angegebenen Operationen aus.
5	Am Ende eines Zyklus führt das Betriebssystem anstehende Aufgaben aus, z. B. Laden und Löschen von Bausteinen, Empfangen und Senden von Globaldaten.
6	Anschließend kehrt die CPU zum Zyklusbeginn zurück und startet die Zykluszeitüberwachung erneut.

Prozessabbilder

Damit der CPU für die Dauer der zyklischen Programmbearbeitung ein konsistentes Abbild der Prozesssignale zur Verfügung steht, greift die CPU beim Ansprechen der Operandenbereiche Eingänge (E) und Ausgänge (A) nicht direkt auf die Signalbaugruppen zu, sondern auf einen internen Speicherbereich der CPU, in dem sich ein Abbild der Ein-/Ausgänge befindet.

Programmieren zyklischer Programmbearbeitung

Sie programmieren die zyklische Programmbearbeitung, indem Sie Ihr Anwenderprogramm mit STEP 7 Lite in den OB 1 und die darin aufgerufenen Bausteine schreiben.

Die zyklische Programmbearbeitung beginnt, sobald das Anlaufprogramm fehlerfrei beendet wurde.

Unterbrechungsmöglichkeiten

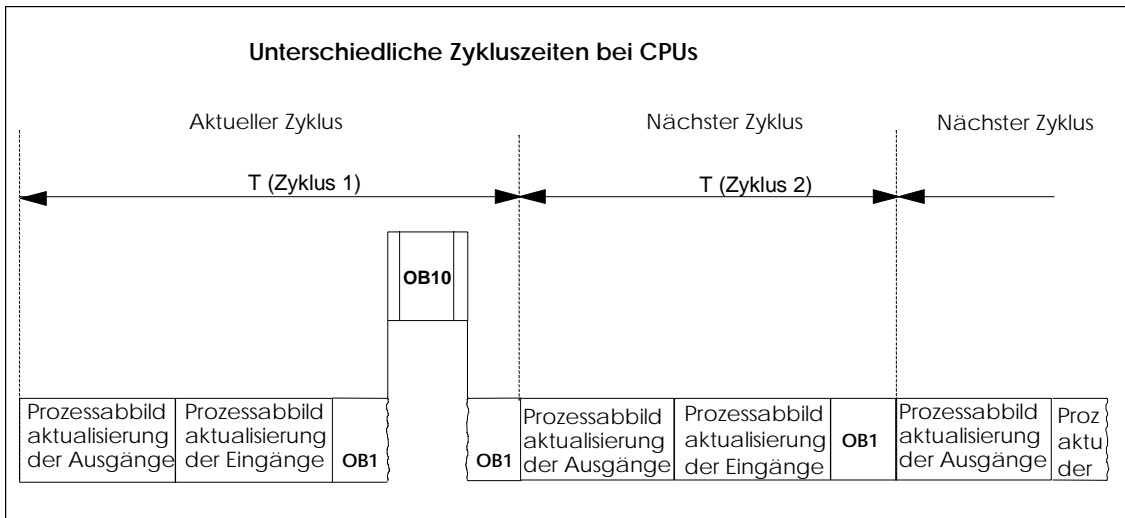
Die zyklische Programmbearbeitung kann unterbrochen werden durch:

- einen Alarm
- einen STOP-Befehl (Betriebsartenschalter, Menübefehl vom PG aus, SFC 46 STP, SFB 20 STOP)
- Netzspannungsausfall
- das Auftreten eines Geräte- oder Programmfehlers

Zykluszeit

Die Zykluszeit ist die Zeit, die das Betriebssystem für die Bearbeitung des zyklischen Programms sowie aller diesen Zyklus unterbrechenden Programmteile (z. B. Bearbeitung anderer Organisationsbausteine) und Systemtätigkeiten (z. B. Prozessabbildaktualisierung) benötigt. Diese Zeit wird überwacht.

Die Zykluszeit (T) ist nicht für jeden Zyklus gleich lang. Das nachfolgende Bild zeigt unterschiedliche Zykluszeiten für CPUs, die durch das Einschleichen eines Uhrzeit-Alarms OB 10 (unterbricht OB 1) verursacht werden.



Zyklusüberwachungszeit

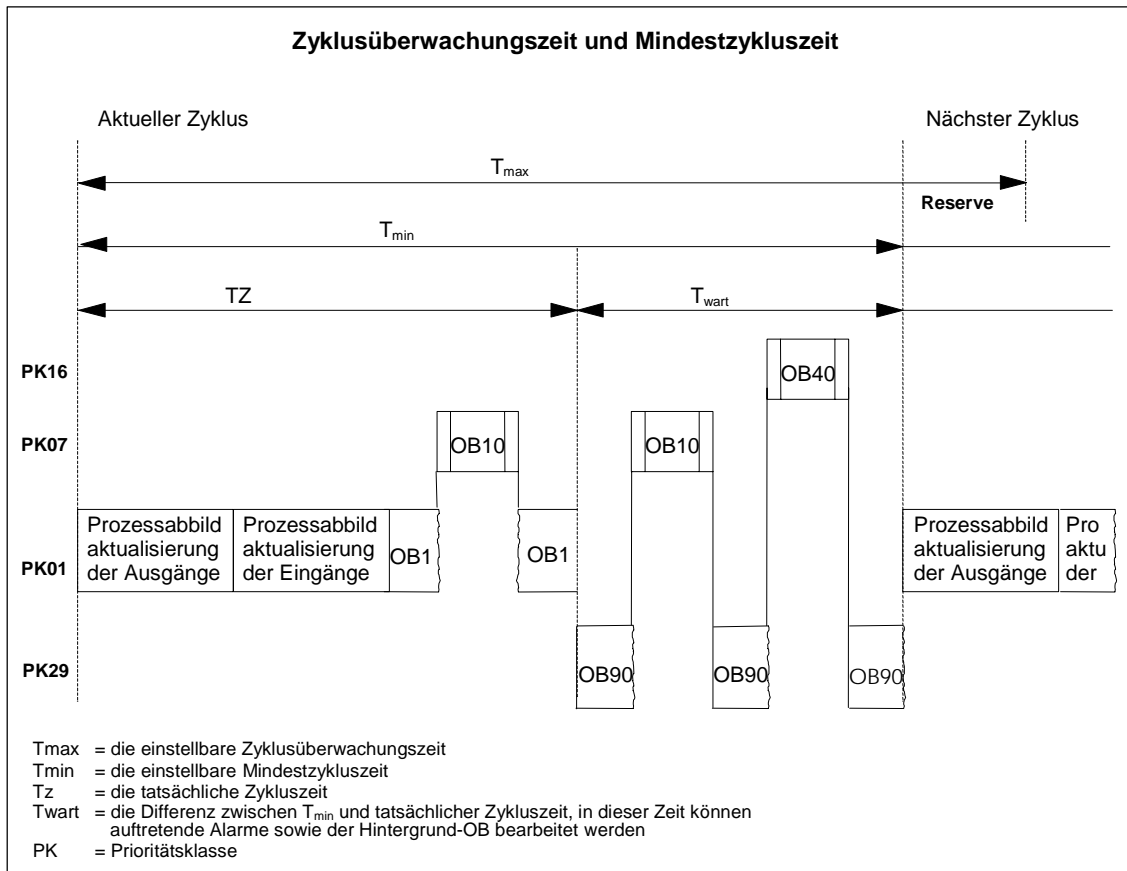
Sie können mit STEP 7 Lite die voreingestellte Zyklusüberwachungszeit ändern. Ist diese Zeit abgelaufen, geht entweder die CPU in STOP, oder es wird der OB 80 aufgerufen, in dem Sie festlegen können, wie die CPU auf den Zeitfehler reagieren soll.

Mindestzykluszeit

Sie können mit STEP 7 Lite eine Mindestzykluszeit einstellen, sofern die CPU diese Funktion unterstützt. Dies ist sinnvoll, wenn

- die Zeitabstände zwischen den Starts der Programmbearbeitung des OB 1 (freier Zyklus) gleich lang sein sollen oder
- bei zu kurzer Zykluszeit die Aktualisierung der Prozessabbilder unnötig oft erfolgen würde.

Das nachfolgende Bild zeigt die Funktion der Zyklusüberwachungszeit und Mindestzykluszeit im Programmablauf.



2.2.3.2 Kommunikationslast

Mit dem CPU-Parameter "Zyklusbelastung durch Kommunikation" können Sie die Dauer von Kommunikationsprozessen, die immer auch die Zykluszeit verlängern, in einem gewissen Rahmen steuern. Kommunikationsprozesse können z. B. sein: Datenübertragung zu einer anderen CPU über MPI oder Laden von Bausteinen, das über PG angestoßen wurde.

Testfunktionen mit dem PG werden durch diesen Parameter kaum beeinflusst. Sie können aber die Zykluszeit erheblich verlängern. Die für Testfunktionen zur Verfügung gestellte Zeit kann im Prozessbetrieb begrenzt werden.

Wirkungsweise des Parameters "Kommunikationslast"

Das Betriebssystem der CPU stellt laufend der Kommunikation den projektierten Prozentsatz der gesamten CPU-Verarbeitungsleistung zur Verfügung (Zeitscheiben-Technik). Wird diese Verarbeitungsleistung für die Kommunikation nicht benötigt, steht sie der übrigen Verarbeitung zur Verfügung.

Auswirkung auf die tatsächliche Zykluszeit

Ohne zusätzliche asynchrone Ereignisse verlängert sich die OB 1-Zykluszeit um einen Faktor, der sich nach folgender Formel berechnen lässt:

$$\frac{100}{100 - \text{"Zyklusbelastung durch Kommunikation (\%)\"}}$$

Beispiel 1 (keine zusätzlichen asynchrone Ereignisse):

Bei Einstellung der Zyklusbelastung durch Kommunikation auf 50 % kann sich eine Verdopplung der OB 1-Zykluszeit ergeben.

Gleichzeitig wird die OB 1-Zykluszeit auch noch durch asynchrone Ereignisse (z. B. Prozessalarme oder Weckalarme) beeinflusst. Durch die Verlängerung der Zykluszeit durch den Kommunikationsanteil treten statistisch gesehen auch mehr asynchrone Ereignisse innerhalb eines OB 1-Zyklus auf. Dies verlängert der OB 1-Zyklus zusätzlich. Diese Verlängerung ist abhängig davon, wieviele Ereignisse pro OB 1-Zyklus auftreten und der Dauer der Ereignisbearbeitung.

Beispiel 2 (zusätzlichen asynchrone Ereignisse berücksichtigt):

Bei einer reinen OB 1-Ablaufzeit von 500 ms kann sich durch eine Kommunikationslast von 50 % eine tatsächliche Zykluszeit von bis zu 1000 ms ergeben (unter der Voraussetzung, dass die CPU immer genügend Kommunikationsaufträge zum Bearbeiten hat). Wenn nun parallel dazu alle 100 ms ein Weckalarm mit 20 ms Bearbeitungszeit abläuft, dann würde sich dieser ohne Kommunikationslast mit insgesamt $5 \cdot 20 \text{ ms} = 100 \text{ ms}$ auf den Zyklus verlängernd auswirken, d. h. die tatsächliche Zykluszeit wäre 600 ms. Da ein Weckalarm auch die Kommunikation unterbricht, wirkt er sich bei 50% Kommunikationslast mit $10 \cdot 20 \text{ ms}$ auf die Zykluszeit aus, d. h. in diesem Fall beträgt die tatsächliche Zykluszeit nicht 1000 ms sondern 1200 ms.

Hinweise

- Überprüfen Sie die Auswirkungen einer Wertänderung des Parameters "Zyklusbelastung durch Kommunikation" im Anlagenbetrieb.
- Die Kommunikationslast muss beim Einstellen der minimalen Zykluszeit berücksichtigt werden, da es sonst zu Zeitfehlern kommt.

Empfehlungen

- Übernehmen Sie nach Möglichkeit den voreingestellten Wert.
- Vergrößern Sie den Wert nur dann, wenn die CPU hauptsächlich zu Kommunikationszwecken eingesetzt wird und das Anwenderprogramm zeitunkritisch ist!
- In allen anderen Fällen den Wert nur verringern!
- Stellen Sie Prozessbetrieb ein und begrenzen Sie die für Testfunktionen dort benötigte Zeit!

2.2.4 Alarmgesteuerte Programmbearbeitung

2.2.4.1 Organisationsbausteine für alarmgesteuerte Programmbearbeitung

Die S7-CPU's geben Ihnen durch die Bereitstellung von Alarm-OBs die Möglichkeit

- Programmteile zeitgesteuert zu bearbeiten
- auf externe Signale des Prozesses optimal zu reagieren.

Das zyklische Anwenderprogramm muss nicht ständig abfragen, OB Alarmereignisse eingetreten sind, sondern das Betriebssystem sorgt im Falle eines Alarms dafür, dass der Teil des Anwenderprogramms bearbeitet wird, der im Alarm-OB steht und festlegt, wie das Automatisierungssystem bei diesem Alarm reagieren soll.

Alarmarten und Anwendungen

Nachfolgende Tabelle zeigt, wie die Alarmarten eingesetzt werden können.

Alarmart	Alarm-OBs	Anwendungsbeispiele
Uhrzeitalarm	OB 10 bis OB 17	Errechnen der Durchflussmenge eines Mischprozesses bei Schichtende.
Verzögerungsalarm	OB 20 bis OB 23	Steuern eines Lüfters, der nach dem Abschalten eines Motors noch 20 s laufen soll, bevor er abgeschaltet wird.
Weckalarm	OB 30 bis OB 38	Abtasten eines Signalpegels für eine regelungstechnische Anlage.
Prozessalarm	OB 40 bis OB 47	Melden, dass der maximale Füllstand eines Behälters erreicht ist.

2.2.4.2 Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17)

Die S7-CPU stellt Uhrzeitalarm-OBs zur Verfügung, die zu einem angegebenen Datum oder in bestimmten Zeitabständen bearbeitet werden können.

Uhrzeitalarme können ausgelöst werden:

- einmalig zu einem bestimmten Zeitpunkt (Absolutzeitangabe mit Datum)
- periodisch mit Angabe des Startzeitpunktes und der Wiederholrate (z. B. minütlich, stündlich, täglich).

Regeln für Uhrzeitalarme

Uhrzeitalarme können nur bearbeitet werden, wenn ein Uhrzeitalarm parametrierbar ist und ein entsprechender Organisationsbaustein im Anwenderprogramm enthalten ist. Ist dies nicht der Fall, wird eine Fehlermeldung in den Diagnosepuffer eingetragen und asynchrone Fehlerbehandlung ausgeführt (OB 80, siehe "Organisationsbausteine für die Fehlerbearbeitung (OB 80 bis OB 87 / OB 121 bis OB 122)").

Periodische Uhrzeitalarme müssen einem realen Datum entsprechen. Die monatliche Wiederholung eines OB 10 mit Startzeit 31.1. ist nicht möglich. In diesem Fall würde der OB nur in den Monaten gestartet, die 31 Tage haben.

Ein Uhrzeit-Alarm, der während des Anlaufs (Neustart (Warmstart) oder Wiederanlauf) aktiviert wird, wird erst nach der Beendigung des Anlaufs bearbeitet.

Uhrzeitalarm-OBs, die durch Parametrierung ausgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Nach Neustart (Warmstart) müssen gestellte Uhrzeitalarme neu aktiviert werden (z. B. mit Hilfe der SFC 30 ACT_TINT im Anlaufprogramm).

Sonderfall: projektierter einmaliger Startzeitpunkt liegt in der Vergangenheit

Folgende Projektierung liegt vor:

- Ausführung: einmalig
- Aktiv: ja
- Startdatum/Uhrzeit: liegt in der Vergangenheit (bezogen auf Echtzeituhr der CPU)

Verhalten der CPU: Nach Kaltstart oder nach Neustart (Warmstart) wird der entsprechende Uhrzeitalarm-OB vom Betriebssystem **einmalig aufgerufen!**

Starten des Uhrzeitalarms

Damit die CPU einen Uhrzeitalarm starten kann, müssen Sie den Uhrzeitalarm erst stellen und dann aktivieren. Es gibt drei Startmöglichkeiten:

- automatischer Start des Uhrzeitalarms durch Parametrieren mit STEP 7 Lite (Parameterblock "Uhrzeitalarme")
- Stellen und Aktivieren des Uhrzeitalarms durch SFC 28 SET_TINT und SFC 30 ACT_TINT aus dem Anwenderprogramm heraus
- stellen des Uhrzeitalarms durch Parametrieren mit STEP 7 Lite und aktivieren des Uhrzeitalarms durch die SFC 30 ACT_TINT aus dem Anwenderprogramm heraus.

Abfragen des Uhrzeitalarms

Um abzufragen, OB und zu welchem Zeitpunkt Uhrzeitalarme gestellt sind, können Sie

- die SFC 31 QRY_TINT aufrufen oder
- die Teilliste "Alarmstatus" der Systemzustandsliste anfordern.

Deaktivieren des Uhrzeitalarms

Sie können Uhrzeitalarme, die noch nicht bearbeitet wurden, mit der SFC 29 CAN_TINT deaktivieren. Deaktivierte Uhrzeitalarme lassen sich mit der SFC 28 SET_TINT erneut stellen und mit der SFC 30 ACT_TINT aktivieren.

Priorität der Uhrzeitalarm-OBs

Alle acht Uhrzeitalarm-OBs haben in der Voreinstellung die gleiche Prioritätsklasse (2) und werden dementsprechend in der Reihenfolge ihres Startereignisses bearbeitet. Die Prioritätsklasse kann durch Parametrierung geändert werden.

Ändern der eingestellten Uhrzeit

Es gibt die folgenden Möglichkeiten, die eingestellte Uhrzeit zu ändern:

- ein Uhrzeitmaster synchronisiert die Uhrzeit für Master und Slaves
- im Anwenderprogramm wird durch SFC 0 SET_CLK die Uhrzeit neu gestellt

Verhalten bei Uhrzeitänderung

Nachfolgende Tabelle zeigt, wie sich Uhrzeitalarme nach dem Ändern der Uhrzeit verhalten.

Wenn...	dann...
durch das Vorstellen der Uhr ein oder mehrere Uhrzeitalarme übersprungen werden,	wird der OB 80 gestartet und in die Startinformation des OB 80 wird eingetragen, welche Uhrzeitalarme übersprungen wurden.
Sie im OB 80 die übersprungenen Uhrzeitalarme deaktiviert haben,	werden die übersprungenen Uhrzeitalarme nicht nachgeholt.
Sie im OB 80 die übersprungenen Uhrzeitalarme nicht deaktiviert haben,	wird der erste übersprungene Uhrzeitalarm nachgeholt, die anderen übersprungenen Uhrzeitalarme werden ignoriert.
durch das Rückstellen der Uhr bereits bearbeitete Uhrzeitalarme erneut anstehen,	wird die Bearbeitung dieser Uhrzeitalarme nicht wiederholt.

2.2.4.3 Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23)

Die S7-CPU's stellen Verzögerungsalarm-OBs zur Verfügung, mit deren Hilfe Sie die zeitverzögerte Bearbeitung von Teilen Ihres Anwenderprogramms programmieren können.

Regeln für Verzögerungsalarme

Verzögerungsalarme können nur bearbeitet werden, wenn sich ein entsprechender Organisationsbaustein im CPU-Programm befindet. Ist dies nicht der Fall, wird eine Fehlermeldung in den Diagnosepuffer eingetragen und asynchrone Fehlerbehandlung ausgeführt (OB 80, siehe "Organisationsbausteine für die Fehlerbearbeitung (OB 80 bis OB 87 / OB 121 bis OB 122)").

Verzögerungsalarm-OBs, die durch Parametrierung ausgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Verzögerungsalarme werden ausgelöst, wenn die in der SFC 32 SRT_DINT angegebene Verzögerungszeit abgelaufen ist.

Starten des Verzögerungsalarms

Um einen Verzögerungsalarm zu starten, müssen Sie in der SFC 32 die Verzögerungszeit festlegen, nach deren Ablauf der entsprechende Verzögerungsalarm-OB aufgerufen werden soll. Die maximal zulässige Länge der Verzögerungszeit entnehmen Sie bitte dem Handbuch *Automatisierungssystem S7-300, Aufbauen, CPU-Daten*.

Priorität der Verzögerungsalarm-OBs

In der Voreinstellung haben die Verzögerungsalarm-OBs die Prioritätsklassen 3 bis 6. Die Prioritätsklassen können durch Parametrierung geändert werden.

2.2.4.4 Organisationsbausteine für den Weckalarm (OB 30 bis OB 38)

Die S7-CPU's stellen Weckalarm-OBs zur Verfügung, die die zyklische Programmbearbeitung in bestimmten Abständen unterbrechen.

Weckalarme werden in bestimmten Zeitintervallen ausgelöst. Startzeitpunkt des Zeittaktes ist der Betriebszustandswechsel von STOP in RUN.

Regeln für Weckalarme

Achten Sie beim Vorgeben der Zeittakte darauf, dass zwischen den Startereignissen der einzelnen Weckalarme genügend Zeit für die Bearbeitung der Weckalarme bleibt.

Weckalarm-OBs, die durch Parametrierung abgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Starten des Weckalarms

Um einen Weckalarm zu starten, müssen Sie mit STEP 7 Lite im Parameterblock "Weckalarme" einen Zeittakt vorgeben. Der Zeittakt ist immer ein ganzzahliges Vielfaches des Grundtaktes von 1 ms.

$$\text{Zeittakt} = n \times \text{Grundtakt } 1 \text{ ms}$$

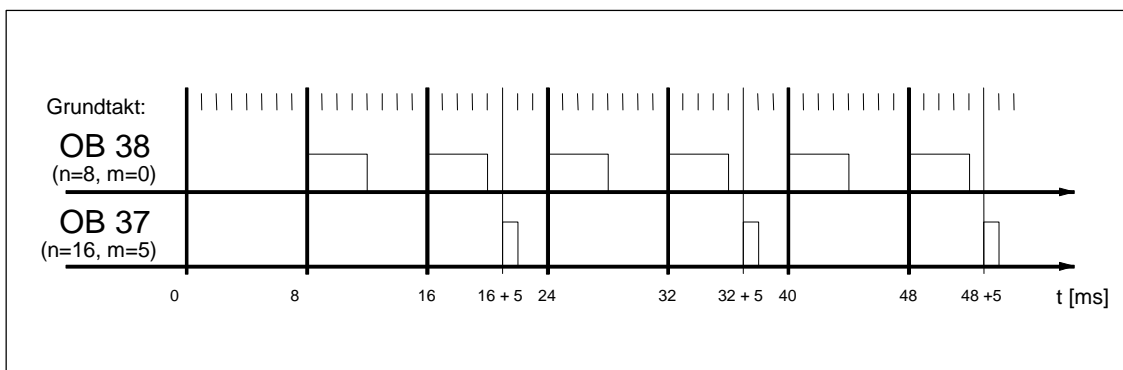
Die neun zur Verfügung stehenden Weckalarm-OBs geben in ihrer Voreinstellung Zeittakte vor (siehe nachfolgende Tabelle). Der Defaultzeittakt wird wirksam, wenn der ihm zugeordnete Weckalarm-OB geladen ist. Sie können jedoch durch Parametrierung die voreingestellten Werte ändern. Die Obergrenze entnehmen Sie bitte dem Handbuch *Automatisierungssystem S7-300, Aufbauen, CPU-Daten*.

Phasenverschiebung bei Weckalarmen

Um zu verhindern, dass die Weckalarme verschiedener Weckalarm-OBs zum gleichen Zeitpunkt eine Startaufforderung erhalten und dadurch möglicherweise ein Zeitfehler (Zykluszeitüberschreitung) entsteht, haben Sie die Möglichkeit, eine Phasenverschiebung vorzugeben. Die Phasenverschiebung sorgt dafür, dass die Bearbeitung eines Weckalarms nach Ablauf des Zeittaktes um einen bestimmten Zeitraum verschoben wird.

$$\text{Phasenverschiebung} = m \times \text{Grundtakt (mit } 0 \leq m < n)$$

Nachfolgendes Bild zeigt die Bearbeitung eines Weckalarm-OBs mit Phasenverschiebung (OB 37) im Gegensatz zu einem Weckalarm ohne Phasenverschiebung (OB 38).



Priorität der Weckalarm-OBs

Nachfolgende Tabelle zeigt die voreingestellten Zeittakte und Prioritätsklassen der Weckalarm-OBs. Zeittakt und Prioritätsklasse können durch Parametrierung geändert werden.

Weckalarm-OBs	Zeittakt in ms	Prioritätsklasse
OB 30	5000	7
OB 31	2000	8
OB 32	1000	9
OB 33	500	10
OB 34	200	11
OB 35	100	12
OB 36	50	13
OB 37	20	14
OB 38	10	15

2.2.4.5 Organisationsbausteine für den Prozessalarm (OB 40 bis OB 47)

Die S7-CPU's stellen Prozessalarm-OBs zur Verfügung, die auf Signale aus den Baugruppen (z. B. Signalbaugruppen SMS, Kommunikationsprozessoren CPs, Funktionsbaugruppen FMs) reagieren. Mit STEP 7 Lite können Sie für parametrierbare Digital- und Analogbaugruppen einstellen, welches Signal den OB starten soll. Bei CPs und FMs verwenden Sie hierzu die entsprechenden Parametriermasken.

Prozessalarme werden ausgelöst, wenn eine prozessalarmfähige Signalbaugruppe mit parametrierter Prozessalarmfreigabe ein empfangenes Prozesssignal an die CPU weiterleitet oder eine Funktionsbaugruppe der CPU einen Alarm meldet.

Regeln für Prozessalarme

Prozessalarme können nur bearbeitet werden, wenn sich ein entsprechender Organisationsbaustein im CPU-Programm befindet. Ist dies nicht der Fall, wird eine Fehlermeldung in den Diagnosepuffer eingetragen und asynchrone Fehlerbehandlung ausgeführt (siehe "Organisationsbausteine für die Fehlerbearbeitung (OB 80 bis OB 87 / OB 121 bis OB 122)").

Prozessalarm-OBs, die durch Parametrierung ausgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Parametrieren prozessalarmfähiger Signalbaugruppen

Jeder Kanal einer prozessalarmfähigen Signalbaugruppe kann einen Prozessalarm auslösen. Deshalb müssen Sie mit STEP 7 Lite in den Parametersätzen prozessalarmfähiger Signalbaugruppen folgendes festlegen:

- Wodurch ein Prozessalarm ausgelöst werden soll;
- welcher Prozessalarm-OB bearbeitet werden soll (die Voreinstellung sieht für die Bearbeitung aller Prozessalarme den OB 40 vor).

Mit STEP 7 Lite aktivieren Sie die Prozessalarmgenerierung der Funktionsbaugruppen. Weitere Parameter weisen Sie in den Parametriermasken dieser Funktionsbaugruppen zu.

Priorität der Prozessalarm-OBs

In der Voreinstellung haben die Prozessalarm-OBs die Prioritätsklassen 16 bis 23. Die Prioritätsklassen können durch Parametrierung geändert werden.

2.2.4.6 Organisationsbausteine für den Anlauf (OB 100/OB 102)

Anlaufarten

Man unterscheidet die folgende Anlaufarten:

- Neustart (Warmstart)
- Kaltstart

Der folgenden Tabelle können Sie entnehmen, welcher OB das Betriebssystem im Anlauf jeweils aufruft.

Anlaufart	Zugehöriger OB
Neustart (Warmstart)	OB 100
Kaltstart	OB 102

Startereignisse für Anlauf-OBs

Die CPU führt einen Anlauf durch

- nach NETZ-EIN
- wenn Sie den Betriebsartenschalter von STOP "RUN"/"RUN-P" umschalten
- nach Anforderung durch eine Kommunikationsfunktion
- nach Synchronisierung im Multicomputing-Betrieb
- bei einem H-System nach dem Ankoppeln (nur auf Reserve-CPU)

Abhängig vom Startereignis, von der vorliegenden CPU und deren eingestellten Parametern wird der zugehörige Anlauf-OB (OB 100 oder OB 102) aufgerufen.

Anlaufprogramm

Sie können die Randbedingungen für das Anlaufverhalten (Initialisierungswerte für RUN, Anlaufwerte für Peripheriebaugruppen) Ihrer CPU bestimmen, indem Sie Ihr Programm für den Anlauf in den Organisationsbausteinen OB 100 für Neustart (Warmstart) oder OB 102 für Kaltstart hinterlegen.

Das Anlaufprogramm kann beliebig lang sein, für seine Ausführung besteht keine zeitliche Begrenzung, die Zykluszeitüberwachung ist nicht aktiv. Zeit- oder alarmgesteuerte Bearbeitung ist im Anlaufprogramm nicht möglich. Im Anlauf führen alle Digitalausgänge den Signalzustand 0.

Anlaufart nach manuellem Anlauf

Bei S7-300-CPUs ist nur manueller Neustart (Warmstart) oder Kaltstart (nur CPU 318-2) möglich.

Anlaufart nach automatischem Anlauf

Bei S7-300-CPUs ist nach NETZ-EIN nur Neustart (Warmstart) möglich.

Soll-Ist-Baugruppenüberwachung

Sie können durch Parametrierung festlegen, dass vor dem Anlauf überprüft wird, OB alle Baugruppen, die in der Konfigurationstabelle aufgeführt sind, tatsächlich gesteckt sind und der Baugruppentyp stimmt.

Ist die Baugruppenüberprüfung aktiviert, wird der Anlauf nicht durchgeführt, wenn ein Soll-Ist-Unterschied festgestellt wird.

Überwachungszeiten

Um einen fehlerfreien Anlauf des Automatisierungssystems zu gewährleisten, können Sie die folgenden Überwachungszeiten parametrieren:

- die maximal zulässige Zeit für die Übertragung der Parameter zu den Baugruppen
- die maximal zulässige Zeit für die Fertigmeldung der Baugruppen nach NETZ-EIN

Nach Ablauf der Überwachungszeiten geht die CPU in STOP, bzw. ist nur Neustart (Warmstart) möglich.

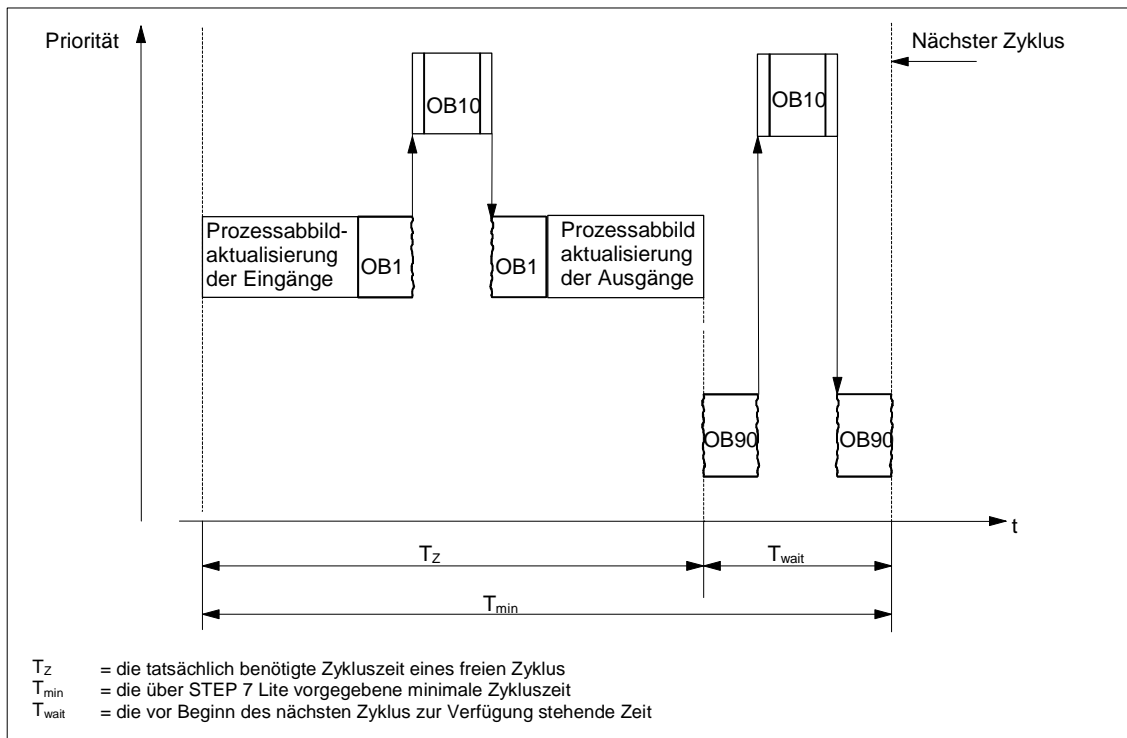
2.2.4.7 Organisationsbaustein für die Programmbearbeitung im Hintergrund (OB 90)

Falls Sie mit STEP 7 Lite eine minimale Zykluszeit vorgegeben haben und diese größer als die tatsächliche Zykluszeit ist, steht der CPU am Ende des zyklischen Programms Bearbeitungszeit zur Verfügung. Diese wird für die Bearbeitung des Hintergrund-OBs genutzt. Falls der OB 90 auf Ihrer CPU nicht vorhanden ist, wartet die CPU, bis die vorgegebene minimale Zykluszeit abgelaufen ist. Sie können über den OB 90 also zeitunkritische Prozesse ablaufen lassen und damit Wartezeiten vermeiden.

Priorität des Hintergrund-OB

Der Hintergrund-OB hat die Prioritätsklasse 29, was der Priorität 0.29 entspricht. Er ist also der OB mit der niedrigsten Priorität. Die Prioritätsklasse kann durch Parametrierung nicht geändert werden.

Das nachfolgende Bild zeigt ein Beispiel für die Bearbeitung des Hintergrundzyklus, des freien Zyklus und des OB 10 (bei bisherigen CPUs).



Programmieren des OB 90

Die Laufzeit des OB 90 wird vom Betriebssystem der CPU nicht überwacht, so dass Sie im OB 90 Schleifen beliebiger Länge programmieren können. Achten Sie auf die Konsistenz der von Ihnen im Hintergrundprogramm verwendeten Daten, indem Sie folgendes bei Ihrer Programmierung berücksichtigen:

- die Rücksetzereignisse des OB 90 (siehe Referenzhandbuch Systemsoftware für S7-300/400, System- und Standardfunktionen),
- die zum OB 90 asynchrone Prozessabbildaktualisierung.

2.2.4.8 Organisationsbausteine für die Fehlerbearbeitung (OB 80 bis OB 87 / OB 121 bis OB 122)

Arten von Fehlern

Die Fehler, die S7-CPU erkennen und auf die Sie mit Hilfe von Organisationsbausteinen reagieren können, lassen sich in zwei Kategorien einteilen:

- **Synchrone Fehler:** Diese Fehler können einem bestimmten Teil des Anwenderprogramms zugeordnet werden. Der Fehler wird während der Bearbeitung von einer bestimmten Operation ausgelöst. Ist der entsprechende synchrone Fehler-OB nicht geladen, geht die CPU in STOP, wenn der Fehler auftritt.
- **Asynchrone Fehler:** Diese Fehler lassen sich nicht direkt dem bearbeiteten Anwenderprogramm zuordnen. Es handelt sich um Prioritätsklassenfehler, Fehler im Automatisierungssystem (z. B. Baugruppendefekte) oder um Redundanzfehler. Ist der entsprechende asynchrone Fehler-OB nicht geladen, geht die CPU in STOP, wenn der Fehler auftritt.

Nachfolgende Tabelle zeigt die Arten von Fehlern, die auftreten können, aufgeteilt nach der Kategorie der Fehler-OBs.

Asynchrone Fehler / Redundanzfehler	Synchrone Fehler
OB 80 Zeitfehler (z. B. Zykluszeit überschritten)	OB 121 Programmierfehler (z. B. DB ist nicht geladen)
OB 82 Diagnosealarm (z. B. Kurzschluss in der Eingangsbaugruppe)	OB 122 Peripheriezugriffsfehler (z. B. Zugriff auf eine Signalbaugruppe, die nicht vorhanden ist)
OB 84 CPU-Hardwarefehler (Fehler an der Schnittstelle zum MPI-Netz)	
OB 85 Programmablauffehler (z. B. OB ist nicht geladen)	
OB 86 Baugruppenträgerausfall	
OB 87 Kommunikationsfehler (z. B. falsche Telegrammkennung bei Globaldaten-Kommunikation)	

Verwenden der OBs für synchrone Fehler

Synchrone Fehler werden während der Bearbeitung einer bestimmten Operation verursacht. Wenn diese Fehler auftreten, erstellt das Betriebssystem einen Eintrag im U-Stack und startet den OB für synchrone Fehler.

Die Fehler-OBs, die von synchronen Fehlern aufgerufen werden, werden als Teil des Programms in derselben Prioritätsklasse wie der Baustein bearbeitet, der beim Erkennen des Fehlers bearbeitet wird. OB 121 und OB 122 können also auf die Werte, die zum Zeitpunkt der Unterbrechung in den Akkumulatoren und anderen Registern gespeichert wurden, zugreifen. Sie können die Werte dazu verwenden, auf die Fehlerbedingung zu reagieren und dann zu der Bearbeitung Ihres Programms zurückzukehren (z. B. bei Zugriffsfehler auf eine Analogeingabebaugruppe im OB 122 mit der SFC 44 RPL_VAL einen Ersatzwert vorgeben). Damit belasten die Lokaldaten der Fehler-OBs aber auch zusätzlich den L-Stack dieser Prioritätsklasse.

Verwenden der OBs für asynchrone Fehler

Wenn das Betriebssystem der CPU einen asynchronen Fehler entdeckt, dann startet es den entsprechenden Fehler-OB (OB 80 bis OB 87). Die OBs für asynchrone Fehler haben die höchste Priorität: Sie können von anderen OBs nicht unterbrochen werden, wenn alle asynchronen Fehler-OBs die gleiche Priorität haben. Treten mehrere asynchrone Fehler-OBs gleicher Priorität gleichzeitig auf, werden sie in der Reihenfolge ihres Auftretens bearbeitet.

Maskieren von Startereignissen

Sie können mit Systemfunktionen (SFC) die Startereignisse für einige Fehler-OBs maskieren bzw. verzögern oder sperren. Nähere Informationen hierzu und zu den Organisationsbausteinen im einzelnen entnehmen Sie dem Referenzhandbuch *Systemsoftware für S7-300/400, System- und Standardfunktionen*.

Art der Fehler-OBs	SFC	Funktion der SFC
Synchrone Fehler-OBs	SFC 36 MSK_FLT	Einzelne Synchronfehlerereignisse maskieren. Maskierte Fehlerereignisse starten keinen Fehler-OB und führen nicht zur programmierten Ersatzreaktion.
	SFC 37 DMSK_FLT	Synchronfehlerereignisse demaskieren
Asynchrone Fehler-OBs	SFC 39 DIS_IRT	Alarm- und Asynchronfehlerereignisse pauschal sperren. Gespernte Fehlerereignisse starten in keinem nachfolgenden CPU-Zyklus Fehler-OBs und führen nicht zur programmierten Ersatzreaktion.
	SFC 40 EN_IRT	Alarm- und Asynchronfehlerereignisse freigeben
	SFC 41 DIS_AIRT	Höherpriorie Alarm- und Asynchronfehlerereignisse bis zum OB-Ende verzögern
	SFC 42 EN_AIRT	Höherpriorie Alarm- und Asynchronfehlerereignisse freigeben

Hinweis

Um Alarme zu ignorieren, ist es effektiver diese per SFC im Anlauf zu sperren, anstatt einen leeren OB (mit Inhalt BE) zu laden.

2.2.5 Bausteinararten für die strukturierte Programmierung

2.2.5.1 Funktionen (FC)

Funktionen gehören zu den Bausteinen, die Sie selbst programmieren. Eine Funktion ist ein Codebaustein "ohne Gedächtnis". Temporäre Variablen der FC werden im Lokaldaten-Stack gespeichert. Diese Daten gehen nach der Bearbeitung der FC verloren. Funktionen können zum Speichern von Daten globale Datenbausteine nutzen.

Weil eine FC keinen zugeordneten Speicher hat, müssen Sie immer Aktualparameter für eine FC angeben. Sie können den Lokaldaten einer FC keine Anfangswerte zuordnen.

Anwendungsbereich

Eine FC enthält ein Programm, das immer dann ausgeführt wird, wenn die FC von einem anderen Codebaustein aufgerufen wird. Funktionen können eingesetzt werden, um

- einen Funktionswert an den aufrufenden Baustein zurückzugeben (Beispiel: mathematische Funktionen).
- eine technologische Funktion auszuführen (Beispiel: Einzelsteuerung mit Binärverknüpfung).

Zuordnen von Aktualparametern zu Formalparametern

Ein Formalparameter ist ein Platzhalter für den "tatsächlichen" Parameter, den Aktualparameter. Aktualparameter ersetzen beim Aufruf einer FC die Formalparameter. Sie müssen den Formalparametern einer FC immer Aktualparameter zuordnen (z. B. dem Formalparameter "Start" einen Aktualparameter "E3.6"). Die Eingangs-, Ausgangs- und Durchgangparameter, die von der FC verwendet werden, werden als Pointer auf die Aktualparameter des Codebausteins gespeichert, der die FC aufgerufen hat.

2.2.5.2 Funktionsbausteine (FB)

Funktionsbausteine gehören zu den Bausteinen, die Sie selbst programmieren. Ein Funktionsbaustein ist ein Baustein "mit Gedächtnis". Er verfügt über einen zugeordneten Datenbaustein als Speicher (Instanz-Datenbaustein). Die Parameter, die an den FB übergeben werden, sowie die statischen Variablen werden im Instanz-DB gespeichert. Die temporären Variablen werden im Lokaldaten-Stack gespeichert.

Daten, die im Instanz-DB gespeichert werden, gehen nicht verloren, wenn die Bearbeitung des FB beendet ist. Daten, die im Lokaldaten-Stack gespeichert werden, gehen nach der Bearbeitung des FB verloren.

Hinweis

Um Fehler beim Arbeiten mit FBs zu vermeiden, lesen Sie Zulässige Datentypen beim Übergeben von Parametern im Anhang.

Anwendungsbereich

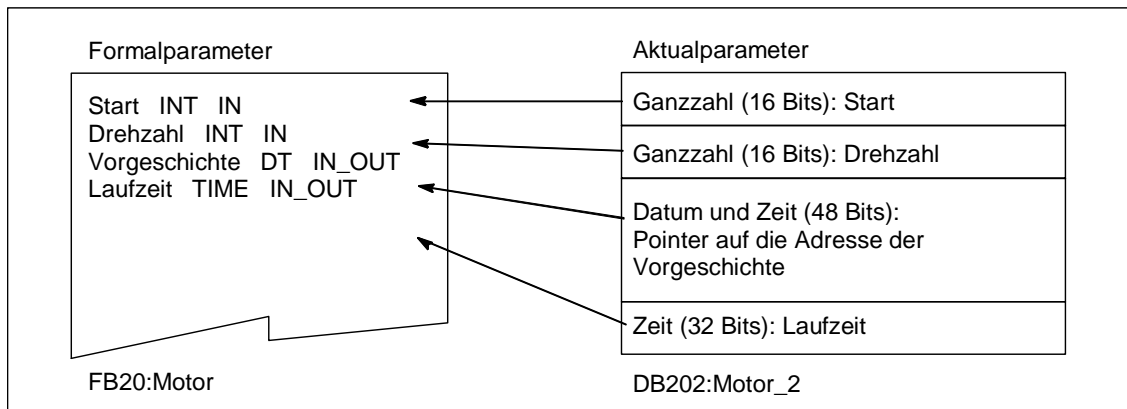
Ein FB enthält ein Programm, das immer dann ausgeführt wird, wenn der FB von einem anderen Codebaustein aufgerufen wird. Funktionsbausteine erleichtern das Programmieren häufig wiederkehrender, komplexer Funktionen.

FBs und Instanz-DBs

Jedem Aufruf eines Funktionsbausteins, der Parameter übergibt, ist ein Instanz-Datenbaustein zugeordnet.

Durch das Aufrufen mehrerer Instanzen eines FB können Sie mit einem FB mehrere Geräte steuern. Ein FB für einen Motortyp beispielsweise kann verschiedene Motoren steuern, indem er verschiedene Instanzdaten für die verschiedenen Motoren verwendet. Die Daten für jeden einzelnen Motor (wie Drehzahl, Ramping, akkumulierte Betriebszeit usw.) können in einem oder mehreren Instanz-DBs gespeichert werden.

Nachfolgendes Bild zeigt die Formalparameter eines FB, der die Aktualparameter verwendet, die im Instanz-DB gespeichert sind.



Variablen vom Datentyp FB

Ist Ihr Anwenderprogramm so strukturiert, dass in einem FB weitere, bereits existierende Funktionsbausteine aufgerufen werden, können Sie in die Variablendeklarationstabelle des aufrufenden FB die aufzurufenden FBs als statische Variablen mit dem Datentyp FB aufnehmen. Damit erreichen Sie eine Verschachtelung von Variablen und die Konzentrierung der Instanzdaten in einem Instanz-Datenbaustein (Multiinstanz).

Zuordnen von Aktualparametern zu Formalparametern

Im allgemeinen ist es in STEP 7 Lite nicht erforderlich, dass Sie den Formalparametern eines FB Aktualparameter zuordnen.

Es gibt jedoch Ausnahmen. Aktualparameter müssen zugeordnet werden:

- bei einem Durchgangsparameter eines zusammengesetzten Datentyps (z. B. STRING, ARRAY oder DATE_AND_TIME)
- bei allen Parametertypen (z. B. TIMER, COUNTER oder POINTER)

STEP 7 Lite ordnet den Formalparametern eines FB die Aktualparameter folgendermaßen zu:

- *Wenn Sie in der Aufrufanweisung Aktualparameter angeben:* Die Operationen des FB verwenden die bereitgestellten Aktualparameter.
- *Wenn Sie in der Aufrufanweisung keine Aktualparameter angeben:* Die Operationen des FB verwenden die Werte, die in dem Instanz-DB gespeichert sind.

Nachfolgende Tabelle zeigt, welchen Variablen des FB Aktualparameter zugeordnet werden müssen.

Variablen	Datentyp		Parametertyp
	Elementarer Datentyp	Zusammengesetzter Datentyp	
Eingang	Parameter nicht erforderlich	Parameter nicht erforderlich	Aktualparameter erforderlich
Ausgang	Parameter nicht erforderlich	Parameter nicht erforderlich	Aktualparameter erforderlich
Durchgang	Parameter nicht erforderlich	Aktualparameter erforderlich	–

Zuordnen von Anfangswerten zu Formalparametern

Sie können den Formalparametern im Deklarationsteil des FB Anfangswerte zuordnen. Diese Werte werden in den Instanz-DB übernommen, der einem FB zugeordnet wird.

Ordnen Sie den Formalparametern in der Aufrufanweisung keine Aktualparameter zu, dann verwendet STEP 7 Lite die Werte, die im Instanz-DB gespeichert sind. Diese Daten können Anfangswerte sein, die in der Variablendeklarationstabelle eines FB eingegeben wurden.

Nachfolgende Tabelle zeigt, welche Variablen einem Anfangswert zugeordnet werden können. Weil die temporären Daten nach der Bearbeitung des Bausteins nicht gespeichert werden, können Sie ihnen keine Werte zuordnen.

	Datentyp		
Variablen	Elementarer Datentyp	Zusammengesetzter Datentyp	Parametertyp
Eingang	Anfangswert zulässig	Anfangswert zulässig	–
Ausgang	Anfangswert zulässig	Anfangswert zulässig	–
Durchgang	Anfangswert zulässig	–	–
Statisch	Anfangswert zulässig	Anfangswert zulässig	–
Temporär	–	–	–

2.2.5.3 Instanz-Datenbausteine

Jedem Aufruf eines Funktionsbausteins, der Parameter übergibt, ist ein Instanz-Datenbaustein zugeordnet. Im Instanz-DB sind die Aktualparameter und die statischen Daten des FB abgelegt. Die im FB deklarierten Variablen bestimmen die Struktur des Instanz-Datenbausteins. Mit Instanz wird der Aufruf eines Funktionsbausteins bezeichnet. Wird z. B. ein Funktionsbaustein im S7-Anwenderprogramm fünfmal aufgerufen, so existieren fünf Instanzen dieses Bausteins.

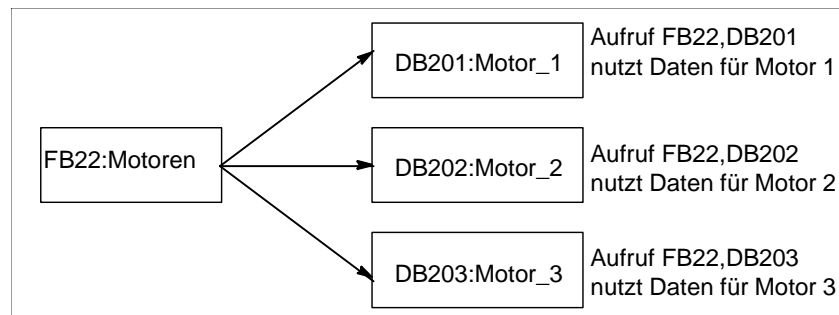
Erzeugen eines Instanz-DB

Bevor Sie einen Instanz-Datenbaustein erzeugen, muss der zuzuordnende FB bereits existieren. Die Nummer dieses FB geben Sie beim Anlegen des Instanz-Datenbausteins an.

Ein Instanz-DB für jede Instanz

Wenn Sie mehrere Instanz-Datenbausteine einem Funktionsbaustein (FB) zuordnen, der einen Motor steuert, so können Sie diesen FB zum Steuern verschiedener Motoren verwenden.

Die unterschiedlichen Daten für die einzelnen Motoren (wie z. B. Drehzahl, Hochlaufzeit, Gesamtbetriebszeit) werden in den verschiedenen Datenbausteinen gespeichert. Je nachdem welcher DB dem FB beim Aufruf zugeordnet wird, kann ein anderer Motor gesteuert werden. Auf diese Weise wird nur ein Funktionsbaustein für mehrere Motoren benötigt (siehe nachfolgendes Bild).

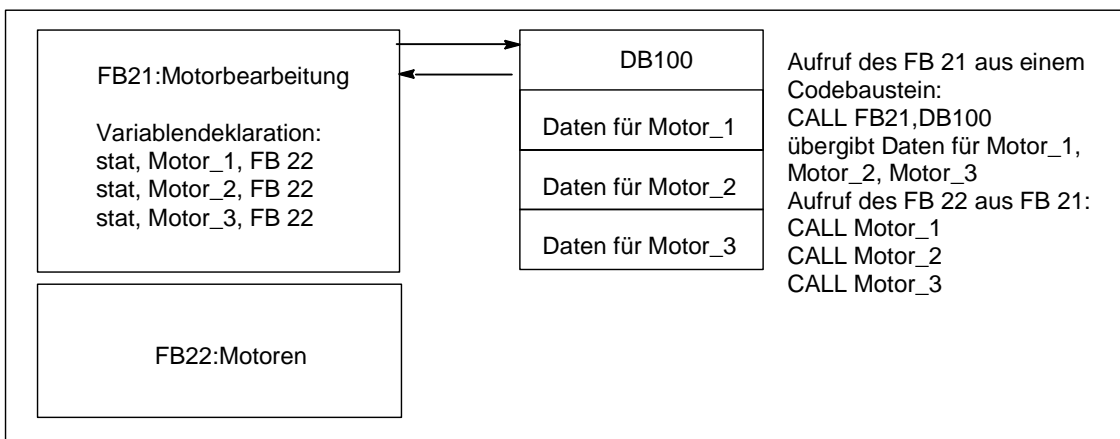


Ein Instanz-DB für mehrere Instanzen eines FB (Multiinstanzen)

Sie können einem FB die Instanzdaten für verschiedene Motoren gemeinsam in einem Instanz-DB übergeben. Dazu müssen Sie den Aufruf der Motorensteuerungen in einem weiteren FB vornehmen und im Deklarationsteil des rufenden FB statische Variablen mit dem Datentyp eines FB für die einzelnen Instanzen deklarieren.

Wenn Sie einen Instanz-DB für mehrere Instanzen eines FB verwenden, sparen Sie Speicherplatz und optimieren den Einsatz von Datenbausteinen.

Im nachfolgenden Bild ist beispielsweise der rufende FB der FB 21 "Motorbearbeitung", die Variablen sind vom Datentyp FB 22 und die Instanzen werden mit Motor_1, Motor_2 und Motor_3 bezeichnet.



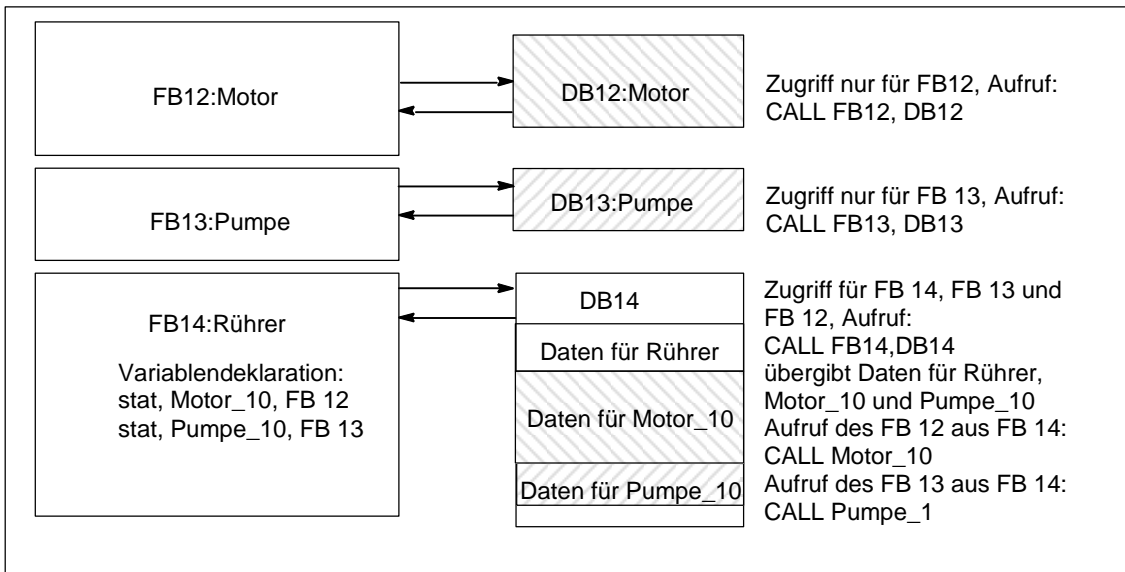
Der FB 22 braucht in diesem Beispiel keinen eigenen Instanz-Datenbaustein, weil seine Instanzdaten im Instanz-Datenbaustein des rufenden FB gespeichert sind.

Ein Instanz-DB für mehrere Instanzen unterschiedlicher FBs (Multiinstanzen)

Sie können in einem Funktionsbaustein Instanzen anderer, bereits erstellter FBs aufrufen. Die dafür benötigten Instanzdaten können sie dem Instanz-Datenbaustein des aufrufenden FBs zuordnen, d. h. Sie brauchen für die aufgerufenen FBs in diesem Fall keine zusätzlichen Datenbausteine.

Für diese Multiinstanzen in einem Instanz-DB müssen Sie im Deklarationsteil des rufenden FBs statische Variablen mit dem Datentyp des aufgerufenen FB für die einzelnen Instanzen deklarieren. Der Aufruf innerhalb des FB erfolgt dann ohne Angabe eines Instanz-DB nur über den Namen der Variable.

In dem Beispiel im Bild werden die zugeordneten Instanzdaten gemeinsam in einem Instanz-DB gespeichert.



2.2.6 Globale Datenbausteine (DB)

Datenbausteine enthalten im Gegensatz zu Codebausteinen keine STEP 7 Lite-Anweisungen. Sie dienen der Aufnahme von Anwenderdaten, d. h. in den Datenbausteinen stehen variable Daten, mit denen das Anwenderprogramm arbeitet. Globale Datenbausteine dienen der Aufnahme von Anwenderdaten, die von allen anderen Bausteinen aus verwendet werden können.

Die Größe von DBs kann variieren. Die maximal zulässige Größe entnehmen Sie dem Handbuch *Automatisierungssystem S7-300, Aufbauen, CPU-Daten*.

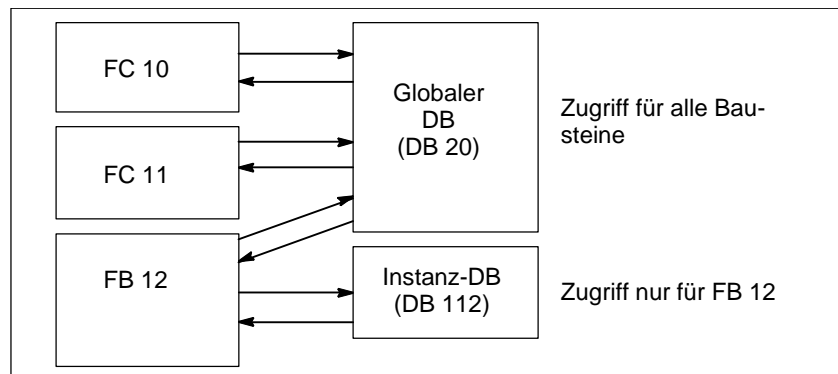
Die Struktur globaler Datenbausteine können Sie beliebig festlegen.

Globale Datenbausteine im Anwenderprogramm

Wird ein Codebaustein (FC, FB oder OB) aufgerufen, so kann er temporär Speicherplatz im Lokaldatenbereich (L-Stack) belegen. Zusätzlich zu diesem Lokaldatenbereich kann ein Codebaustein einen Speicherbereich in Form eines DB öffnen. Im Gegensatz zu den Daten im Lokaldatenbereich werden die in einem DB enthaltenen Daten nicht gelöscht, wenn der DB geschlossen wird, bzw. die Bearbeitung des zugehörigen Codebausteins beendet ist.

Jeder FB, FC oder OB kann die Daten aus einem globalen DB lesen oder selbst Daten in einen globalen DB schreiben. Diese Daten bleiben im DB auch dann erhalten, wenn der DB verlassen wird.

Ein globaler DB und ein Instanz-DB können gleichzeitig geöffnet sein.
Nachfolgendes Bild zeigt die verschiedenen Zugriffe auf Datenbausteine.



2.2.6.1 Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC)

Vorgefertigte Bausteine

Sie müssen nicht jede Funktion selbst programmieren. S7-CPU's bieten Ihnen vorgefertigte Bausteine, die Sie aus dem Anwenderprogramm heraus aufrufen können.

Nähere Informationen finden Sie in der Referenzhilfe zu Systemfunktionsbausteinen und Systemfunktionen (Sprünge in Sprachbeschreibungen und Baustein-Hilfen,).

Systemfunktionsbausteine

Ein Systemfunktionsbaustein SFB ist ein Funktionsbaustein, der in die S7-CPU integriert ist. Weil SFBs Teil des Betriebssystems sind, werden sie nicht als Teil des Programms geladen. Wie FBs sind SFBs Bausteine "mit Gedächtnis". Sie müssen auch für SFBs Instanz-Datenbausteine erstellen und als Teil des Programms in die CPU laden.

S7-CPU's bieten SFBs

- zur Kommunikation über projektierte Verbindungen (nicht in STEP 7 Lite projektierbar)
- für integrierte Sonderfunktionen (z. B. SFB 29 "HS_COUNT" auf der CPU 312 IFM und der CPU 314 IFM)

Systemfunktionen

Eine Systemfunktion ist eine vorprogrammierte Funktion, die in die S7-CPU integriert ist. Sie können die SFC aus Ihrem Programm aufrufen. Weil SFCs Teile des Betriebssystems sind, werden sie nicht als Teil des Programms geladen. Wie FCs sind SFCs Bausteine "ohne Gedächtnis".

CPUs bieten SFCs für:

- Kopier- und Bausteinfunktionen
- die Programmkontrolle
- die Hantierung der Uhr und des Betriebsstundenzählers
- die Übertragung von Datensätzen
- die Hantierung von Uhrzeit- und Verzögerungsalarmen
- die Hantierung von Synchronfehlerereignissen, Alarm- und Asynchronfehlerereignissen
- die Auskunft über statische und dynamische Systemdaten, z. B. Diagnose
- die Prozessabbildaktualisierung und die Bitfeldbearbeitung
- die Adressierung von Baugruppen
- die dezentrale Peripherie (nicht mit STEP 7 Lite projektierbar)
- die Globaldaten-Kommunikation (nicht mit STEP 7 Lite projektierbar)
- die Kommunikation über nichtprojektierte Verbindungen
- Erzeugung bausteinbezogener Meldungen (nicht mit STEP 7 Lite projektierbar)

Weitere Informationen

Nähere Informationen zu SFBs und SFCs entnehmen Sie dem Referenzhandbuch *Systemsoftware für S7-300/400, System- und Standardfunktionen*. Dem Handbuch *Automatisierungssystem S7-300, Aufbauen, CPU-Daten* entnehmen Sie, welche SFBs und SFCs verfügbar sind.

3 Starten und Bedienen

3.1 Starten von STEP 7 Lite

Nach dem Start von Windows finden Sie auf der Windows-Oberfläche ein Symbol für STEP 7 Lite.

STEP 7 Lite wird am schnellsten mit einem Doppelklick auf dieses Symbol gestartet.

Alternativ können Sie STEP 7 Lite auch über die Schaltfläche "Start" auf der Task-Leiste starten. Den Eintrag finden Sie unter "Simatic".

Hinweis

Nähere Informationen über Windows-Standardbedienungen und Standardoptionen finden Sie in Ihrem Windows-Benutzerhandbuch oder in der Online-Hilfe Ihres Windows Betriebssystems.

Vorgehen

Sie erstellen Automatisierungslösungen in Form von "Projekten". Sie erleichtern sich das Arbeiten, wenn Sie sich zuvor mit einigen grundlegenden Themen vertraut machen:

- der Benutzungsoberfläche,
- einigen Basisbedienungen,
- der Online-Hilfe.

3.2 Aufrufen der Hilfefunktionen

Online-Hilfe

Die Online-Hilfe bietet Ihnen Informationen für die Stelle an, an der Sie sie benötigen. So können Sie schnell und zielsicher Information nachschlagen, ohne in Handbüchern suchen zu müssen. In der Online-Hilfe finden Sie:

- **STEP 7 Lite Hilfe:** informiert über die grundsätzliche Vorgehensweise und das notwendige Grundwissen für die Konfigurierung und Programmierung eines Automatisierungssystems.
- **Direkthilfe** (Tasten Shift+F1): informiert über ein aktives Element - z. B. in einem Dialogfeld.
- **Info:** liefert Informationen zur aktuellen Version der Applikation.

Aufrufen der Online-Hilfe

Sie können die Online-Hilfe auf verschiedene Arten aufrufen:

- Wählen Sie einen Menübefehl aus dem Menü "Hilfe" in der Menüleiste.
- Klicken Sie in der Symbolleiste das Symbol für den Hilfezeiger an und selektieren Sie das Element, zu dem Sie Hilfe benötigen.
- Drücken Sie die Tasten Shift+F1 und selektieren Sie mit dem Hilfezeiger das Element, zu dem Sie Hilfe benötigen.
- Drücken Sie die Taste F1, um die STEP 7 Lite Hilfe aufzurufen.

Aufrufen der Kurzinfo

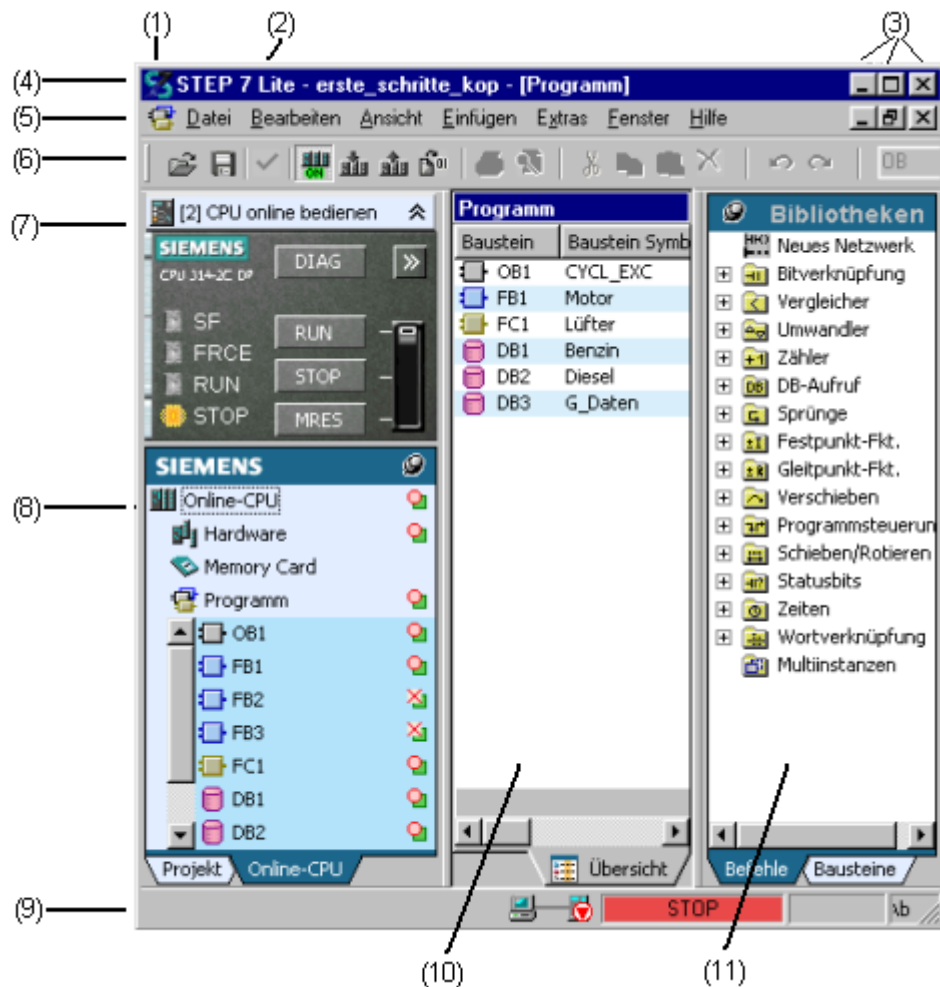
Eine Kurzinfo wird z. B. zu Schaltflächen in der Symbolleiste oder zu den Elementen in den Befehls-/Bausteinbibliotheken angezeigt, wenn Sie den Mauszeiger auf eine Schaltfläche oder ein Element positionieren und ihn dort kurze Zeit stehen lassen.

Durch ein Symbol in der rechten unteren Ecke der Kurzinfo wird angezeigt, wo direkt aus der Kurzinfo die Direkthilfe aufgerufen werden kann. Sie können die Direkthilfe entweder nach einer weiteren kurzen Wartezeit automatisch aufrufen oder aktiv durch Klicken auf die Kurzinfo.

3.3 Benutzungsoberfläche und Bedienung

3.3.1 Aufbau der Benutzungsoberfläche

Die Bereiche der Benutzungsoberfläche von STEP 7 Lite sind im nachfolgenden Bild dargestellt:



(1)	Systemmenü (Maximieren, Schließen etc.)	(7)	CPU-Bedienpanel
(2)	Titel des aktiven Fensters	(8)	Projektfenster
(3)	Schaltflächen für Minimieren, Maximieren, Schließen	(9)	Statusleiste
(4)	Titelleiste	(10)	Arbeitsbereich: enthält Informationen, die Sie sich anzeigen lassen oder bearbeiten
(5)	Menüleiste	(11)	Bibliotheken
(6)	Symbolleiste		

Titelleiste und Menüleiste

Titelleiste und die Menüleiste befinden sich immer am oberen Rand des Fensters. Die Titelleiste enthält den Fenstertitel und Symbole für die Fenstersteuerung. Die Menüleiste enthält alle Menüs, die im Fenster zur Verfügung stehen.

Symbolleiste

Die Symbolleiste enthält Symbole, über die Sie schnell durch Mausklick auf häufig verwendete und aktuell verfügbare Menübefehle zugreifen können. Eine Kurzinformation über die Funktion eines Symbols wird aufgeblendet, wenn Sie den Mauszeiger kurze Zeit auf dem Symbol positioniert lassen.

Wenn in der aktuellen Arbeitsumgebung kein Zugriff auf ein Symbol möglich ist, so wird dieses grau dargestellt.

CPU-Bedienpanel

Das CPU-Bedienpanel stellt ein Abbild einer SIMATIC S7-300 Zentralbaugruppe dar. Besteht eine Online-Verbindung zu einer CPU, so wird der Betriebszustand der CPU anhand der Leuchtanzeigen und der Schlüsselschalterstellung angezeigt. Über einzelne Schaltflächen, wie "RUN" und "STOP" kann die CPU bedient werden.

Bibliotheken

In diesem Bereich finden Sie unter dem Register "Befehle" KOP- und FUP-Operationen und unter dem Register "Bausteine" alle dem System bekannten Bibliotheken.

Projektfenster

Dieser Bereich ermöglicht die Auswahl von Objekten des Projekts, um sie zu bearbeiten.

Beispiel: Wenn Sie auf die Zeile "Symboltabelle" im Projektfenster klicken, dann wird im Arbeitsbereich die Symboltabelle zur Bearbeitung angezeigt.

Arbeitsbereich


















In diesem Bereich wird, abhängig vom ausgewähltem Objekt im Projektfenster, die entsprechende Sicht für die Bearbeitung angezeigt.

Beispiel: Wenn Sie im Projektfenster einen Baustein ausgewählt haben, können Sie im "Bausteineditor" diesen Baustein editieren.

Statusleiste

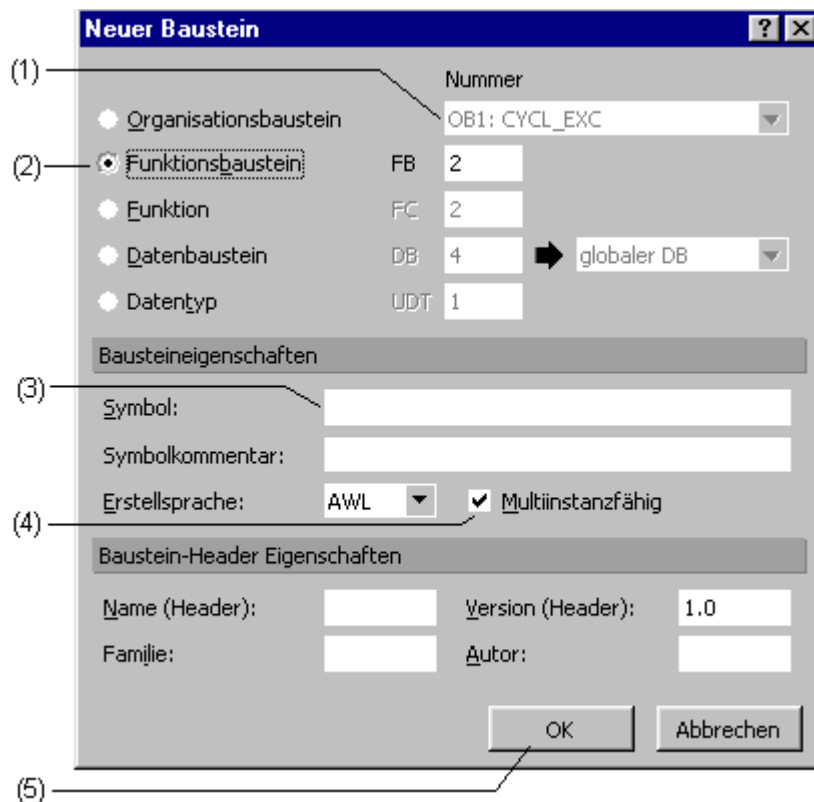
In der Statusleiste werden kontextabhängige Informationen angezeigt.

3.3.2 Symbole im Projektfenster

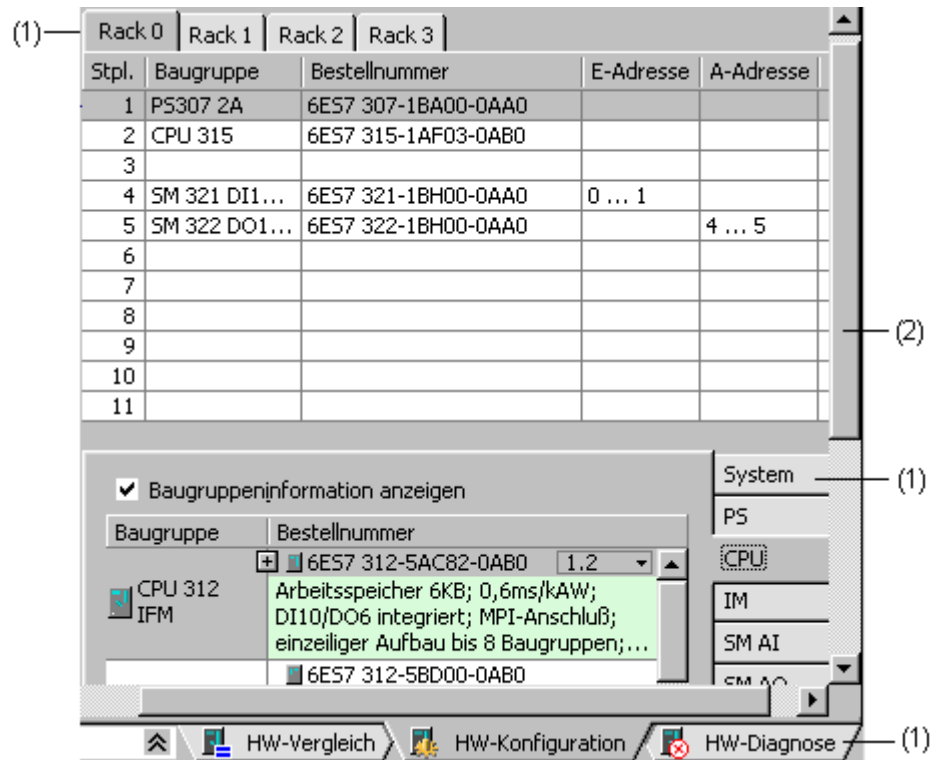
Symbol	Bedeutung	Symbol	Bedeutung
	Offlineprojekt		Online-CPU
	Hardware-Konfiguration		Symboltabelle
	Beobachten/Steuern		Querverweise ("Querverweisliste", "Verwendete Operanden", "Programmstruktur")
	Projektdokumentation		Programm (Summe aller Code- und Datenbausteine)
	Online-/Offlinebausteine Symbolisiert einen Stellvertretereintrag für Onlinebausteine, die nicht im Offlineprojekt vorhanden sind. Ein Doppelklick auf das Symbol öffnet ein Dialogfeld, in dem die Bausteine aus der CPU in das Offlineprojekt geladen werden können.		Codebaustein. Je nach Bausteintyp ist das Symbol unterschiedlich gefärbt.
	Datenbaustein		Memory Card (Micro Memory Card MMC)
	Geändert nicht gespeichert Der gelbe Stern symbolisiert, dass das Objekt geändert aber der Inhalt noch nicht übernommen oder gespeichert wurde.		Objekt ist im Projekt und der Online-CPU gleich.
	Bausteinschutz Das Vorhängeschloss symbolisiert, dass der Baustein einen Bausteinschutz besitzt; ohne besondere Berechtigung kann er nicht gelesen und verändert werden.		Das Objekt ist in der aktuellen Sicht des Projektfensters vorhanden, fehlt aber in der nicht angewählten Sicht (Sicht "Projekt" / Sicht "Online-CPU"). Weitere Hilfen erhalten Sie über den Hilfezeiger in STEP 7 Lite.
	Das Objekt im Projekt stimmt nicht mit dem Objekt in der Online-CPU überein. Weitere Hilfen erhalten Sie über den Hilfezeiger in STEP 7 Lite.		

3.3.3 Elemente in Fenstern und Dialogfeldern

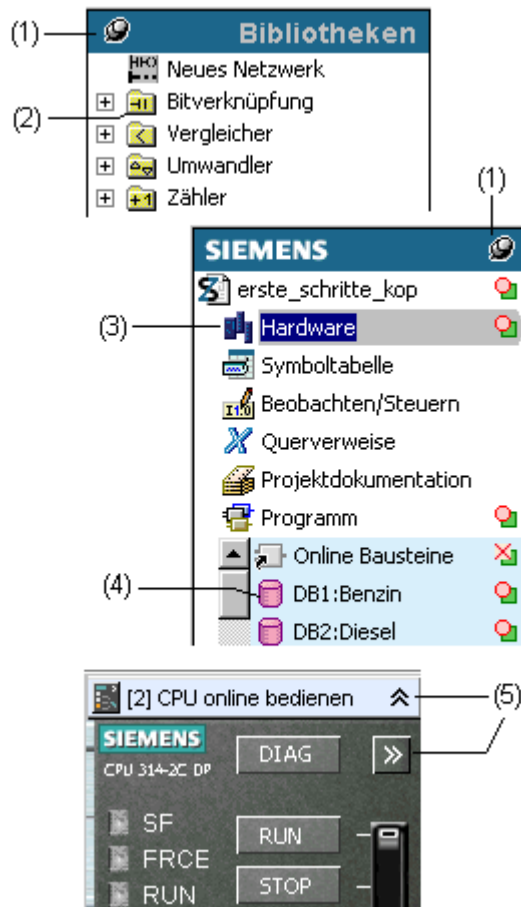
Zur Bedienung, Auswahl und Eingabe in Fenstern und Dialogfeldern stehen spezifische Elemente zur Verfügung, die im Folgenden anhand von Beispielen erklärt werden.





(1)	Klapplisten: Ein nach unten gerichteter Pfeil zeigt an, dass für dieses Feld eine Auswahlliste zur Verfügung steht.
(2)	Runde Optionsfelder zur Auswahl einer von mehreren Möglichkeiten.
(3)	Textfelder zur Eingabe von Text oder Zahlen über die Tastatur.
(4)	Eckige Optionskästchen zur Auswahl einer oder mehrerer Möglichkeiten.
(5)	Schaltflächen



(1)	Register: Der Inhalt mancher Fenster und Dialogfelder wird der bessern Übersicht wegen in Register aufgeteilt. Um ein bestimmtes Register "nach vorne" zu holen, klicken Sie einfach auf die Registerbezeichnung.
(2)	Bildlaufleiste zur Anzeige eines zur Zeit unsichtbaren Bereichs eines Fensters oder Dialogfeldes. Ziehen Sie am Verschiebepalken oder klicken Sie auf die Pfeile, um den sichtbaren Ausschnitt zu verschieben.



(1)	<p>Pinnadeln: Die Pinnadel dient zum Arretieren des Projektfensters und der Bibliotheken. Ein nicht arretiertes Fenster blendet sich automatisch aus, um einen möglichst großen Arbeitsbereich zur Verfügung zu stellen. Befindet sich der Mauszeiger am rechten bzw. linken Rand des STEP 7 Lite Fensters, so wird das Projektfenster und die Bibliothek eingeblendet. So lange sich der Mauszeiger innerhalb der ausblendbaren Bereiche befindet, bleiben sie eingeblendet.</p> <p>Durch Einfachklick auf das Symbol kann der Zustand Arretiert und nicht Arretiert gewechselt werden.</p> <p> Das Projektfenster bzw. die Bibliotheken sind arretiert.</p> <p> Das Projektfenster bzw. die Bibliotheken sind nicht arretiert.</p>
(2)	<p>Ordner: Die Befehlsübersicht in den Bibliotheken ist in einzelne Befehlsordner untergliedert. Mit einem Doppelklick können Sie den Ordner öffnen und den gewünschten Befehl per Drag&Drop in den Bausteineditor einfügen.</p>
(3)	<p>Symbole zum Aufruf zentraler Objekte und Funktionen (z. B. Hardware, Symboltabelle, Steuern & Beobachten etc.)</p>
(4)	<p>Bausteine des Projekts: Durch Doppelklick auf die Datenbausteine bzw. Codebausteine wird der dazugehörige Editor gestartet</p>
(5)	<p>Schaltflächen zum Erweitern und Reduzieren von Fenstern.</p> <p>Klicken Sie auf die Schaltfläche, um das Fenster zu erweitern bzw. zu reduzieren. Je nach Ansicht wechselt das Symbol der Schaltfläche.</p>

3.3.4 Sitzungsgedächtnis

STEP 7 Lite kann sich den Fensterinhalt, also die geöffneten Projekte, sowie die Fensteranordnung merken. Ebenso wird die eingestellte Mnemonik (deutsch oder englisch) beibehalten.

- Über Menübefehl **Fenster > Anordnung speichern** werden der aktuelle Fensterinhalt und die aktuelle Fensteranordnung gespeichert.
- Über Menübefehl **Fenster > Anordnung wiederherstellen** werden der mit Menübefehl **Fenster > Anordnung speichern** gespeicherte Fensterinhalt und die Fensteranordnung wiederhergestellt.

Achtung

Online-Fensterinhalte, z. B. die Bausteine in der angeschlossenen CPU, werden nicht gespeichert.

Eventuell eingegebene Passwörter für den Zugriff auf Zielsysteme (z. B. S7-300) werden über das Sitzungsende hinaus nicht gespeichert.

3.3.5 Ändern der Fensteranordnung

Um alle angezeigten Fenster hintereinander gestaffelt anzuordnen, wählen Sie den Menübefehl **Fenster > Anordnen > Überlappend**.

Um alle angezeigten Fenster gleichmäßig untereinander anzuordnen, wählen Sie den Menübefehl **Fenster > Anordnen > Untereinander**.

Um alle angezeigten Fenster gleichmäßig nebeneinander anzuordnen, wählen Sie den Menübefehl **Fenster > Anordnen > Nebeneinander**.

3.3.6 Speichern und Wiederherstellen der Fensteranordnung

STEP 7 Lite bietet die Möglichkeit, die aktuelle Anordnung von Fenstern zu speichern und zu einem anderen Zeitpunkt wiederherzustellen. Die Einstellung können Sie über den Menübefehl **Extras > Einstellungen** vornehmen.

Was wird gespeichert?

Wenn Sie die Fensteranordnung speichern, werden folgende Informationen aufgezeichnet:

- Geöffnete Fenster und die zugehörigen Fensterpositionen
- Reihenfolge von eventuell übereinanderliegenden Fenstern

Fensteranordnung speichern

Um die aktuelle Anordnung der Fenster zu speichern, wählen Sie den Menübefehl **Fenster > Anordnung speichern**.

Fensteranordnung wiederherstellen

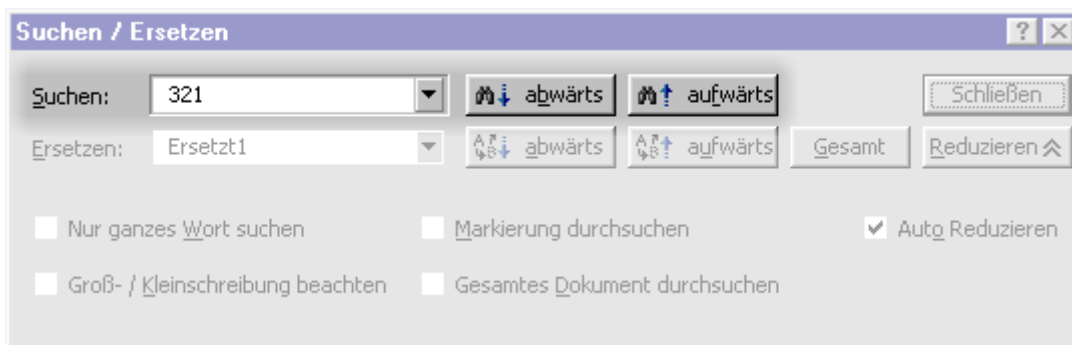
Um eine gespeicherte Fensteranordnung wieder herzustellen, wählen Sie den Menübefehl **Fenster > Anordnung wiederherstellen**.

3.3.7 Suchen bzw. Ersetzen von Begriffen

Bei der Bearbeitung von Projekten und bei der Recherche innerhalb eines Projekts, bieten die Suchen- bzw. die Suchen-/Ersetzenfunktionen von STEP 7 Lite eine wertvolle Hilfe. Grundsätzlich stehen die Funktionen in HW-Konfigurieren, in der Symboltabelle, in den Querverweisen und im Bausteineditor zur Verfügung. Die angebotenen Optionen des Dialogs hängen von selektierter Sicht ab. So stehen z.B. die Ersetzenfunktionen nur in Sichten zur Verfügung, in denen editiert werden kann.

Suchen eines Begriffs

Verwenden Sie zum Suchen eines Begriffs die Klappliste und Schaltflächen der oberen Zeile des Dialogs.

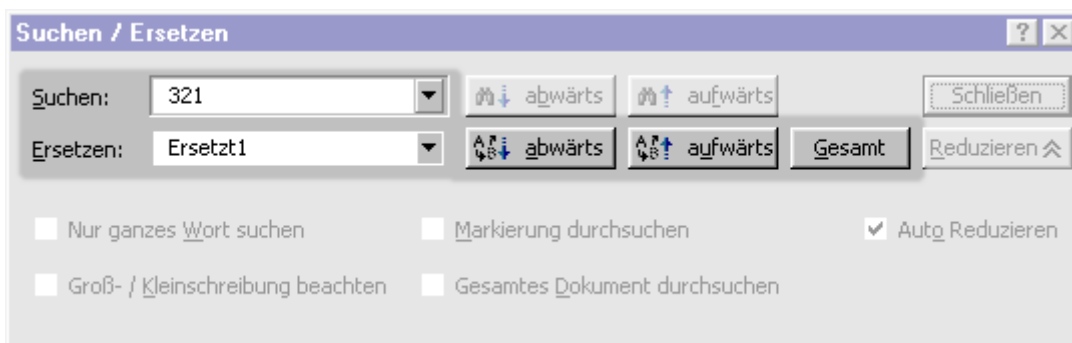


Gehen Sie wie folgt vor:

1. Tragen Sie den zu suchenden Begriff in das Eingabefeld der Klappliste ein oder wählen Sie einen vormaligen Begriff aus der Klappliste aus.
2. Starten Sie die Suche mit der Suchschaltfläche "abwärts" oder "aufwärts". Die Suche beginnt an der Position der Schreibmarke in die angegebene Richtung. Betätigen Sie die Schaltfläche erneut, um die Suche fortzusetzen.

Suchen und Ersetzen eines Begriffs

Verwenden Sie zum Suchen und Ersetzen eines Begriffs die Klappliste der Suchfunktion und die Klappliste und Schaltflächen der zweiten Zeile des Dialogs.

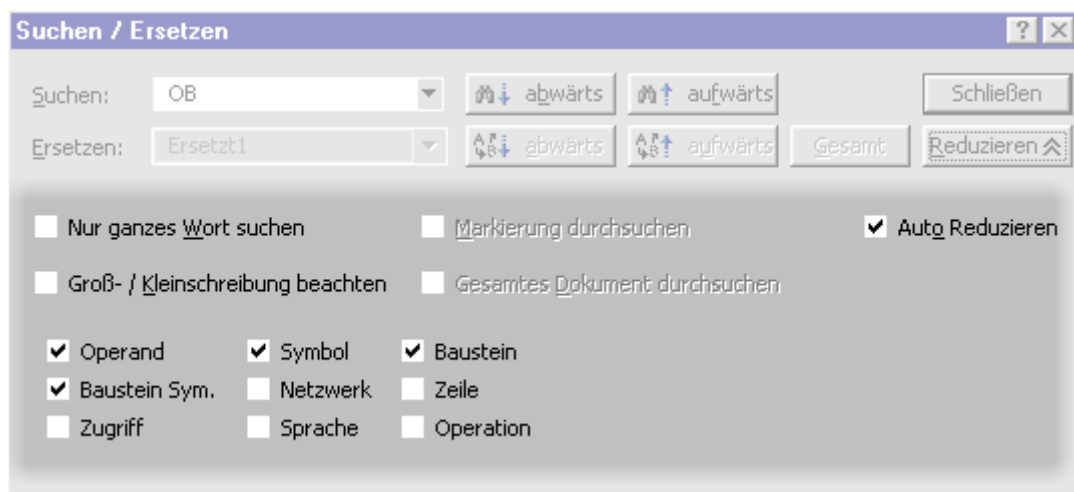


Gehen Sie wie folgt vor:

1. Tragen Sie den zu suchenden Begriff in das Eingabefeld der Such-Klappliste ein oder wählen Sie einen vormaligen Begriff aus der Klappliste aus.
2. Tragen Sie den zu ersetzenden Begriff in das Eingabefeld der Ersetz-Klappliste ein oder wählen Sie einen früheren Begriff aus der Klappliste aus.
3. Starten Sie die Suchen und Ersetzenfunktion. Folgende Möglichkeiten stehen zur Verfügung:
 - Suchen und Ersetzen "abwärts"
 - Suchen und Ersetzen "aufwärts"
 - Suchen und Ersetzen "Gesamt"

Anzahl der Suchergebnisse durch Angabe von Suchkriterien erweitern oder einengen

Sie können die Anzahl der Suchergebnisse erweitern oder einengen, indem Sie zusätzliche Suchkriterien anwählen. Die Suchkriterien finden Sie im unteren Teil des Dialogs, wenn der Dialog über die Schaltfläche "Erweitern/Reduzieren" erweitert wurde. Die Anzahl und Art der angezeigte Suchkriterien hängt von der aktiven Sicht ab. Weitere Informationen zu den Suchkriterien finden Sie in der Direkthilfe von STEP 7 Lite.



3.3.8 Schritte zum Manipulieren von Objekten

3.3.8.1 Umbenennen von Objekten

Um Objekte umzubenennen, gehen Sie folgendermaßen vor:

1. Markieren Sie das gewünschte Objekt.
2. Klicken Sie auf den Namen des markierten Objekts, um den Namen editierbar zu machen. Das Namensfeld wird umrandet dargestellt, und der Mauszeiger wird zu einer Einfügemarke.
3. Editieren Sie den Objektnamen. Im Allgemeinen gelten dafür die Namenskonventionen der verwendeten Windows-Version.
4. Um das Umbenennen zu beenden, haben Sie folgende Möglichkeiten:
 - Betätigen Sie die Eingabetaste, um den geänderten Namen zu übernehmen. Wenn der neue Name nicht zulässig ist, wird der vorherige wiederhergestellt.
 - Betätigen Sie ESC, um den Editiervorgang abzubrechen und den vorherigen Objektnamen wiederherzustellen.

3.3.8.2 Verschieben von Objekten

Die Bausteinobjekte können in Reihenfolge ihrer Auflistung im Projektfenster verschoben werden. Gehen Sie wie folgt vor:

1. Markieren Sie den Baustein, den Sie verschieben möchten, und halten Sie die linke Maustaste gedrückt.
2. Bewegen Sie die Maus an die Stelle, an die der Baustein verschoben werden soll. Ein schwarzer Balken zeigt an welcher Stelle der Baustein eingefügt werden kann.
3. Lassen Sie den Baustein durch freigegeben der linken Maustaste fallen.

3.3.8.3 Löschen von Objekten

Um ein Objekt zu löschen, gehen Sie folgendermaßen vor:

1. Markieren Sie das Objekt, das Sie löschen möchten.
2. Um das Objekt zu löschen, haben Sie folgende Möglichkeiten:
 - Wählen Sie den Menübefehl **Bearbeiten > Löschen**.
 - Betätigen Sie die DEL-Taste.
3. Bestätigen Sie den Löschvorgang in der angezeigten Meldung durch Klicken auf die Schaltfläche "Ja".

3.4 Bedienung über Tastatureingabe

Internationale Tastenbezeichnung	Deutsche Tastenbezeichnung
HOME-Taste	POS1-Taste
END-Taste	ENDE-Taste
PAGE-UP-Taste	BILD-AUF-Taste
PAGE-DOWN-Taste	BILD-AB-Taste
CTRL-Taste	STRG-Taste
ENTER-Taste	Eingabetaste
DEL-Taste	ENTF-Taste
INSERT-Taste	EINFG-Taste

3.4.1 Tastenkombinationen für Menübefehle

Jeder Menübefehl kann über das Eingeben der entsprechenden ALT-Tastenkombination ausgelöst werden.

Drücken Sie nacheinander folgende Tasten:

- ALT-Taste
- Buchstabe, der im gewünschten Menü unterstrichen ist (z. B. ALT, D für Menü "Datei" - wenn Menü "Datei" in der Menüleiste eingetragen ist). Das Menü wird aufgeschlagen.
- Buchstabe, der im gewünschten Menübefehl unterstrichen ist (z. B. N für Menübefehl "Neu"). Handelt es sich um einen Menübefehl mit weiteren Untermenüs, so werden diese daraufhin aufgeschlagen. Verfahren Sie in gleicher Weise, bis Sie den gesamten Menübefehl durch Eingabe der jeweiligen Buchstaben ausgewählt haben.

Mit Eingabe des letzten Buchstabens der Tastenkombination wird der Menübefehl ausgelöst.

Beispiele:

Menübefehle	Tasten
Datei > Öffnen	ALT, D, O

Shortcuts für Menübefehle

Funktion		Shortcut
Neu > Baustein	(Menü "Datei")	CTRL+N
Speichern	(Menü "Datei")	CTRL+S
Schließen	(Menü "Datei")	CTRL+F4
Projekt Öffnen	(Menü "Datei")	CTRL+O
Laden in CPU	(Menü "Datei")	CTRL+L
Drucken	(Objekt) (Menü "Datei")	CTRL+P
Beenden	(Menü "Datei")	ALT+F4
Rückgängig	(Menü "Bearbeiten")	CTRL+Z
Wiederherstellen	(Menü "Bearbeiten")	CTRL+Y
Ausschneiden	(Menü "Bearbeiten")	CTRL+X
Kopieren	(Menü "Bearbeiten")	CTRL+C
Einfügen	(Menü "Bearbeiten")	CTRL+V
Löschen	(Menü "Bearbeiten")	DEL
Umbenennen	(Menü "Bearbeiten")	F2
Alles markieren	(Menü "Bearbeiten")	CTRL+A
Suchen/Ersetzen	(Menü "Bearbeiten")	CTRL+F
Gehe zu > Netzwerk/Zeile	(Menü "Bearbeiten")	CTRL+E
Gehe zu > Verwendungsstelle	(Menü "Bearbeiten")	CTRL+ALT+Q
Gehe zu > Vorheriger Fehler	(Menü "Bearbeiten")	ALT+F7
Gehe zu > Nächster Fehler	(Menü "Bearbeiten")	ALT+F8
Baustein öffnen	(Menü "Bearbeiten")	CTRL+ALT+O
Symbole	(Menü "Bearbeiten")	ALT+RETURN
Netzwerk	(Menü "Einfügen")	CTRL+R
Symbol	(Menü "Einfügen")	CTRL+J
Beobachten	(Menü "Test")	CTRL+F7
KOP	(Menü "Ansicht")	CTRL+1
FUP	(Menü "Ansicht")	CTRL+3
AWL	(Menü "Ansicht")	CTRL+2
Vergrößern	(Menü "Ansicht")	CTRL+Num+
Verkleinern	(Menü "Ansicht")	CTRL+Num-
Symbolische Darstellung	(Menü "Ansicht")	CTRL+Q
Symbolinformation	(Menü "Ansicht")	CTRL+SHIFT+Q
Symbolauswahl	(Menü "Ansicht")	CTRL+7
Kommentar	(Menü "Ansicht")	CTRL+SHIFT+K
CPU Bedienpanel	(Menü "Ansicht")	CTRL+ALT+C
Projektfenster	(Menü "Ansicht")	CTRL+ALT+P
Bibliotheken	(Menü "Ansicht")	CTRL+ALT+L

Funktion		Shortcut
Ansicht aktualisieren	(Menü "Ansicht")	F5
Einstellungen	(Menü "Extras")	CTRL+ALT+E
Baugruppenzustand	(Menü "Extras")	CTRL+D
Überlappend	(Menü "Fenster")	SHIFT+F5
Untereinander	(Menü "Fenster")	SHIFT+F2
Nebeneinander	(Menü "Fenster")	SHIFT+F3
HW Konfiguration	(Menü "Fenster")	CTRL+ALT+H
Symboltabelle	(Menü "Fenster")	CTRL+ALT+T
Beobachten/Steuern	(Menü "Fenster")	CTRL+ALT+W
Querverweise	(Menü "Fenster")	CTRL+ALT+X
STEP 7 Lite Hilfe	(Menü "Hilfe")	F1
Direkthilfe	(Menü "Hilfe")	SHIFT+F1 (Besteht ein aktueller Kontext, z. B. markierter Menübefehl, wird das entsprechende Hilfethema aufgerufen, andernfalls das Inhaltsverzeichnis der Hilfe.)
Aufruf Kontextmenü		SHIFT+F10

3.4.2 Tastenkombinationen für das Bewegen des Mauszeigers

Bewegen des Mauszeigers in der Menüleiste / im Kontextmenü

Funktion	Tasten
Zur Menüleiste	F10
Zum Aufruf des Kontextmenüs	SHIFT+F10
Zu dem Menü, das den unterstrichenen Buchstaben X enthält	ALT+X
Zugeordneter Menübefehl	Unterstrichener Buchstabe im Menübefehl
Einen Menübefehl nach links	PFEIL LINKS
Einen Menübefehl nach rechts	PFEIL RECHTS
Einen Menübefehl nach oben	PFEIL OBEN
Einen Menübefehl nach unten	PFEIL UNTEN
Aktiviert den markierten Menübefehl	ENTER
Verlässt das Menü bzw. zurück zum Text	ESC

Bewegen der Einfügemarke beim Bearbeiten von Texten

Funktion	Tasten
Eine Zeile nach oben oder ein Zeichen nach links in einem Text, der nur aus einer Zeile besteht	PFEIL OBEN
Eine Zeile nach unten oder ein Zeichen nach rechts in einem Text, der nur aus einer Zeile besteht	PFEIL UNTEN
Ein Zeichen nach rechts	PFEIL RECHTS
Ein Zeichen nach links	PFEIL LINKS
Ein Wort nach rechts	CTRL+PFEIL RECHTS
Ein Wort nach links	CTRL+PFEIL LINKS
Zum Anfang der Zeile	HOME
Zum Ende der Zeile	ENDE
Einen Bildschirm nach oben	PAGE-UP
Einen Bildschirm nach unten	PAGE-DOWN
Zum Textanfang	CTRL+HOME
Zum Textende	CTRL+END

Bewegen der Einfügemarke in Dialogfeldern

Funktion	Tasten
Zum nächsten Eingabefeld (von links nach rechts und von oben nach unten)	TAB
Ein Eingabefeld zurück	SHIFT+TAB
Zu dem Eingabefeld, das den unterstrichenen Buchstaben X enthält	ALT+X
Markiert in Auswahlliste	RICHTUNGSTASTEN
Öffnet eine Auswahlliste	ALT+PFEIL UNTEN
Markiert ein Objekt bzw. macht Markierung rückgängig	LEERTASTE
Bestätigt Einträge und schließt das Dialogfeld (Schaltfläche "OK")	ENTER
Schließt das Dialogfeld, ohne die Auswahl zu speichern (Schaltfläche "Abbrechen")	ESC

3.4.3 Tastenkombinationen für das Markieren von Texten

Funktion	Tasten
Ein Zeichen nach rechts	SHIFT+PFEIL RECHTS
Ein Zeichen nach links	SHIFT+PFEIL LINKS
Bis zum Anfang der Kommentarzeile	SHIFT+HOME
Bis zum Ende der Kommentarzeile	SHIFT+END
Eine Zeile nach oben	SHIFT+PFEIL OBEN
Eine Zeile nach unten	SHIFT+PFEIL UNTEN
Einen Bildschirm nach oben	SHIFT+PAGE-UP
Einen Bildschirm nach unten	SHIFT+PAGE-DOWN
Bis zum Dateianfang	CTRL+SHIFT+HOME
Bis zum Dateiende	CTRL+SHIFT+END

3.4.4 Tastenkombinationen für den Zugriff auf die Online-Hilfe

Funktion	Tasten
Aktivieren des Fragezeichen-Symbols für Direkthilfe	SHIFT+F1
Öffnen der Hilfe zu STEP 7 Lite	F1
Schließen des Hilfenfensters	ALT+F4

3.4.5 Tastenkombinationen für das Umschalten zwischen Fenstertypen

Funktion	Tasten
Umschalten zwischen Teilfenstern	F6
Zurückschalten auf das vorherige Teilfenster, wenn kein andockbares Fenster vorhanden ist	SHIFT+F6
Umschalten zwischen Dokumentfenster und andockbarem Fenster des Dokuments (z. B. Variablendeklarationsfenster) Wenn es keine andockbaren Fenster gibt, wird auf das vorherige Teilfenster zurückgeschaltet.	SHIFT+F6
Umschalten zwischen Dokumentfenstern	CTRL+F6
Zurückschalten auf das vorherige Dokumentfenster	SHIFT+CTRL+F6
Umschalten zwischen Nichtdokumentfenstern (Applikationsrahmen und andockbare Fenster des Applikationsrahmens; bei Rückkehr zum Rahmen wird zu dem Dokumentfenster umgeschaltet, das zuletzt aktiv war)	ALT+F6
Umschalten zwischen Projektfenster, CPU-Bedienpanel, Bibliotheksfenster und aktivem Dokumentfenster	CTRL+ALT+F6
Zurückschalten auf das vorherige Nichtdokumentfenster	SHIFT+ALT+F6
Schließen des aktiven Fensters	CTRL+F4

3.5 Bedienung über Teleservice

Das Optionspaket TeleService bietet die Möglichkeit, von einem PG/PC aus über das Telefonnetz eine Online-Verbindung zu einer entfernten Anlage aufzubauen. Diese Anlage können Sie dann wie gewohnt mit STEP 7 Lite bearbeiten.

Wegen der längeren Reaktionszeiten ist diese Art der Bedienung nur zu Service-Zwecken zu empfehlen.

Voraussetzungen

Voraussetzungen für das Arbeiten mit TeleService sind:

- Das Optionspaket TeleService muss installiert sein.
- Unter Windows muss ein lokales Modem installiert und eingerichtet sein.
- Die entfernte Anlage muss über einen entsprechend parametrisierten TS Adapter und ein Modem an ein Telefonnetz angeschlossen sein.
- Mit "PG/PC-Schnittstelle einstellen" müssen Sie den Zugangspunkt für Applikationen und dessen Eigenschaften auf den TS Adapter einstellen.

Aufruf

Wenn das Optionspaket installiert ist, können Sie TeleService über den Menübefehl **Extras > Optionspakete > TeleService** aufrufen.

Hinweis

Weitere Information finden Sie in der Dokumentation und in der Online-Hilfe des Optionspakets.

4 Einrichten und Bearbeiten des Projekts

4.1 Was ist ein STEP 7 Lite Projekt

Die Projektdaten eines STEP 7 Lite Projekts umfassen alle Daten einer SIMATIC S7-300, C7 oder einem modularen Dezentralen Peripheriesystem ET 200X bzw. ET 200S (stand alone).

Projekte dienen dazu, die bei der Erstellung einer Automatisierungslösung anfallenden Daten geordnet abzulegen. In einem STEP 7 Lite-Projekt sind die Daten für eine Station zusammengefasst; insbesondere:

- Die Konfigurationsdaten über den Hardware-Aufbau und Parametrierungsdaten für Baugruppen der Station
- Die Symboltabelle der Station
- Die Variablen tabellen zum Beobachten/Steuern der Station
- Die Projektdokumentation mit Angaben zum Inhalt und zur Form der Dokumentation.
- Das Anwenderprogramm der Station




Projektfenster (Register "Projekt" im Vordergrund)

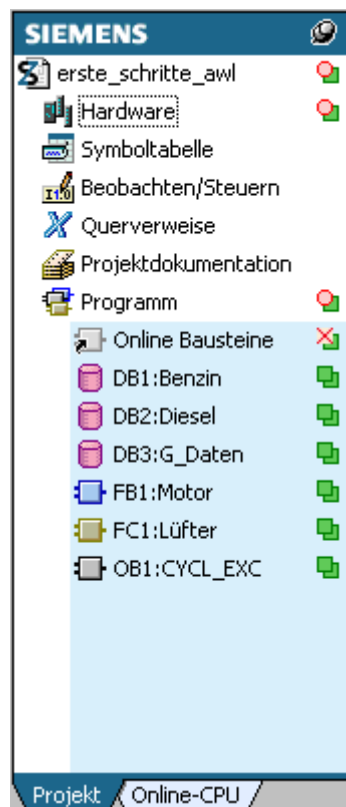
Das Projektfenster zeigt im Register "Projekt":

- Das Projektsymbol mit dem Namen des aktuellen Projekts. Mit einem Doppelklick wird eine Übersicht der Bausteine und der Hardwarekonfiguration angezeigt.
- Das Hardware symbol - mit einem Doppelklick gelangen Sie in eine Sicht zur Parametrierung der Hardware-Konfiguration, zur Hardware-Diagnose und zum Hardware-Vergleich.
- Das Symbol der Symboltabelle - mit einem Doppelklick gelangen Sie in eine Sicht zum Anlegen und Verwalten der Projektsymbole.
- Das Symbol zum "Beobachten/Steuern" - mit einem Doppelklick gelangen Sie in eine Sicht zum Beobachten/Steuern bzw. Beobachten/Forcen von Variablen.
- Das Querverweissymbol - über Doppelklicken gelangen Sie in eine Sicht zur Anzeige der Querverweise, der verwendeten Operanden und der Programmstruktur.
- Das Symbol der Projektdokumentation - mit dem Doppelklick gelangen Sie in eine Sicht zum Zusammenstellen und Gestalten Ihrer STEP 7 Lite Projektdokumentation.

- Ein Symbol für das Anwenderprogramm, unter dem die Bausteine des Projekts angeordnet sind. Ein Doppelklick auf das Programmsymbol öffnet eine Übersicht aller im Projekt vorhandener Bausteine. Der Doppelklick auf ein Bausteinsymbol öffnet den Baustein im passenden Bausteineditor.

Besteht eine Onlineverbindung zu einer CPU, so werden neben den Symbolen zum Projekt, der Hardware und dem Programm mit seinen Bausteinen Zusatzsymbole mit folgender Bedeutung eingeblendet.

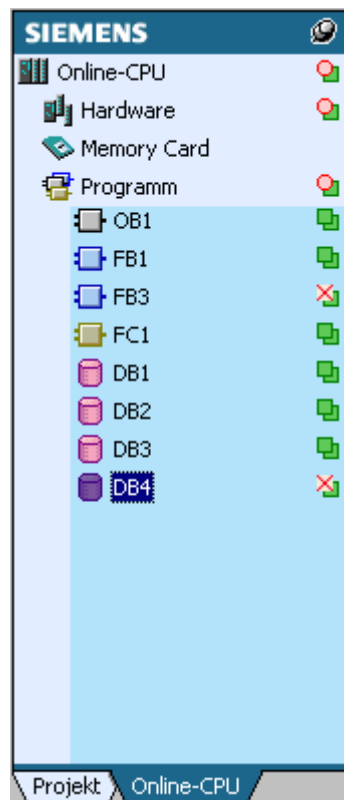
Symbol	Bedeutung
	Objekt ist im Projekt und der Online-CPU gleich.
	Das Objekt im Projekt stimmt nicht mit dem Objekt in der Online-CPU überein. Weitere Hilfen erhalten Sie über den Hilfezeiger in STEP 7 Lite.
	Das Objekt ist in der aktuellen Sicht des Projektfensters vorhanden, fehlt aber in der nicht angewählten Sicht (Sicht "Projekt" / Sicht "Online-CPU"). Weitere Hilfen erhalten Sie über den Hilfezeiger in STEP 7 Lite.



Projektfenster (Register "Online-CPU" im Vordergrund)

Das Projektfenster zeigt bei einer Onlineverbindung zu einer CPU im Register "Online-CPU":

- Das Symbol für die "Online-CPU". Mit einem Doppelklick wird eine Übersicht der auf der CPU abgelegten Bausteine, der Hardwarekonfiguration und eventuell ein Sammeleintrag für Dateien auf einer SIMATIC Micro Memory Card (MMC) angezeigt. Der Sammeleintrag wird nur beim Einsatz einer CPU 31xC mit gesteckter MMC angezeigt.
- Das Hardware-symbol - mit einem Doppelklick gelangen Sie in eine Sicht zur Anzeige der auf der CPU abgelegten Hardware-Konfiguration und zur Hardware-Diagnose.
- Das Symbol einer Memory Card. Das Symbol ist nur beim Einsatz einer CPU 31xC mit gesteckter Micro Memory Card sichtbar. Mit einem Doppelklick gelangen Sie zu einer Übersicht der auf der MMC gespeicherten Dateien.
- Ein Symbol für das Anwenderprogramm, unter dem die Bausteine der CPU angeordnet sind. Ein Doppelklick auf das Programmsymbol öffnet eine Übersicht aller auf der CPU vorhandenen Bausteine (je nach Einstellung in **Extras > Einstellungen > Systembausteine anzeigen**, auch mit Systembausteinen). Ein Doppelklick auf ein Bausteinsymbol öffnet den Baustein schreibgeschützt im passenden Bausteineditor.



4.2 Einrichten eines Projekts

4.2.1 Anlegen eines Projekts

Um Ihre Automatisierungsaufgabe im Rahmen einer Projektverwaltung zu lösen, legen Sie zunächst ein neues Projekt an.

1. Wählen Sie den Menübefehl **Datei > Neu**.
Ein neues Projekt wird mit vorbelegten Inhalten im Projektfenster (z. B. Symbol für die Hardware-Konfiguration) angelegt.
Wenn bereits ein Projekt geöffnet ist, wird es zunächst geschlossen und anschließend das neue Projekt geöffnet.
2. Das neue Projekt erhält den Namen "Neues Projekt". Legen Sie den endgültigen Namen des Projekts beim Speichern (Speichern/Speichern unter) des Projekts fest.

4.2.2 Einfügen eines Programms

Bereits angelegte Komponenten

Wenn Sie ein Projekt angelegt haben, sind die Komponenten "Hardware-Konfiguration", "Symboltabelle" und "Programm" bereits angelegt.

Einfügen von Bausteinen

1. Wählen Sie den Menübefehl **Einfügen > Baustein....**
2. Im anschließend aufgeblendeten Dialogfeld "Neuer Baustein" können Sie die Art des Bausteins (z. B. Funktion) und seine Eigenschaften bestimmen (z. B. symbolischer Name).

Gruppieren von Bausteinen

Zur besseren Übersicht können Sie bei einer größeren Anzahl von Bausteinen Kategorien (Zwischenüberschriften) einfügen und die Bausteine per Drag&Drop in diesen Kategorien sortieren.

1. Wählen Sie den Menübefehl **Einfügen > Kategorie**.
2. Benennen Sie die Kategorie.
3. Verschieben Sie passende Bausteine unter die neue Kategorie.

Verwenden von Bausteinen aus Bausteinbibliotheken

Für die Erstellung von Anwenderprogrammen können Sie auch Bausteine aus den Standardbibliotheken benutzen, die im Lieferumfang enthalten sind. Diese Bausteine finden Sie im Bibliotheksfenster unter dem Register "Bausteine". Wird das Bibliotheksfenster nicht angezeigt, wählen Sie den Menübefehl **Ansicht > Anpassen > Bibliotheken**. Weitere Hinweise zu den Standardbibliotheken finden Sie in Übersicht der Bausteinbibliotheken.

Erstellen einer Symboltabelle

Eine (leere) Symboltabelle (Objekt "Symboltabelle") wird beim Erzeugen des Programms automatisch angelegt. Wenn Sie das Objekt "Symboltabelle" wählen, wird das Fenster "Symboltabelle" aufgeblendet und darin die Symboltabelle angezeigt. Weitere Hinweise finden Sie in Eingeben mehrerer globaler Symbole in der Symboltabelle.

4.3 Bearbeiten eines Projekts

Öffnen eines Projekts

Um ein bereits bestehendes, bereits gespeichertes Projekt zu öffnen, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Datei > Öffnen....**
2. Wählen Sie ein Projekt aus:
 - Projekte, die an diesem PG/PC bereits bearbeitet wurden, finden Sie im Register "Zuletzt bearbeitete".
 - Projekte, die Sie noch nicht bearbeitet haben, finden Sie im Register "Dateisystem". Wählen Sie hierzu in der Explorer-Sicht des Registers den gewünschten Pfad und das gewünschte Projekt.
 - Wurde das Projekt auf der Micro Memory Card einer CPU 31xC gespeichert, so finden Sie das Projekt im Register "Memory Card". Voraussetzung ist, dass zur CPU 31xC eine Onlineverbindung besteht. Um die längeren Übertragungszeiten beim Zwischenspeichern zu umgehen, wird das Projekt als "Kopie" geöffnet.

Kopieren eines Projekts


Ein Projekt kopieren Sie, indem Sie dieses über den Menübefehl **Datei > Speichern unter** unter einem anderen Namen abspeichern.

Projektteile wie Bausteine kopieren Sie mit dem Menübefehl **Bearbeiten > Kopieren** in die Zwischenablage.



Eine Schritt-für-Schritt-Anleitung zum Kopieren eines Projekts finden Sie in **Kopieren eines Projekts und Kopieren eines Projektteils**.

4.3.1 Änderungen übernehmen und speichern

Wenn Sie z. B. Anweisungen im Anwenderprogramm editiert haben, werden Sie bereits durch ein Symbol im Projektfenster darauf aufmerksam gemacht, dass Änderungen gegenüber dem gespeicherten Stand vorliegen, diese Änderungen aber noch nicht gespeichert sind. Sie haben die Möglichkeit, die Änderungen zu übernehmen oder zu speichern.

	Symbol für geänderte aber noch nicht übernommene oder gespeicherte Änderungen (Beispiel)
---	--

Unterschied zwischen "Übernehmen" und "Speichern"

	Übernehmen	Die Änderungen werden zwischengespeichert, so dass die Projektdaten konsistent und in allen geöffneten Fenstern aktuell sind. Wenn Sie das Projekt schließen ohne zu speichern, sind diese Änderungen verloren, da die zwischengespeicherten Daten nach Schließen des Projekts gelöscht werden.
	Speichern	Die Änderungen werden dauerhaft in der Projektdatei gespeichert und stehen beim nächsten Öffnen des Projekts wieder zur Verfügung. Bei der Erstellung neuer Filter für die Symboltabelle oder für die Querverweisliste ist ein vorheriges Übernehmen notwendig, bevor sie gespeichert werden können. Die Filter werden ebenfalls mit allen sonstigen Änderungen im Projekt gespeichert.

Empfohlene Vorgehensweise

Wählen Sie "Übernehmen", wenn der gespeicherte Stand des Projekts nicht überschrieben werden soll (weil z. B. die durchgeführten Änderungen nur vorläufig oder noch nicht vollständig sind und nochmals angepasst werden).

Wählen Sie "Speichern", wenn der gespeicherte Stand des Projekts aktualisiert werden soll. Es werden **alle** Änderungen des Projekts gespeichert. Bei "Speichern unter" wird der aktuelle Projektstand unter einem neuen Namen bzw. unter einem neuen Pfad abgespeichert.

Besonderheiten bei Vorlagen für Projektdokumentation

Vorlagen für die Projektdokumentation werden in einer separaten Datei unabhängig vom Projekt gespeichert (*.k7d). Für jedes Projekt sind sie dann über die Funktion "Laden" verfügbar.

Die Einstellungen für die Projektdokumentation werden zusammen mit den übrigen Daten des Projekts gespeichert.

Besonderheiten beim Arbeiten mit Filtern

Für die Symboltabelle oder für die Querverweisliste können Sie einen neuen Filter erstellen oder einen erstellten Filter abändern. Hierzu klicken Sie auf die Schaltfläche "Filter bearbeiten", um im Folgedialog Bezeichnung und Einstellungen für den Filter zu bearbeiten.

Neben "Abbrechen" (Verlassen des Dialogs ohne Änderungen) haben Sie folgende Möglichkeiten, den Dialog zu verlassen:

Klick auf Schaltfläche "Übernehmen"	Die Einstellungen für den Filter werden zwischengespeichert, so dass der Filter nach Verlassen des Dialogs in der Klappliste "Filter" ausgewählt werden kann. Wenn Sie das Projekt speichern, ist der Filter automatisch mitgespeichert. Wenn Sie das Projekt anschließend nicht speichern, steht der Filter beim nächsten Öffnen des Projekts nicht mehr zur Verfügung.
Klick auf Schaltfläche "Filtern" (ohne vorheriges Übernehmen)	Die Einstellungen für den Filter werden zwischengespeichert, so dass der Filter nach Verlassen des Dialogs in der Klappliste "Filter" ausgewählt werden kann. Allerdings wird der Name des Filters mit einem Sternchen (*). Das Sternchen deutet an, dass dieser Filter beim Speichern des Projekts nicht mitgespeichert wird! Diese Vorgehensweise empfiehlt sich, wenn sie den Filter später nicht mehr benötigen und daher nicht mehr zur Auswahl in der Klappliste "Filter" angeboten werden soll.

4.3.2 Schritte zum Bearbeiten von Projekten

4.3.2.1 Kopieren eines Projekts

Wenn Sie ein Projekt kopieren möchten, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Datei > Speichern unter**.
2. Tragen Sie im Dialogfeld "Speichern unter" den Namen des neuen Projekts und ggf. einen neuen Ablagepfad ein. Bestätigen Sie mit "OK".

4.3.2.2 Kopieren eines Projektteils

Wenn Sie ein Projektteil wie z. B. Bausteine in ein anderes Projekt kopieren möchten, gehen Sie folgendermaßen vor:

Kopieren mit Menübefehlen

1. Öffnen Sie STEP 7 Lite ein zweites mal.
2. Öffnen Sie in einem STEP 7 Lite das Quellprojekt, im anderen das Zielprojekt.
3. Markieren Sie im Projektfenster der Quelle das zu kopierende Projektteil.
4. Wählen Sie in STEP 7 Lite der Quelle den Menübefehl **Bearbeiten > Kopieren**.
5. Wechseln Sie in STEP 7 Lite des Ziels und wählen Sie den Menübefehl **Bearbeiten > Einfügen**.

Kopieren direkt mit der Maus (Drag&Drop)

1. Öffnen Sie STEP 7 Lite ein zweites mal.
2. Öffnen Sie in einem STEP 7 Lite das Quellprojekt, im anderen das Zielprojekt.
3. Markieren Sie im Projektfenster der Quelle das zu kopierende Projektteil, das Sie kopieren möchten, und halten Sie die linke Maustaste gedrückt.
4. Ziehen Sie das Objekt in das Projektfenster des Zielbereichs und lassen es durch freigegeben der linken Maustaste an der korrespondierenden Position fallen.

Gehen Sie in der gleichen Reihenfolge für alle zu kopierenden Projektteile vor. Schließen Sie nach Abschluss der Kopierarbeiten das STEP 7 Lite des Quellprojekts.

4.3.2.3 Konfigurieren der Hardware (prinzipiell)

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf das Objekt "Hardware", um den Arbeitsbereich zum Konfigurieren der Hardware zu öffnen.
2. Wählen Sie den Stationstyp (z. B. S7-300), um in der passende Sicht Baugruppen zu konfigurieren und zu parametrieren.
3. Wählen Sie Baugruppen aus, die Sie unter Berücksichtigung der Steckplatzregeln auf die Steckplätze verteilen.
4. Stellen Sie, falls erforderlich, die Baugruppenparameter für die einzelnen Baugruppen ein.

4.3.2.4 Erstellen der Software im Projekt (prinzipiell)

Um die Software für Ihr Projekt zu erstellen, gehen Sie folgendermaßen vor:

1. Markieren Sie das Objekt "Symbole" und legen Sie die Symbole fest. (Dieser Schritt ist auch zu einem späteren Zeitpunkt durchführbar.)
2. Fügen Sie neue Bausteine ein (Menübefehl **Einfügen > Baustein**) und editieren Sie sie im Arbeitsbereich.
3. Doppelklicken Sie das Symbol für die Projektdokumentation im Projektfenster. Bestimmen Sie in der Sicht "Projektdokumentation" die Druckobjekte und nehmen Sie die Einstellungen für den Druck vor oder wählen Sie eine Dokumentationsvorlage und drucken das Projekt über die Schaltfläche "Dokumentation drucken" aus.

4.4 Löschen und Umbenennen eines Projekts

Verwenden Sie den Dateexplorer Ihres Betriebssystems, um ein Projekt zu löschen oder umbenennen. Ihr Projekt erkennen Sie an der Namensweiterung ".k7p".

Achten Sie beim Löschen und Umbenennen eines Projekts darauf, dass das Projekt nicht in STEP 7 Lite geöffnet ist.

5 Konfigurieren der Hardware

5.1 Grundlagen zum Hardware-Konfigurieren mit STEP 7 Lite

5.1.1 Einführung zum Konfigurieren der Hardware

Konfigurieren

Unter "Konfigurieren" verstehen wir im Folgenden das Anordnen von Baugruppen und Schnittstellenmodulen in einer grafischen Sicht, die einen Stationsaufbau (z. B. S7-300) repräsentiert.

Baugruppenträger werden sowohl grafisch als auch zusätzlich durch eine Konfigurationstabelle repräsentiert, die wie der "reale" Baugruppenträger eine festgelegte Anzahl von steckbaren Baugruppen zulässt. Die Konfigurationstabelle enthält mehr Informationen zu den Baugruppen (z. B. genaue Bezeichnungen und Adressen).

Sie können die Konfiguration in andere STEP 7 Lite-Projekte kopieren, gegebenenfalls modifizieren und in eine oder mehrere existierende Stationen laden. Beim Anlauf des Automatisierungssystems vergleicht die CPU die mit STEP 7 Lite erstellte Sollkonfiguration mit der tatsächlichen Istkonfiguration der Anlage. Eventuelle Fehler können somit sofort erkannt und gemeldet werden.

Parametrieren

Unter "Parametrieren" verstehen wir im Folgenden das Einstellen von Eigenschaften bei parametrierbaren Baugruppen für den zentralen Aufbau.

Beispiel: Eine CPU ist eine parametrierbare Baugruppe. Die Zyklusüberwachungszeit ist ein Parameter, den Sie einstellen können.

Die Parameter werden in die CPU geladen und von der CPU an die entsprechenden Baugruppen übertragen. Baugruppen lassen sich sehr einfach ersetzen, da die mit STEP 7 Lite erstellten Parameter im Anlauf automatisch in die neue Baugruppe geladen werden.

Wann ist "Hardware konfigurieren" erforderlich?

Die Eigenschaften der S7-Automatisierungssysteme und der Baugruppen sind so voreingestellt, dass Sie in vielen Fällen nicht konfigurieren müssen.

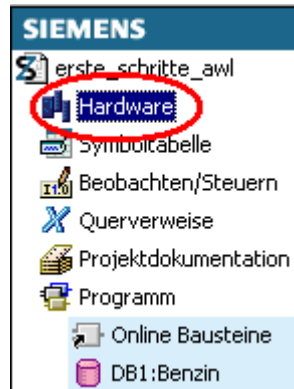
Das Konfigurieren ist zwingend erforderlich, wenn Sie voreingestellte Parameter einer Baugruppe verändern möchten (z. B. Prozessalarm bei einer Baugruppe freigeben).

5.1.2 Grundsätzliche Bedienung beim Hardware konfigurieren

Starten der Hardware-Konfiguration

Wenn Sie ein neues Projekt angelegt haben, öffnen Sie folgendermaßen den Arbeitsbereich für das Konfigurieren und Parametrieren von Baugruppen:

- Doppelklicken Sie auf das Symbol "Hardware"



Arbeitsbereich für das Konfigurieren

Der Arbeitsbereich für das Konfigurieren eines Automatisierungssystems besteht aus folgenden Bereichen:

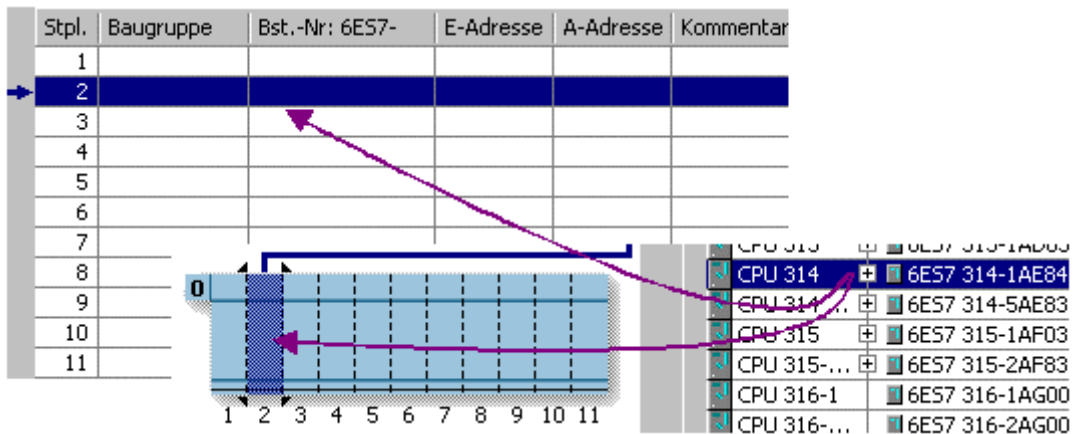
- Grafische Übersicht, in der die Baugruppenträger mit Baugruppen realitätsnah dargestellt sind.
- Tabelle(n), die einzelne Baugruppenträger repräsentieren, im Unterschied zur grafischen Übersicht aber weitere Informationen zu den Baugruppen enthalten (Bestellnummer, Adressen, etc.).
- Der "Hardware Katalog", aus dem Sie die benötigten Hardwarekomponenten auswählen, z. B. Baugruppen und Schnittstellenmodule.

5.1.2.1 Grundsätzliche Schritte beim Konfigurieren einer Station

Unabhängig davon, welche Aufbautechnik eine Station hat - Sie konfigurieren immer mit folgenden Schritten:

1. Markieren Sie eine Hardwarekomponente im "Hardware Katalog".
2. Ziehen Sie die ausgewählte Komponente per Drag&Drop
 - auf einen Steckplatz des Baugruppenträgers in der grafischen Sicht oder
 - auf eine Zeile in der Konfigurationstabelle, die den Aufbau des Baugruppenträgers repräsentiert

Das folgende Bild zeigt die grundsätzliche Bedienung:

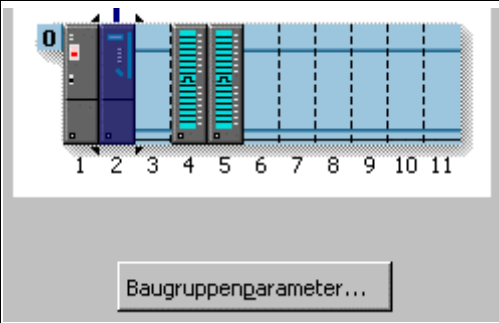


5.1.2.2 Aufbau der Sicht 'HW-Konfiguration'

Die Sicht "HW-Konfiguration" zeigt zwei Ansichten der aktuellen Stationskonfiguration:

- Die grafische Sicht mit realitätsnaher Anordnung der Baugruppen auf ihren Steckplätzen
- Die tabellarische Sicht mit Detailinformationen zu den gesteckten Baugruppen (z. B. Adressen und Bestellnummern)

Außerdem ist in dieser Sicht der Katalog mit den Komponenten sichtbar, aus dem die Baugruppenträger ('Racks') bestückt werden müssen.

Bereich in HW-Konfig-Sicht		Bedeutung																																																		
		<p>Grafische Sicht der HW-Konfiguration mit markiertem Steckplatz/markierter Baugruppe.</p> <p>Von der markierten Baugruppe aus geht ein Pfeil auf den gleichwertigen Steckplatz in der tabellarischen Sicht der HW-Konfiguration.</p> <p>Über die Schaltfläche "Baugruppenparameter" können Sie bei parametrierbaren Baugruppen (z. B. wie hier die CPU) den Dialog zum Parametrieren öffnen.</p>																																																		
<table border="1" data-bbox="240 1189 1018 1429"> <thead> <tr> <th colspan="2">Rack 0</th> <th colspan="2">Rack 1</th> <th colspan="2">Rack 2</th> <th colspan="2">Rack 3</th> </tr> <tr> <th>Stpl.</th> <th>Baugrup</th> <th>Bestellnummer</th> <th>E-Adresse</th> <th>A-Adresse</th> <th colspan="2">Kommentar</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>PS 307...</td> <td>6ES7 307-1BA00-...</td> <td></td> <td></td> <td colspan="2"></td> </tr> <tr> <td>2</td> <td>CPU 315</td> <td>6ES7 315-1AF03-...</td> <td></td> <td></td> <td colspan="2"></td> </tr> <tr> <td>3</td> <td></td> <td></td> <td></td> <td></td> <td colspan="2"></td> </tr> <tr> <td>4</td> <td>SM 32...</td> <td>6ES7 321-1BH00-...</td> <td>0 ... 1</td> <td></td> <td colspan="2"></td> </tr> <tr> <td>5</td> <td>SM 32...</td> <td>6ES7 322-1BH00-...</td> <td></td> <td>4 ... 5</td> <td colspan="2"></td> </tr> </tbody> </table>		Rack 0		Rack 1		Rack 2		Rack 3		Stpl.	Baugrup	Bestellnummer	E-Adresse	A-Adresse	Kommentar		1	PS 307...	6ES7 307-1BA00-...					2	CPU 315	6ES7 315-1AF03-...					3							4	SM 32...	6ES7 321-1BH00-...	0 ... 1				5	SM 32...	6ES7 322-1BH00-...		4 ... 5			<p>Markierter Steckplatz/markierte Baugruppe in der tabellarischen Sicht auf die Hardware-Konfiguration.</p> <p>Über die Register am oberen Rand kann auf verschiedene Baugruppenträger zugegriffen werden.</p>
Rack 0		Rack 1		Rack 2		Rack 3																																														
Stpl.	Baugrup	Bestellnummer	E-Adresse	A-Adresse	Kommentar																																															
1	PS 307...	6ES7 307-1BA00-...																																																		
2	CPU 315	6ES7 315-1AF03-...																																																		
3																																																				
4	SM 32...	6ES7 321-1BH00-...	0 ... 1																																																	
5	SM 32...	6ES7 322-1BH00-...		4 ... 5																																																

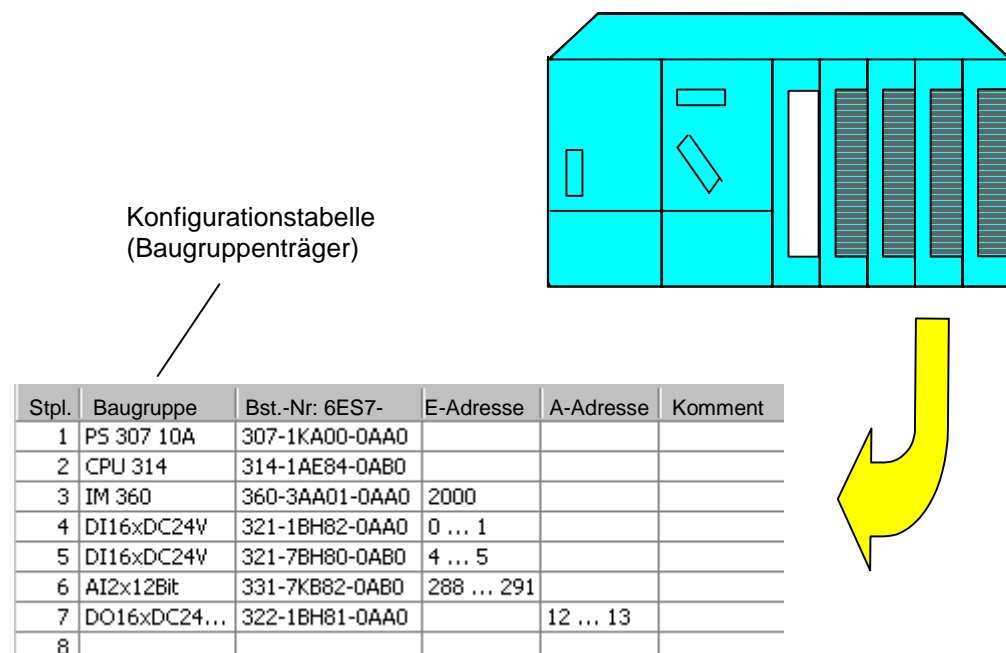
Bereich in HW-Konfig-Sicht	Bedeutung																																	
<div style="border: 1px solid gray; padding: 5px;"> <input type="checkbox"/> Baugruppeninformation anzeigen <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 20%;">Baugruppe</th> <th style="width: 60%;">Bestellnummer</th> <th style="width: 20%;"></th> </tr> </thead> <tbody> <tr> <td>CPU 312 ...</td> <td>6ES7 312-5AC82-0AB0</td> <td>1.2</td> </tr> <tr> <td>CPU 312 C</td> <td>6ES7 312-5BD00-0AB0</td> <td></td> </tr> <tr> <td>CPU 313</td> <td>6ES7 313-1AD03-0AB0</td> <td>1.2</td> </tr> <tr> <td>CPU 313 C</td> <td>6ES7 313-5BE00-0AB0</td> <td></td> </tr> <tr> <td>CPU 313 ...</td> <td>6ES7 313-6CE00-0AB0</td> <td></td> </tr> <tr> <td>CPU 313 ...</td> <td>6ES7 313-6BE00-0AB0</td> <td></td> </tr> <tr> <td>CPU 314</td> <td>6ES7 314-1AE84-0AB0</td> <td>1.2</td> </tr> <tr> <td>CPU 314 ...</td> <td>6ES7 314-6CF00-0AB0</td> <td></td> </tr> <tr> <td>CPU 314 ...</td> <td>6ES7 314-6BF00-0AB0</td> <td></td> </tr> <tr> <td>CPU 314</td> <td>6ES7 314-5AE83-0AB0</td> <td>1.2</td> </tr> </tbody> </table> </div>	Baugruppe	Bestellnummer		CPU 312 ...	6ES7 312-5AC82-0AB0	1.2	CPU 312 C	6ES7 312-5BD00-0AB0		CPU 313	6ES7 313-1AD03-0AB0	1.2	CPU 313 C	6ES7 313-5BE00-0AB0		CPU 313 ...	6ES7 313-6CE00-0AB0		CPU 313 ...	6ES7 313-6BE00-0AB0		CPU 314	6ES7 314-1AE84-0AB0	1.2	CPU 314 ...	6ES7 314-6CF00-0AB0		CPU 314 ...	6ES7 314-6BF00-0AB0		CPU 314	6ES7 314-5AE83-0AB0	1.2	<div style="border: 1px solid gray; padding: 5px; margin-top: 5px;"> System PS CPU IM SM AI SM AO SM AI/O SM DI SM DO </div> <p>Katalog mit Baugruppenliste.</p> <p>Wenn Sie auf das Symbol "+" vor der Bestellnummer klicken, wird eine Liste von Baugruppen gleichen Typs, aber unterschiedlicher Bestellnummer angezeigt. An oberster Stelle steht immer die aktuellste Baugruppe.</p> <p>Bei Baugruppen mit unterschiedlichen Betriebssystem-Versionen (Firmware) kann auf eine bestimmte Version über eine Klappliste zugegriffen werden.</p> <p>Über Register am seitlichen Rand des Katalogs kann auf unterschiedliche Baugruppenkategorien zugegriffen werden.</p>
Baugruppe	Bestellnummer																																	
CPU 312 ...	6ES7 312-5AC82-0AB0	1.2																																
CPU 312 C	6ES7 312-5BD00-0AB0																																	
CPU 313	6ES7 313-1AD03-0AB0	1.2																																
CPU 313 C	6ES7 313-5BE00-0AB0																																	
CPU 313 ...	6ES7 313-6CE00-0AB0																																	
CPU 313 ...	6ES7 313-6BE00-0AB0																																	
CPU 314	6ES7 314-1AE84-0AB0	1.2																																
CPU 314 ...	6ES7 314-6CF00-0AB0																																	
CPU 314 ...	6ES7 314-6BF00-0AB0																																	
CPU 314	6ES7 314-5AE83-0AB0	1.2																																

5.1.2.3 Konfigurationstabelle als Abbild eines Baugruppenträgers

Für den zentralen Aufbau ordnen Sie die Baugruppen neben der CPU auf einem Baugruppenträger und fortführend auf weiteren Baugruppenträgern an. Die Anzahl der bestückbaren Baugruppenträger richtet sich nach der verwendeten CPU.

Genau wie in Ihrer realen Anlage platzieren Sie mit STEP 7 Lite Baugruppen auf Baugruppenträgern. Der Unterschied: In STEP 7 Lite werden Baugruppenträger zusätzlich durch "Konfigurationstabellen" repräsentiert, die soviel Zeilen haben wie Baugruppen auf dem realen Baugruppenträger steckbar sind.

Das folgende Bild zeigt an einem Beispiel die Umsetzung eines realen Aufbaus in die Konfigurationstabelle. Die Konfigurationstabelle entspricht dem verwendeten Baugruppenträger.



5.1.2.4 Festlegen der Eigenschaften von Komponenten

Wenn Sie Komponenten in der HW-Konfiguration-Sicht angeordnet haben, kommen Sie immer auf folgende Weise in einen Dialog zum Ändern der voreingestellten Eigenschaften (Parameter):

- Doppelklicken Sie auf die Komponente oder wählen Sie den Menübefehl **Bearbeiten > Baugruppenparameter**.
- Mit rechter Maustaste: Bewegen Sie den Mauszeiger auf die Komponente, drücken die rechte Maustaste und wählen aus dem Kontextmenü den Befehl **Baugruppenparameter**.
- Klicken Sie auf die Schaltfläche "Baugruppenparameter" unterhalb der grafischen Übersicht.

Eigenschaften von Zentralbaugruppen

Von besonderer Bedeutung für das Systemverhalten sind die Eigenschaften der CPUs. Im Parametrierdialog einer CPU können Sie z. B. Anlaufverhalten, Speicherbereiche, Remanenzverhalten, Taktmerker, Schutzstufe und Passwort einstellen, um nur einige Eigenschaften zu nennen. Was einstellbar ist und in welchen Wertebereichen, das "weiß" STEP 7 Lite.

Weitere Möglichkeiten zur Parametrierung

Bei einigen Baugruppen haben Sie die Möglichkeit, Parameter im Anwenderprogramm einzustellen (z. B. bei Analogbaugruppen). Dafür rufen Sie im Anwenderprogramm die Systemfunktionen (SFCs) WR_PARM, WR_DPARM und PARM_MOD auf. Diese Einstellungen gehen aber beim Anlauf (Neustart (Warmstart)) verloren.

Genauere Informationen über die Systemfunktionen finden Sie im Referenzhandbuch *Systemsoftware für S7-300/400, System- und Standardfunktionen*.

5.1.2.5 Wissenswertes zu Steckplatz- und sonstigen Regeln

STEP 7 Lite unterstützt Sie beim Konfigurieren einer Station, so dass Sie in der Regel unmittelbar Rückmeldung darüber bekommen, wenn z. B. eine Baugruppe nicht auf dem gewünschten Steckplatz steckbar ist.

Achten Sie auf eingblendete Meldungen, die über Ursachen und Folgen einer Bedienung Aufschluss geben. Zusätzlich haben Sie die Möglichkeit, über die Hilfe zu den Meldungen weitere Informationen zu erhalten.

Nicht berücksichtigt werden zusätzliche, nur temporär (für einen bestimmten Ausgabestand) gültige Regeln, wie z. B. Einschränkungen der nutzbaren Steckplätze aufgrund einer Funktionseinschränkung bei einzelnen Baugruppen. Beachten Sie daher immer die Dokumentation bzw. die aktuelle Produktinformation zu den Baugruppen.

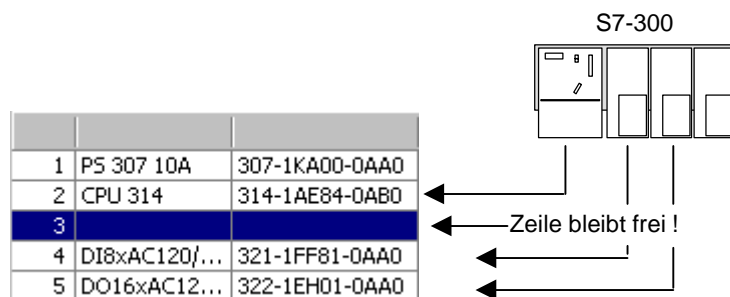
5.2 Konfigurieren der Baugruppen

5.2.1 Regeln für die Anordnung von Baugruppen (SIMATIC 300)

Grundregel

Baugruppen müssen lückenlos aneinandergereiht werden.

Ausnahme: Bei Aufbau mit einem Baugruppenträger müssen Sie einen Steckplatz der Konfigurationstabelle freilassen (reserviert für Anschaltungsbaugruppe). Das ist bei S7-300 der Steckplatz 3. Im tatsächlichen Aufbau haben Sie keine Lücke, da der Rückwandbus sonst unterbrochen wäre!



Steckplatzregeln (S7-300)

Baugruppenträger 0:

- Steckplatz 1: Nur Stromversorgung (z. B. 6ES7 307-...) oder leer
- Steckplatz 2: Nur Zentralbaugruppe (z. B. 6ES7 314-...)
- Steckplatz 3: Anschaltungsbaugruppe (z. B. 6ES7 360-.../361-...) oder leer
- Steckplatz 4 bis 11: Signal- oder Funktionsbaugruppen, Kommunikationsprozessoren oder leer

Baugruppenträger 1 bis 3:

- Steckplatz 1: Nur Stromversorgungsbaugruppe (z. B. 6ES7 307-...) oder leer
- Steckplatz 2: leer
- Steckplatz 3: Anschaltungsbaugruppe
- Steckplatz 4 bis 11: Signal- oder Funktionsbaugruppen, Kommunikationsprozessoren (abhängig von der verwendeten Anschaltungsbaugruppe) oder leer

5.2.1.1 Besondere Regeln zur Platzhalterbaugruppe (DM 370 Dummy)

Die Platzhalterbaugruppe (DM 370 Dummy) ist eine Baugruppe, die Sie anstelle einer später einzusetzenden Baugruppe stecken können.

Die Baugruppe hält je nach Schalterstellung Adressraum für eine Baugruppe frei (z. B. für eine Digitalein/-ausgabebaugruppe) oder nicht (z. B. für eine Anschaltungsbaugruppe).

Schalterstellung auf DM 370 Dummy	Bedeutung	Bestellnummer
A	Adressraum reservierbar	6ES7370-0AA01-0AA0
NA	kein Adressraum reserviert	Keine (Baugruppe ist "unsichtbar"; sie wird nicht konfiguriert)

5.2.1.2 Besondere Regeln zur Digital-Simulationsbaugruppe SIM 374 IN/OUT 16

Mit der Digital-Simulationsbaugruppe SIM 374 IN/OUT 16 können Sie digitale Ein- und Ausgänge simulieren.

Diese Baugruppe finden Sie **nicht** im Fenster "Hardware Katalog!". Sie müssen statt der SIM 374 die zu simulierende Baugruppe in der Konfigurationstabelle platzieren!

Schalterstellung auf SIM 374 IN/OUT 16	Zu platzierende Baugruppe
16xOutput	6ES7322-1BH00-0AA0
8xOutput 8xInput	6ES7323-1BH00-0AA0
16xInput	6ES7321-1BH00-0AA0

5.2.2 Regeln für die Anordnung von Modulen (ET 200S und ET 200X)

5.2.2.1 Regeln für die Anordnung von Modulen bei ET 200S

Der Maximalausbau des Dezentralen Peripheriegerätes beträgt 64 Module (einschließlich IM 151/CPU). Die Module müssen lückenlos gesteckt werden.

Das Dezentrale Peripheriegerät ET 200S beginnt mit einer IM 151/CPU.

Nach dem Interfacemodul oder zu Beginn jeder Potentialgruppe folgt ein Powermodul.

Nach einem Powermodul folgen Digitale oder Analoge Module.

Das Dezentrale Peripheriegerät ET 200S endet mit dem Abschlussmodul, dass aber nicht konfiguriert wird.

Besonderheiten bei der Parametrierung von Vergleichsstellen

Beachten Sie folgende Reihenfolge:

1. Platzieren Sie das analoge Elektronikmodul und stellen Sie einen Kanal für die Vergleichsstellen-Funktion auf den Messbereich "RTD-4L Pt 100 Kl." ein.
2. Doppelklicken Sie auf die IM 151/CPU und spezifizieren Sie die Vergleichsstelle(n) im Abschnitt "Baugruppenparameter" des Parametrierdialogs: Steckplatz und Kanal des RTD-Moduls. Nicht vergessen, die Vergleichsstelle zu aktivieren.
3. Platzieren Sie das analoge Elektronikmodul für Temperaturmessung mittels Thermoelement (TC-Modul) und parametrieren Sie es mit der Vergleichsstellennummer (des RTD-Moduls).

5.2.2.2 Regeln für die Anordnung von Modulen bei ET 200X

Eine ET 200X Station besteht aus einem Basismodul (BM 147) und bis zu 7 Erweiterungsmodulen (EM). Die Module müssen lückenlos gesteckt sein.

Powermodule: Pro Basismodul können maximal 7 Erweiterungsmodule Powermodule PM 148 DO 4 x DC 24V/2A sein.

Verbraucherabzweige: Maximal 6 Verbraucherabzweige (EM 300...) sind steckbar. Sie können an beliebigen Plätzen im ET 200X gesteckt werden.

Pneumatic-Interface-Module: Maximal 1 Pneumatic-Interface-Modul (EM 148-P DO 16 P/CPV...) als letztes Erweiterungsmodul im ET 200X-Aufbau anschließbar.

Der Maximalausbau ist abhängig von der Stromaufnahme der einzelnen Module. Im Handbuch zum Dezentralen Peripheriesystem ET 200X sind alle Konfigurationsmöglichkeiten mit ihren Begrenzungen aufgeführt. Außerdem finden Sie dort Möglichkeiten, die Begrenzung zu erhöhen.

5.2.3 Schritte zum Konfigurieren von Baugruppen

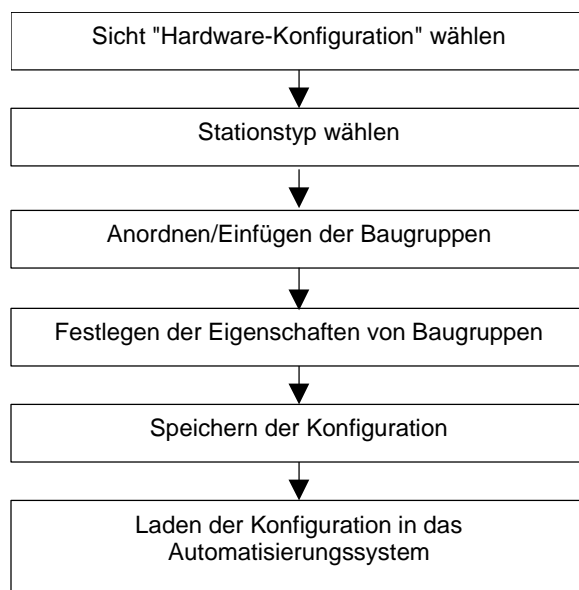
5.2.3.1 Überblick: Vorgehensweise beim Konfigurieren und Parametrieren einer Station

Voraussetzung

Sie haben ein Projekt geöffnet oder neu angelegt.

Prinzipielle Vorgehensweise

Um einen Aufbau zu konfigurieren und zu parametrieren gehen Sie folgendermaßen vor:



5.2.3.2 Auswählen eines Stationstyps

Wenn Sie die Sicht "HW-Konfigurieren" gewählt haben, dann ist beim ersten Öffnen der Hardware-Konfiguration einer Station die Auswahl des Stationstyps erforderlich.

Dazu müssen Sie die Auswahl im obersten Register des Katalogs treffen - das Register ist bereits geöffnet und der Stationstyp S7-300 vorgewählt.

Wenn Sie eine andere Option wählen, stellt sich automatisch der Arbeitsbereich auf den entsprechenden Stationstyp ein; z. B. ändert sich die maximal steckbare Anzahl von Baugruppen.


5.2.3.3 Anordnen von Baugruppen im Baugruppenträger

Voraussetzung

Die Hardwarekonfiguration ist geöffnet und das Fenster ist so angeordnet, dass Baugruppenträger (grafische oder tabellarische Sicht) und Hardware-Katalog sichtbar sind.

Vorgehensweise

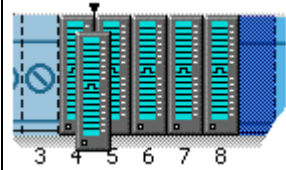
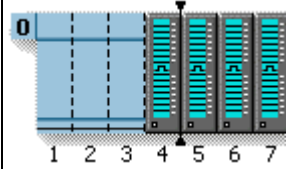
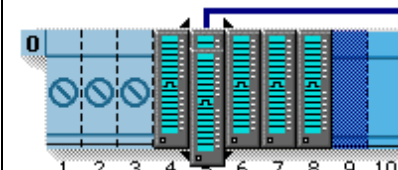
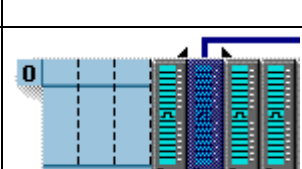
1. Wählen Sie eine Baugruppe (z. B. eine CPU) aus dem Hardware Katalog.
2. Ziehen Sie die Baugruppe per Drag&Drop in die entsprechende Zeile des Baugruppenträgers (Konfigurationstabelle). STEP 7 Lite prüft, OB Steckplatzregeln verletzt werden (eine S7-300-CPU darf z. B. nur auf Steckplatz 2 gesteckt werden).

 Symbol für Verletzung von Steckplatzregeln

3. Wiederholen Sie die Schritte 1 und 2, bis der Baugruppenträger mit den gewünschten Baugruppen vollständig bestückt ist.

Alternativ können Sie auch die entsprechende Zeile oder mehrere Zeilen in der Konfigurationstabelle markieren und im Fenster "Hardware Katalog" auf die gewünschte Baugruppe doppelklicken. Wenn mehrere Zeilen markiert sind, werden alle markierten Zeilen mit der Baugruppe auf einmal bestückt.

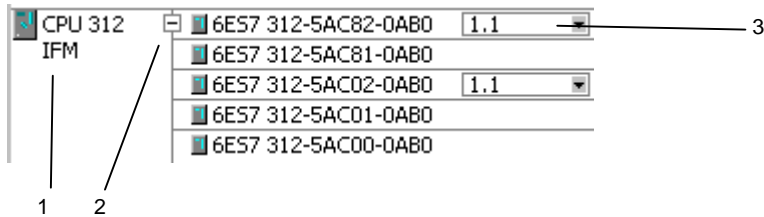
Besonderheiten in der grafischen Sicht

Anzeige in der grafischen Sicht	Bedeutung
	Sie haben eine Baugruppe per Drag&Drop zwischen zwei belegte Steckplätze positioniert. Wenn Sie jetzt die Baugruppe "fallen lassen", dann wird sie auf Steckplatz 5 gesteckt und die Baugruppen rechts daneben werden automatisch um einen Steckplatz nach rechts geschoben.
	Sie haben die Lücke zwischen zwei Baugruppen markiert. Wenn Sie jetzt im Katalog auf eine steckbare Baugruppe doppelklicken, dann wird diese Baugruppe auf Steckplatz 5 gesteckt und die Baugruppen rechts daneben werden automatisch um einen Steckplatz nach rechts geschoben.
	Sie haben eine Baugruppe per Drag&Drop auf einen belegten Steckplatz positioniert. Wenn Sie jetzt die Baugruppe "fallen lassen", dann wird sie auf Steckplatz 5 gesteckt und die "darunter" liegende Baugruppe wird gelöscht ("Baugruppentausch"). Die Parametrierung (z. B. bei einer CPU) wird von der "neuen" Baugruppe übernommen.
	Sie haben einen belegten Steckplatz markiert. Wenn Sie jetzt im Katalog auf eine steckbare Baugruppe doppelklicken, dann wird diese Baugruppe auf Steckplatz 5 gesteckt und die ursprünglich gesteckte Baugruppe wird gelöscht ("Baugruppentausch").

5.2.3.4 Anzeige der CPU-Betriebssystem-Version in der Baugruppenliste

Wenn von einer CPU unterschiedliche Betriebssystem-Versionen existieren, dann ist die Version in einer separaten Klappliste neben der CPU mit ihrer Bestell-Nr. auszuwählen.

Kontrollieren Sie die Betriebssystem-Version der von Ihnen verwendeten CPU und wählen Sie genau diese aus.



1	CPU-Typ-Bezeichnung
2	Symbol zum Öffnen/Schließen weiterer CPU-Bestellnummern gleichen CPU-Typs
3	Klappliste zum Auswählen der CPU-Betriebssystem-Version für eine Bestellnummer

5.2.3.5 Anordnen von C7-Komplettsystemen (Besonderheiten)

In einem C7-Komplettsystem (z. B. C7-620) sind in einem Gehäuse integriert:

- SIMATIC 300-CPU
- Ein- und Ausgänge (Digital und Analog)
- Anschaltungsbaugruppe IM 360 für die Kopplung weiterer SIMATIC 300-Baugruppen
- zeilenorientiertes OP mit Druckerschnittstelle

Bei den CPUs der Reihe C7-621 haben Sie die Möglichkeit, im Baugruppenträger 0 zusätzlich Signalbaugruppen der S7-300 Reihe zu stecken. Dazu muss die Erweiterungsbaugruppe 6ES7 621-1AD00-6AE3 auf Steckplatz 3 im Baugruppenträger 0 gesteckt werden.

Vorgehensweise

1. Wählen Sie ein C7-Komplettsystem aus dem Katalog. Diese Systeme sind im Register "C7" (System S7-300) zu finden.
2. Ziehen Sie das C7-Komplettsystem per Drag&Drop in das Stationsfenster. Die integrierten Ein- und Ausgänge "verteilen" sich automatisch auf die Steckplätze neben der CPU.
3. Falls Sie das C7-Komplettsystem erweitern wollen:
Ordnen Sie dem Baugruppenträger Baugruppen zu. Wichtig: Die Anschaltungsbaugruppen müssen in allen Baugruppenträgern gesteckt sein, damit eine Kopplung möglich ist!

5.2.3.6 Festlegen der Eigenschaften von Baugruppen/Schnittstellen

Einführung

Eigenschaften von Komponenten wie z. B. Baugruppen oder Schnittstellen sind im folgenden Adressen und Parameter. Nur wenn Sie die voreingestellten Werte ändern möchten, lesen Sie die folgenden Abschnitte.

Voraussetzung

Sie haben die Komponente, deren Eigenschaften Sie ändern wollen, in der Konfigurationstabelle angeordnet.

Vorgehensweise

Jede Komponente (Baugruppe, Schnittstelle) hat voreingestellte Eigenschaften, z. B. voreingestellte Messarten und Messbereiche bei Analogeingabebaugruppen.

Wenn Sie diese Einstellungen ändern wollen, dann gehen Sie folgendermaßen vor:

1. Doppelklicken Sie in der Konfigurationstabelle auf die zu parametrierende Komponente (z. B. Baugruppe) oder markieren Sie die Zeile, und wählen Sie den Menübefehl **Bearbeiten > Baugruppenparameter**.

Alternativ:

Mit rechter Maustaste: Bewegen Sie den Mauszeiger auf die Komponente, drücken die rechte Maustaste und wählen aus dem Kontextmenü den Befehl **Baugruppenparameter**.

Mit Schaltfläche "Baugruppenparameter": Bewegen Sie den Mauszeiger auf die Komponente und klicken Sie auf die Schaltfläche "Baugruppenparameter".

2. Legen Sie die Eigenschaften der Komponente mit Hilfe der angezeigten Registerdialoge fest.
Die Liste im linken Bereich des Dialogfeldes erleichtert das Navigieren zu den einzelnen Parametern.

Besonderheiten bei CPUs mit integrierten Ein- und Ausgängen

CPUs mit integrierten Ein- und Ausgängen wie z. B. CPU 31x C ("Kompakt-CPUs") haben in der Spalte "E-Adresse" eine Schaltfläche "Details". Wenn Sie auf diese Schaltfläche klicken, werden zusätzliche Zeilen mit den Adressen der integrierten Ein- und Ausgänge angezeigt. Ein Doppelklick auf eine dieser Zeilen öffnet ebenso den Parametrierdialog wie ein Doppelklick auf die Zeile, in der die CPU steckt.

5.2.3.7 Adressen zuweisen

Wir unterscheiden bei der Zuweisung von Adressen zwischen Teilnehmeradressen und Ein-/Ausgangsadressen (Peripherieadressen).

Teilnehmeradressen

Teilnehmeradressen sind Adressen der Schnittstellen von Baugruppen (MPI- und PROFIBUS-Adressen). Sie werden benötigt, um die verschiedenen Teilnehmer eines Subnetzes adressieren zu können, z. B. eine IM 151/CPU (ET 200S) an einem PROFIBUS-Subnetz. Die Adressen können Sie beim Hardware konfigurieren vergeben (Schnittstellen-Parameter). Die Teilnehmeradresse der CPU, mit der das PG verbunden ist, wird in der Titelzeile des CPU-Bedienpanels in eckigen Klammern angezeigt.

Die Teilnehmeradresse bleibt auch beim Urlöschen der CPU erhalten.

Hinweis zur Einstellung von PROFIBUS-Adressen der CPUs von ET 200S und ET 200X:

Die bei der Parametrierung eingegebene Adresse muss mit der Adresse an den DIP-Schaltern der CPU übereinstimmen, ansonsten läuft die CPU nicht an!

Ein-/Ausgangsadressen

Ein-/Ausgangsadressen (Peripherieadressen) werden dazu benötigt, um im Anwenderprogramm Eingänge zu lesen bzw. Ausgänge zu setzen.

5.2.3.8 Ein-/Ausgangsadressen zuweisen

Ein- und Ausgangsadressen vergibt STEP 7 Lite bereits beim Platzieren von Baugruppen in der Konfigurationstabelle. Damit hat jede Baugruppe ihre Anfangsadresse (Adresse des ersten Kanals); die Adressen der übrigen Kanäle ergeben sich aus dieser Anfangsadresse. Diese Adressen können nicht geändert werden; jeder Steckplatz hat eine fest zugeordnete Anfangsadresse:

Steckplatz 4: Adresse 0 (Digitalbaugruppe) oder Adresse 256 (Analogbaugruppe)

Steckplatz 5: Adresse 4 (Digitalbaugruppe) oder Adresse 272 (Analogbaugruppe)

Etc...

5.2.3.9 Tipps zum Bearbeiten von Stationskonfigurationen

Verschieben von Baugruppen

Baugruppen oder andere Komponenten können Sie einfach per Drag&Drop auf einen anderen zulässigen Steckplatz innerhalb der Station verschieben.

Baugruppen tauschen

Wenn Sie bereits eine Konfiguration erstellt haben und eine Baugruppe durch eine andere aus dem Hardware Katalog austauschen wollen, dann gehen Sie folgendermaßen vor:

1. Ziehen Sie die neue Baugruppe (z. B. CPU) per Drag&Drop auf den Steckplatz der zu ersetzenden Baugruppe.
 - Wenn die Baugruppe durch eine "kompatible" Baugruppe ersetzt werden soll und bereits parametrierung ist (z. B. CPU oder Analogbaugruppe), dann wird die Parametrierung übernommen.
 - Wenn die neue Baugruppe nicht alle Parameter bzw. Einstellungen übernehmen kann, werden Sie durch eine Meldung darauf aufmerksam gemacht und Sie können den Vorgang ggf. abbrechen.
 - Wenn die neue Baugruppe komplett anders ist als die zu ersetzende Baugruppe, dann wird nach Rückfrage die zu ersetzende Baugruppe gelöscht und die neue Baugruppe stattdessen eingefügt.
2. Bestätigen Sie ggf. im Folgedialog das Ersetzen der Baugruppe.

Die Parametrierung wird nur bei "kompatiblen" Baugruppen übernommen. Wenn die Baugruppen nicht kompatibel sind, wird beim Tausch die "alte" Baugruppe gelöscht und die neue Baugruppe gesteckt. Sie müssen in dem Fall die Baugruppe neu parametrieren.

Beispiel: Eine parametrierte CPU können Sie durch eine CPU mit neuer Bestellnummer tauschen - die komplette Parametrierung (z. B. MPI-Adresse) wird von der neuen Baugruppe übernommen.

Welche Baugruppen zu einer gesteckten und markierten Baugruppe kompatibel sind, wird im Register "Kompatibel" des Hardware Katalogs angezeigt.

Tipp

In jedem Fall können Sie über den Menübefehl **Bearbeiten > Rückgängig** den Tausch rückgängig machen.

Markieren mehrerer Zeilen in der Konfigurationstabelle

Wenn Sie mehrere Zeilen in einer Konfigurationstabelle markieren möchten, um mehrere Baugruppen zu kopieren oder zu löschen, dann gehen Sie folgendermaßen vor:

Alle Zeilen markieren:	Wählen Sie den Menübefehl Bearbeiten > Alles markieren
Zusammenhängenden Bereich markieren:	Klicken Sie auf die erste Zeile des zu markierenden Bereichs Halten Sie die Umschalttaste gedrückt und klicken Sie auf die letzte Zeile des zu markierenden Bereichs
Mehrere Zeilen markieren:	Drücken Sie die Taste STRG, halten sie gedrückt, und klicken Sie dann auf alle Zeilen, die Sie auswählen möchten

5.2.4 Wissenswertes zu ET 200S-Motorstartern (High Feature)

5.2.4.1 Anlagenzustände erkennen durch Stromwerte

Mit Hilfe des Motorstromes und der Stromgrenzwerte kann auf verschiedene Anlagenzustände geschlossen werden:

Anlagenzustand	Stromwert	Schutz durch:
Anlage wird schwergängiger, z.B. wegen Lagerschaden	Strom ist größer als normal	Stromgrenzwerte
Anlage wird leichtgängiger, z.B. weil das Verarbeitungsmaterial der Anlage ausgegangen ist	Strom ist kleiner als normal	Stromgrenzwerte
Anlage ist blockiert	sehr hoher Strom fließt	Blockierschutz (durch Blockierstrom und Blockierzeit)
Motor läuft im Leerlauf, z.B. wegen Anlagenschaden oder Motor noch nicht angeschlossen (Inbetriebnahme)	Sehr kleiner Strom fließt (< 18,75 % vom Bemessungsbetriebsstrom I_e)	Nullstromerkennung

5.2.4.2 Blockierstrom

Wenn der Blockierstrom überschritten wird, erkennt der Motorstarter eine Blockierung. Ab diesem Zeitpunkt wird die Überwachung der Blockierzeit gestartet. Deren Dauer wird durch die Blockierzeit - unabhängig von der Abschaltklasse - bestimmt.

Hinweis

Wenn nach Ablauf der Blockierzeit immer noch eine Blockierung vorhanden ist, schaltet der Motorstarter ab.

Einstellbereich: 800% vom Bemessungsbetriebsstrom fest eingestellt.

5.2.4.3 Blockierzeit

Zeit, in der eine Blockierung ohne Abschalten vorhanden sein darf. Nach Ablauf der Blockierzeit und immer noch vorhandener Blockierung schaltet der Motorstarter ab.

Einstellbereich: 1 Sekunde fest eingestellt.

Hinweis

Der Blockierschutz ist sofort nach dem Einschalten aktiv!

5.2.4.4 Verhalten bei Nullstromerkennung

Die Nullstromerkennung spricht an, wenn der Motorstrom in allen 3 Phasen niedriger wird als 18,75 % vom Bemessungsbetriebsstrom.

Mit diesem Geräteparameter bestimmen Sie, wie sich der Motorstarter bei Nullstromerkennung verhalten soll:

- Warnen (Verhalten wie bei "Sammelwarnung")
- Abschalten (Verhalten wie bei "Abschaltung ohne Wiederanlauf")

Hinweis

Beim Einschalten des Motors wird die Nullstromerkennung für ca. 1 Sekunde unterdrückt !

5.2.4.5 Unsymmetrie

Drehstrom-Asynchronmotoren reagieren auf geringe Unsymmetrien der Netzspannung mit einer höheren unsymmetrischen Stromaufnahme. Dadurch erhöht sich die Temperatur in der Ständer- und Läuferwicklung.

Hinweis

Beim Einschalten des Motors wird die Unsymmetrie-Auswertung für ca. 500 Millisekunden unterdrückt.

5.2.4.6 Thermisches Motormodell

Aus den gemessenen Motorströmen und den Geräteparametern "Bemessungsbetriebsstrom" und "Abschaltklasse" wird nach dem thermischen Motormodell die Wicklungstemperatur des Motors berechnet. Daraus wird abgeleitet, OB der Motor überlastet ist oder im normalen Betriebsbereich arbeitet.

5.2.4.7 Wiederbereitschaftszeit

Die Wiederbereitschaftszeit ist die Zeitvorgabe für das Abkühlverhalten, nach der bei Überlastauslösung die Auslösung quitiert werden kann. Spannungsausfälle während dieser Zeit verlängern die Zeitvorgabe entsprechend. Die Wiederbereitschaftszeit nach Überlastauslösung beträgt ca. 90 Sekunden.

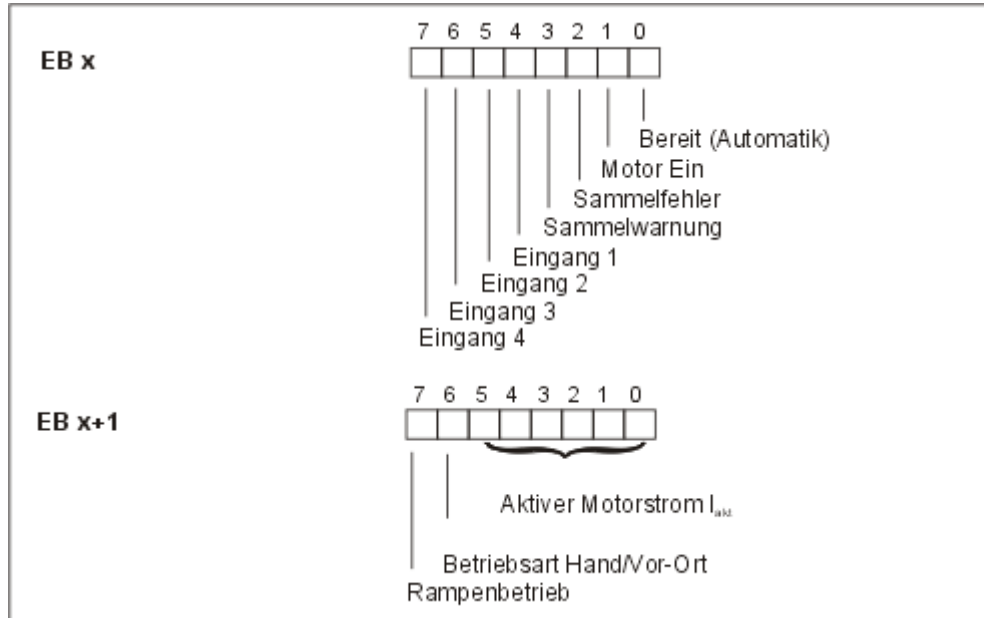
5.2.4.8 Übersicht: Mögliche Aktionen des Motorstarters

Aktion	Erläuterung
Keine Aktion	Die "Eingang n – Pegel" werden im Prozessabbild 1-aktiv angezeigt. Die "Eingang n – Pegel" führen zu keiner Reaktion im Motorstarter.
Abschaltung ohne Wiederanlauf	Führt zur Abschaltung von Motor und Bremsausgang. Im Prozessabbild wird das Bit "Sammelfehler" gesetzt. Quittierung nach Beseitigung der Abschaltursache erforderlich (über Prozessabbild oder durch Drehschalter am Gerät).
Abschaltung mit Wiederanlauf (Autoreset) Hinweis: Wiederanlauf bedeutet, dass bei anstehendem Einschalt-Kommando der Motorstarter selbstständig wieder einschaltet, wenn die Fehlerursache beseitigt wurde (Autoreset).	Führt zur Abschaltung von Motor und Bremsausgang. Im Prozessabbild wird das Bit "Sammelfehler" gesetzt. Automatische Quittierung nach Beseitigung der Abschaltursache.
Abschaltung Endlage-Rechtslauf / Abschaltung Endlage-Linkslauf	Unabhängig von der Drehrichtung werden Motor und Bremsausgang abgeschaltet. Im Prozessabbild wird das Bit "Sammelfehler" gesetzt. Erneutes Einschalten des Bremsausganges (DO 0.2) ist möglich, nachdem Motor-RECHTS (DO 0.0) und Motor-LINKS (DO 0.1) und Bremsausgang (DO 0.2) auf 0 gesetzt wurden. Abschaltung Endlage-Rechtslauf: Erneutes Einschalten des Motors nur mit Gegenbefehl "Motor-LINKS" (DO 0.1) möglich. Abschaltung Endlage-Linkslauf: Erneutes Einschalten des Motors nur mit Gegenbefehl "Motor-RECHTS" (DO 0.0) möglich.
Sammelwarnung (Warnen)	Im Prozessabbild wird das Bit "Sammelwarnung" gesetzt. Der Motorstarter und der Bremsausgang werden nicht abgeschaltet
Betriebsart Hand-Vor-Ort	Steuerung nur noch über "Eingang n - Aktion: Motor-RECHTS und Motor-LINKS" (s. unten) möglich! Steuerung über Feldbus (Betriebsart Automatik) nicht möglich! Betriebsart Automatik ist erst wieder möglich, wenn Betriebsart Hand-Vor-Ort zurückgenommen wird und nicht "Eingang n - Aktion: Motor-RECHTS oder Motor-LINKS" aktiv ist.
Notstart	Schaltet bei anstehenden EIN-Schaltbefehl den Motor trotz anstehender Abschaltursache ein. Schaltet bei anstehenden EIN-Schaltbefehl für den Bremsausgang auch diesen ein. Nur als Schließer zulässig
Motor-RECHTS / Motor-LINKS	Für diese Aktionen muss sich der Motorstarter in "Betriebsart Hand-Vor-Ort" befinden. Motor-RECHTS: Motor und Bremsausgang ein- und ausschalten (Rechtslauf). Motor-LINKS: Motor und Bremsausgang ein- und ausschalten (Linkslauf). Nur als Schließer zulässig

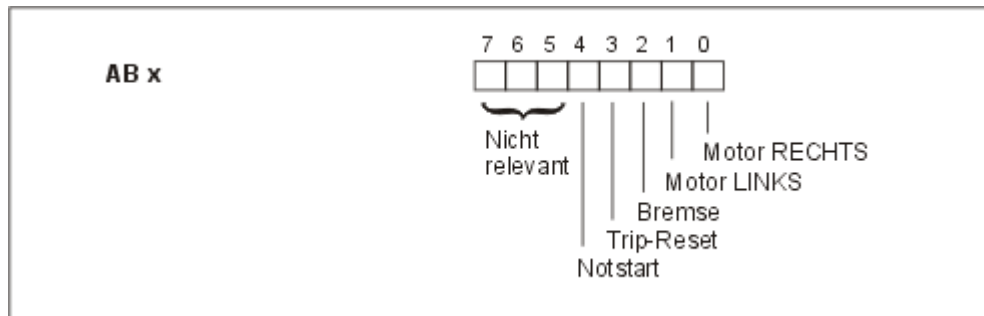
5.2.4.9 Belegung des Motorstarters im Prozessabbild

Im folgenden ist die Belegung des Motorstarters im Prozeßabbild dargestellt, wobei der Motorstarter die Anfangsadresse x hat.

Prozeßabbild der Eingänge



Prozeßabbild der Ausgänge



5.3 Speichern einer Konfiguration und Konsistenzprüfung

Um eine Konfiguration mit allen eingestellten Parametern und Adressen zu speichern, wählen Sie den Menübefehl **Datei > Speichern**. Auch inkonsistente (fehlerhafte) Konfigurationen können so gespeichert werden.

Vor dem Laden sollten Sie mit dem Menübefehl **Extras > Konsistenz prüfen** die Konfiguration prüfen und die gemeldeten Fehler beseitigen.

6 Programmieren von Bausteinen

6.1 Festlegen von Symbolen

6.1.1 Absolute und symbolische Adressierung

In einem STEP 7 Lite-Programm arbeiten Sie mit Operanden wie E/A-Signalen, Merkern, Zählern, Zeiten, Datenbausteinen und Funktionsbausteinen. Sie können diese Operanden in Ihrem Programm absolut adressieren, die Lesbarkeit von Programmen steigt jedoch erheblich, wenn Sie dafür Symbole (z. B. Motor_A_Ein oder Bezeichnungen gemäß eines in Ihrer Branche üblichen Kennzeichnungssystems) benutzen. In Ihrem Anwenderprogramm lässt sich dann ein Operand über dieses Symbol ansprechen.

Absolute Adresse

Eine absolute Adresse besteht aus einem Operandenkennzeichen und einer Adresse (z. B. A 4.0, E 1.1, M 2.0, FB 21).

Symbolische Adressierung

Sie können Ihr Programm übersichtlicher gestalten und die Fehlerbehebung vereinfachen, wenn Sie die absoluten Adressen mit symbolischen Namen versehen.

STEP 7 Lite kann automatisch die symbolischen Namen in die erforderlichen absoluten Adressen übersetzen. Wenn Sie auf ARRAYS, STRUCTs, Datenbausteine, Lokaldaten, Codebausteine und anwenderdefinierte Datentypen vorzugsweise über symbolische Namen zugreifen, dann müssen Sie den absoluten Adressen jedoch zunächst symbolische Namen zuordnen, bevor Sie die Daten symbolisch adressieren können.

Sie können beispielsweise dem Operanden A 4.0 den symbolischen Namen MOTOR_EIN zuordnen und dann MOTOR_EIN als Adresse in einer Programmanweisung verwenden. Über symbolische Adressen können Sie leichter erkennen, inwieweit Elemente des Programms mit den Komponenten Ihres Prozesssteuerungsprojekts übereinstimmen.

Hinweis

In einem symbolischen Namen (Variablenbezeichnung) sind zwei Unterstriche nacheinander (wie z. B.: MOTOR__EIN) nicht erlaubt.

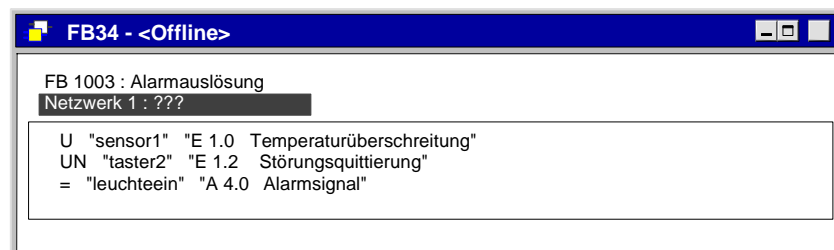
Unterstützung bei der Programmeingabe

In den Programmiersprachen KOP, FUP und AWL können Sie Adressen, Parameter und Bausteinennamen absolut oder als Symbol angeben.

Mit dem Menübefehl **Ansicht > Anzeigen mit > Symbolischer Darstellung** schalten Sie zwischen der Darstellung absoluter und symbolischer Adressierung um.

Zur leichteren Programmierung mit symbolischer Adressierung können Sie sich zu den vorkommenden Symbolen die zugehörige absolute Adresse mit dem Symbolkommentar anzeigen lassen. Dieses erreichen Sie über den Menübefehl **Ansicht > Anzeigen mit > Symbolinformation**. Dadurch wird hinter jeder AWL-Anweisung der Zeilenkommentar entsprechend ersetzt. Die Anzeige können Sie nicht bearbeiten; Änderungen müssen Sie in der Symboltabelle bzw. in der Variablendeklarationstabelle vornehmen.

Nachfolgendes Bild zeigt Ihnen eine Symbolinformation in AWL.



Beim Ausdruck des Bausteins wird die aktuelle Bildschirmdarstellung mit Anweisungskommentar oder Symbolkommentar wiedergegeben.

6.1.2 Globale und lokale Symbole

Ein Symbol ermöglicht Ihnen, anstelle von absoluten Adressen mit einer aussagekräftigen Bezeichnung zu arbeiten. Mit der Kombination aus kurzen Symbolen und ausführlicheren Kommentaren können Sie sowohl die Anforderung einer effektiven Programmerstellung als auch einer guten Programmdokumentation erfüllen.

Man unterscheidet zwischen lokalen und globalen Symbolen.

	Globale Symbole	Lokale Symbole
Gültigkeitsbereich	<ul style="list-style-type: none"> • Gelten im gesamten Anwenderprogramm • Können von allen Bausteinen benutzt werden • Haben in allen Bausteinen die gleiche Bedeutung • Die Bezeichnung muss im gesamten Anwenderprogramm eindeutig sein 	<ul style="list-style-type: none"> • Sind nur in dem Baustein bekannt, in dem sie definiert wurden • Die gleiche Bezeichnung kann in verschiedenen Bausteinen für unterschiedliche Zwecke verwendet werden
Zulässige Zeichen	<ul style="list-style-type: none"> • Buchstaben, Ziffern, Sonderzeichen • Umlaute außer 0x00, 0xFF und Anführungszeichen • Symbol muss bei Verwendung von Sonderzeichen in Anführungs- und Schlusszeichen gesetzt werden 	<ul style="list-style-type: none"> • Buchstaben • Ziffern • Unterstrich (<u> </u>)
Einsatz	<p>Sie können globale Symbole definieren für:</p> <ul style="list-style-type: none"> • E/A-Signale (E, EB, EW, ED, A, AB, AW, AD) • Peripherie-Eingänge / -Ausgänge (PE, PA) • Merker (M, MB, MW, MD) • Zeiten (T)/ Zähler (Z) • Codebausteine (OB, FB, FC, SFB, SFC) • Datenbausteine (DB) • Anwenderdefinierte Datentypen 	<p>Lokale Symbole können Sie definieren für:</p> <ul style="list-style-type: none"> • Bausteinparameter (Eingangs-, Ausgangs- und Durchgangparameter) • statische Daten eines Bausteins • temporäre Daten eines Bausteins
Ort der Definition	Symboltabelle	Variablendeklarationstabelle des Bausteins

6.1.3 Darstellung von globalen oder lokalen Symbolen

Sie können die globalen und lokalen Symbole im Anweisungsteil eines Programms folgendermaßen auseinanderhalten:

- Symbole aus der Symboltabelle (global) werden in Anführungszeichen ".." dargestellt.
- Symbole aus der Variablendeklarationstabelle des Bausteins (bausteinlokal) werden mit vorangestelltem "#" dargestellt.

Die Kennzeichnung in Anführungszeichen oder mit "#" müssen Sie nicht eingeben. Bei der Programmeingabe in KOP, FUP oder AWL wird die Kennzeichnung nach der Syntaxprüfung entsprechend ergänzt.

Wenn jedoch Verwechslungen denkbar sind, weil z. B. gleiche Symbole in Symboltabelle und Variablendeklarationstabelle verwendet wurden, müssen Sie, wenn Sie das globale Symbol verwenden wollen, dieses explizit kennzeichnen. Nicht gekennzeichnete Symbole werden in diesem Fall als bausteinlokale Variablen interpretiert.

Die Kennzeichnung globaler Symbole ist zudem erforderlich, wenn das Symbol Leerzeichen oder Sonderzeichen enthält.

Hinweis

Mit dem Menübefehl **Ansicht > Anzeigen mit > Symbolischer Darstellung** schalten Sie zwischen der Anzeige der vereinbarten globalen Symbole und der zugehörigen absoluten Adresse um.

6.1.4 Einstellen des Operandenvorrangs (absolut/symbolisch)

Über den Menübefehl **Extras > Einstellungen** (Abschnitt "Allgemein") können Sie den Operandenvorrang "Absolut" bzw. "Symbol" wählen.

Werden Zuordnungen in der Symboltabelle nachträglich geändert, so kann über diese Einstellung festgelegt werden, OB im Anwenderprogramm entweder der absolute Operand oder das Symbol geändert wird. Der eingestellte Operandenvorrang wirkt sich erst dann aus, wenn der Codebaustein geöffnet und anschließend gespeichert wird.

Bei der Einstellung "Operandenvorrang **Absolut**" bleibt nach einer Zuordnungsänderung in der Symboltabelle der absolute Operand im Anwenderprogramm erhalten (das Symbol wird geändert); bei "Operandenvorrang **Symbol**" bleibt der symbolische Operand erhalten (der absolute Operand wird geändert). Bei Bausteinaufrufen, wie CALL, CC oder UC, ist immer nur die absolute Bausteinnummer maßgeblich (d. h. es wird immer nur das Symbol geändert).

Beispiel:

Im folgenden Beispiel wird dargestellt, wie sich der Operandenvorrang bei einer Änderung der Symboltabelle im Anwenderprogramm auswirkt.

Anweisungen vor der Änderung	U "Symbol_A" O "Symbol_B"		(Symbol_A = E0.1) (Symbol_B = E0.2)
Zuordnungsänderung in der Symboltabelle	Symbol_A = E0.1 Symbol_B = E0.2	--> -->	Symbol_A = E0.2 Symbol_B = E0.1
Baustein wird geöffnet mit "Operandenvorrang Absolut"	U "Symbol_B" O "Symbol_A"		(E0.1) (E0.2)
Baustein wird geöffnet mit "Operandenvorrang Symbol"	U "Symbol_A" O "Symbol_B"		(E0.2) (E0.1)

6.1.5 Symboltabelle für globale Symbole

In der Symboltabelle definieren Sie globale Symbole.

6.1.5.1 Struktur und Bestandteile der Symboltabelle

Struktur der Symboltabelle

Status	Symbol	Adresse	Datentyp	Kommentar
	Automatik Ein	E 0.5	BOOL	für die Speicherfunktion (einschalten)
	Automatikbetrieb	A 4.2	BOOL	Speicherfunktion

Status

Die Spalte zeigt an, OB einem Symbol spezielle Objekteigenschaften zugewiesen wurden:

- **?** Markiert ein unbenutztes Symbol. Die Symbolzelle wird blau hinterlegt.
- **=** Markiert ein Symbol, zu dem es ein Duplikat (absolut/symbolisch) gibt. Die entsprechenden Zellen werden Pastellrot hinterlegt.
- **X** Markiert syntaktisch fehlerhafte Einträge. Die entsprechende Zelle wird rot hinterlegt.

Symbol

Der Name des Symbols darf maximal 24 Zeichen lang sein.

Für Operanden aus Datenbausteinen (DBD, DBW, DBB, DBX) können Sie in der Symboltabelle keine Namen vergeben. Deren Namen sind durch die Deklaration in den Datenbausteinen festgelegt.

Für Organisationsbausteine (OBs) und einige Systemfunktionsbausteine (SFBs) und Systemfunktionen (SFCs) existieren vordefinierte Symboltabelleneinträge, die Sie beim Bearbeiten der Symboltabelle Ihres Programms in diese importieren.

Adresse

Eine Adresse ist die Kennzeichnung für einen bestimmten Operanden.
Beispiel: Eingang E 12.1

Die Syntax der Adresse wird bei der Eingabe überprüft. Außerdem wird geprüft, OB die Adresse dem angegebenen Datentyp zugeordnet werden darf.

Datentyp

Sie können unter den verschiedenen Datentypen wählen, die Ihnen STEP 7 Lite zur Verfügung stellt. Das Feld wird nach Eingabe der Adresse mit einem gültigen Default-Datentyp vorbelegt. Wenn es mehrere Datentypen für diese Adresse gibt, werden die anderen gültigen Datentypen in einer Auswahlliste zur Verfügung gestellt.

Kommentar

Allen Symbolen können Sie Kommentare zuordnen. Mit einer Kombination aus kurzen Symbolen und ausführlichen Kommentaren können Sie sowohl Anforderungen an eine gute Programmdokumentation als auch an eine effektive Programmerstellung erfüllen. Ein Kommentar kann maximal 80 Zeichen lang sein.

6.1.5.2 Zulässige Adressen und Datentypen in der Symboltabelle

Es ist nur eine Notation über die gesamte Symboltabelle möglich. Die Umschaltung zwischen der deutschen und englischen Mnemonik muss unter **Extras > Einstellungen...** erfolgen.

Englische Mnemonik	Deutsche Mnemonik	Erläuterung:	Datentyp:	Adressbereich:
I	E	Eingangsbit	BOOL	0.0..65535.7
IB	EB	Eingangsbyte	BYTE, CHAR	0..65535
IW	EW	Eingangswort	WORD, INT, S5TIME, DATE	0..65534
ID	ED	Eingangsdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
Q	A	Ausgangsbit	BOOL	0.0..65535.7
QB	AB	Ausgangsbyte	BYTE, CHAR	0..65535
QW	AW	Ausgangswort	WORD, INT, S5TIME, DATE	0..65534
QD	AD	Ausgangsdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
M	M	Merkerbit	BOOL	0.0..65535.7
MB	MB	Merkerbyte	BYTE, CHAR	0..65535
MW	MW	Merkerwort	WORD, INT, S5TIME, DATE	0..65534
MD	MD	Merkerdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
PIB	PEB	Peripherieeingangsbyte	BYTE, CHAR	0..65535
PID	PED	Peripherieeingangsdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
PIW	PEW	Peripherieeingangswort	WORD, INT, S5TIME, DATE	0..65534
PQB	PAB	Peripherieausgangsbyte	BYTE, CHAR	0..65535
PQD	PAD	Peripherieausgangsdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
PQW	PAW	Peripherieausgangswort	WORD, INT, S5TIME, DATE	0..65534
T	T	Zeit	TIMER	0..65535
C	Z	Zähler	COUNTER	0..65535
FB	FB	Funktionsbaustein	FB	0..65535
OB	OB	Organisationsbaustein	OB	1..65535
DB	DB	Datenbaustein	DB, FB, SFB, UDT	1..65535
FC	FC	Funktion	FC	0..65535
SFB	SFB	Systemfunktionsbaustein	SFB	0..65535
SFC	SFC	Systemfunktion	SFC	0..65535
UDT	UDT	Anwenderdefinierter Datentyp	UDT	0..65535

6.1.5.3 Unvollständige und mehrdeutige Symbole in der Symboltabelle

Unvollständige Symbole

Sie haben die Möglichkeit, auch unvollständige Symbole abzuspeichern. Dadurch können Sie beispielsweise zunächst nur den Namen festlegen und die Adressangaben zu einem späteren Zeitpunkt nachtragen. Insbesondere können Sie die Arbeiten an der Symboltabelle zu einem beliebigen Zeitpunkt unterbrechen und den Zwischenstand speichern. Um das Symbol bei der Software-Erstellung (ohne Fehlermeldung) benutzen zu können, müssen jedoch die Bezeichnung des Symbols, die Adresse und der Datentyp eingetragen sein.

Entstehung mehrdeutiger Symbole

Mehrdeutige Symbole entstehen, wenn Sie ein Symbol in die Symboltabelle einfügen, dessen Name (Symbol) und/oder Adresse bereits in einem anderen Symbol vorhanden sind. Dadurch werden sowohl das ursprünglich vorhandene und das eingefügte Symbol mehrdeutig.

Dieser Fall tritt z. B. ein, wenn Sie ein Symbol kopieren und einfügen, um den Eintrag in der Kopie danach leicht zu modifizieren.

Kennzeichnung mehrdeutiger Symbole

In der Symboltabelle sind mehrdeutige Symbole durch grafische Hervorhebung (Farbe, Schriftart) gekennzeichnet. Die veränderte Darstellung weist Sie auf eine erforderliche Nachbearbeitung hin. Sie können alle Symbole anzeigen oder die Anzeige so filtern, dass nur eindeutige oder nur mehrdeutige Symbole angezeigt werden.

Beseitigung der Mehrdeutigkeit

Ein mehrdeutiges Symbol wird eindeutig, wenn Sie die Komponente (Name und/oder Adresse) ändern, die die Mehrdeutigkeit verursacht hat. Sind zwei Symbole mehrdeutig, und Sie machen ein Symbol eindeutig, so wird das andere ebenfalls wieder eindeutig.

6.1.6 Eingabemöglichkeiten von globalen Symbolen

Für die Eingabe von Symbolen, die bei der späteren Programmierung weiterverwendet werden, gibt es mehrere Möglichkeiten:

- Eingabe direkt in die Symboltabelle
Sie können Symbole und ihre zugeordneten Adressen direkt in eine "Symboltabelle" eintragen. Dieses Verfahren empfiehlt sich zur Eingabe mehrerer Symbole und zur Ersterstellung der Symboltabelle, weil Sie die bereits definierten Symbole am Bildschirm angezeigt bekommen und so leichter die Übersicht behalten.
- Eingabe über ein Dialogfeld
Sie können im Bausteineditor ein Dialogfeld öffnen und darin ein neues Symbol definieren. Das Vorgehen eignet sich zur Definition einzelner Symbole, beispielsweise wenn Sie während der Programmierung feststellen, dass ein Symbol fehlt oder korrigiert werden muss. Sie ersparen sich so das Anzeigen der Symboltabelle.
- Importieren von Symboltabellen aus anderen Tabelleneditoren
Sie können die Daten für die Symboltabelle auch mit dem von ihnen bevorzugten Tabelleneditor erstellen und die erzeugte Datei dann in die Symboltabelle importieren.
- Eingabe der Bausteinsymbole über ein Dialogfeld
Sie können im Projektfenster entweder über den Menübefehl **Datei > Neu > Baustein...** oder über den Kontextmenübefehl **Neu > Baustein...** zum Dialogfeld "Neuer Baustein" gelangen. Das Kontextmenü blenden sie durch klicken der rechten Maustaste beim markierten Objekt ein. In diesem Dialogfeld können Sie das Bausteinsymbol definieren.
- Eingabe der Bausteinsymbole im Bausteineditor
Sie können im Bausteineditor die Sicht "Eigenschaften" wählen und den symbolischen Name des Bausteins dort editieren bzw. verändern. Sobald Sie im Feld "Symbol" bzw. "Symbolkommentar" eine Änderung vornehmen, wird diese sofort gespeichert. Auch wenn Sie die Eigenschaften-Sicht verlassen ohne zu speichern, werden die Änderungen in diesen Feldern trotzdem übernommen und in alle Sichten entsprechend angepasst.
- Eingabe der Bausteinsymbole im Projektfenster
Sie können im Projektfenster entweder mit zweimaligem Anklicken des Bausteins oder über den Kontextmenübefehl **Umbenennen** zur Namensänderung eines Bausteins gelangen. Das Kontextmenü blenden sie durch klicken der rechten Maustaste beim markierten Objekt ein.

6.1.6.1 Allgemeine Hinweise zur Eingabe von Symbolen

Um neue Symbole in die Symboltabelle einzutragen, navigieren Sie in die erste leere Zeile der Tabelle und füllen dann die Felder aus. Neue Zeilen können Sie mit dem Menübefehl **Einfügen > Zeile > Vor Markierung/Nach Markierung** in die Symboltabelle vor bzw. nach der aktuellen Zeile einfügen. Wenn die Zeile vor der Mauszeiger-Position bereits einen Operanden enthält, werden Sie beim Einfügen von neuen Symbolen unterstützt durch eine Vorbelegung der Spalten "Adresse" und "Datentyp". Die Adresse wird abgeleitet aus der vorhergehenden Zeile, als Datentyp wird der Default-Datentyp eingetragen.

Vorhandene Einträge können Sie mit Befehlen im Menü "Bearbeiten" kopieren und anschließend ändern. Auch nicht vollständig definierte Symbole können gespeichert werden.

Bei der Eingabe der Symbole in der Tabelle sind folgende Besonderheiten zu beachten:

Spalte	Hinweis
Symbol	Der Name muss innerhalb der gesamten Symboltabelle eindeutig sein. Wenn Sie das Feld verlassen, wird ein mehrdeutiges Symbol gekennzeichnet. Das Symbol darf maximal 24 Zeichen lang sein. Es sind keine Anführungszeichen " zugelassen.
Adresse	Wenn Sie das Feld verlassen, wird geprüft, OB die eingegebene Adresse zulässig ist.
Datentyp	Nach Eingabe der Adresse wird dieses Feld mit einer gültigen Default-Einstellung vorbelegt. Wenn es mehrere Datentypen für diese Adresse gibt, werden die anderen gültigen Datentypen in einer Auswahlliste zur Verfügung gestellt.
Kommentar	Hier können Sie Anmerkungen eingeben, um die Funktionen der Symbole kurz zu erklären (max. 80 Zeichen). Die Angabe des Kommentars ist optional.

6.1.6.2 Eingeben einzelner globaler Symbole im Dialog

Die hier beschriebene Vorgehensweise zeigt Ihnen, wie Sie beim Programmieren von Bausteinen Symbole über Dialogfelder ändern oder neu definieren können, ohne dass dabei die Symboltabelle angezeigt werden muss.

Die Vorgehensweise ist nützlich, wenn Sie nur ein einzelnes Symbol bearbeiten wollen. Um mehrere Symbole zu ändern, sollten Sie die Symboltabelle öffnen und direkt in der Symboltabelle arbeiten.

Aktivieren der Anzeige von Symbolen im Baustein

Aktivieren Sie bei geöffnetem Baustein die Anzeige der Symbole im Bausteineditor mit dem Menübefehl **Ansicht > Anzeigen mit > Symbolischer Darstellung**. Ein Häkchen vor dem Menübefehl kennzeichnet, dass die symbolische Darstellung aktiv ist.

Definieren von Symbolen bei der Programmeingabe

1. Markieren Sie im Anweisungsteil Ihres Programms die absolute Adresse, der Sie ein Symbol zuweisen wollen.
2. Wählen Sie entweder den Menübefehl **Bearbeiten > Symbole...** oder den Kontextmenübefehl **Symbole bearbeiten....** Das Kontextmenü blenden sie durch klicken der rechten Maustaste beim markierten Objekt ein.
3. Füllen Sie das angezeigte Dialogfeld aus. Tragen Sie insbesondere ein Symbol ein und schließen Sie den Dialog ab.

Das definierte Symbol wird in die Symboltabelle eingetragen.

Hinweise

Angaben, die zu mehrdeutigen Symbolen führen, werden im Dialog nach dem Verlassen des Eingabefeldes für das Symbol und in der Statusspalte der Symboltabelle mit dem Gleichheitssymbol **=** markiert. Die entsprechenden Zellen werden Pastellrot hinterlegt.

Wurde die Anweisung, deren Operanden zum Aufruf des Dialogs selektiert worden ist, noch nicht gespeichert, wird das Symbol im Dialog "Symbole bearbeiten" und in der Symboltabelle als nicht verwendetes Symbol mit einem "?" blau hinterlegt gekennzeichnet.

Ein Operand mit mehrdeutiger Symbolzuweisung wird in KOP, FUP und AWL absolut angezeigt. Für diesen Operanden können Sie den Dialog "Symbole bearbeiten" nicht aufrufen.

6.1.6.3 Eingeben mehrerer globaler Symbole in der Symboltabelle

Symboltabelle öffnen

Sie haben mehrere Möglichkeiten zum Öffnen einer Symboltabelle:

- Doppelklick auf die Symboltabelle im Projektfenster
- Markieren der Symboltabelle im Projektfenster und Einblenden des Kontextmenüs mit dem Menübefehl **Öffnen** durch Klicken der rechten Maustaste.

Die Symboltabelle zum aktuellen Programm wird in einem eigenen Fenster angezeigt. Nun können Sie Symbole erzeugen oder ändern. Beim ersten Öffnen der Symboltabelle ist die Symboltabelle noch leer.

Eingeben von Symbolen

Um neue Symbole in die Symboltabelle einzutragen, navigieren Sie in die erste leere Zeile der Tabelle und füllen dann die Felder aus. Neue Leerzeilen können Sie mit dem Menübefehl **Einfügen > Zeile vor Markierung/Zeile nach Markierung** in die Symboltabelle vor bzw. nach der aktuellen Zeile einfügen. Vorhandene Einträge können Sie mit Befehlen im Menü **Bearbeiten** kopieren und anschließend ändern. Speichern und schließen Sie dann die Symboltabelle. Sie können auch Symbole speichern, die noch nicht vollständig definiert sind.

Sortieren von Symbolen

Die Datensätze in der Symboltabelle lassen sich alphabetisch nach Symbol, Adresse, Datentyp oder Kommentar sortieren.

Klicken Sie auf die Spaltenüberschrift, so dass die Sortierung ausgelöst wird. Die Sortierung wird durch den blauen senkrechten Pfeil, der am rechten Rand erscheint angezeigt. Die Sortierrichtung wird durch die Pfeilrichtung dargestellt.

Die Symboltabelle wird nach den Einträgen in dieser Spalte sortiert. Durch Wiederholung der Aktion können Sie die Sortierreihenfolge umkehren.

Filtern von Symbolen

Mit einem Filter wählen Sie aus allen Datensätzen einer Symboltabelle eine Teilmenge aus.

Über die Schaltfläche "Filter bearbeiten.." öffnen Sie das Dialogfeld "Filter bearbeiten".

Hier können Sie Kriterien definieren, die die Datensätze erfüllen müssen, um herausgefiltert zu werden. Sie können filtern nach

- Namen, Adressen, Datentypen, Kommentaren
- Symbolen mit Status "gültig", "ungültig (mehrdeutig, unvollständig)"

Die einzelnen Kriterien sind mit UND verknüpft. Die herausgefilterten Datensätze beginnen mit den angegebenen Zeichenfolgen.

Sie erfahren unter Filtern der Symboltabelle mehr über die Möglichkeiten in dem Dialogfeld "Filter bearbeiten".

6.1.6.4 Exportieren und Importieren von Symboltabellen

Sie können die angezeigte Symboltabelle in eine Textdatei exportieren, um Sie beispielsweise mit einem beliebigen Texteditor weiterzubearbeiten.

Sie können Tabellen, die Sie mit einem anderen Werkzeug erzeugt haben, in ihre Symboltabelle importieren und hier weiterbearbeiten. Die Import-Funktion lässt sich beispielsweise benutzen, um unter STEP5/ST erzeugte Zuordnungslisten nach Konvertierung in die Symboltabelle aufzunehmen.

Zur Auswahl steht das Dateiformat *.SDF.

Sie können die gesamte Symboltabelle, eine durch Filter begrenzte Teilmenge oder in der Tabellendarstellung markierte Zeilen exportieren.

6.1.7 Schritte zum Bearbeiten der Symboltabelle

6.1.7.1

Öffnen einer Symboltabelle

Die "Symboltabelle" wird automatisch unter einem neuen Projekt angelegt. Um in einem Baustein Symbole für Globaldaten verwenden zu können, müssen diese in der Symboltabelle zugeordnet werden.

Öffnen Sie die "Symboltabelle" durch Doppelklick auf das Objekt im Projektfenster oder blenden sie durch klicken der rechten Maustaste ein Kontextmenü ein. Mit dem Menübefehl **Öffnen** können Sie dann die Symboltabelle öffnen.

6.1.7.2 Definieren einzelner Symbole

1. Aktivieren Sie im "Bausteineditor" die symbolische Ansicht mit dem Menübefehl **Ansicht > Anzeigen mit > Symbolischer Darstellung**. Ein Häkchen vor dem Menübefehl kennzeichnet, dass die symbolische Darstellung aktiv ist.
2. Klicken Sie im Netzwerk auf den Operand, für den Sie ein Symbol vereinbaren möchten.
3. Wählen Sie entweder den Menübefehl **Bearbeiten > Symbole...** oder den Kontextmenübefehl **Symbole bearbeiten....** Das Kontextmenü blenden sie durch klicken der rechten Maustaste beim markierten Objekt ein.
4. Tragen Sie im folgenden Dialogfeld das Symbol, den Datentyp des Operanden und ggf. den Kommentar ein. Das Symbol muss innerhalb der gesamten Symboltabelle eindeutig sein und darf maximal 24 Zeichen lang sein. Es sind keine Anführungszeichen zugelassen.
5. Bestätigen Sie Ihre Eingaben mit "OK". Das definierte Symbol wird in die Symboltabelle eingetragen und anstelle des markierten Operanden in den Anweisungsteil eingefügt.

6.1.7.3 Einfügen von Symbolzeilen

Um vor der Mauszeiger-Position eine Symbolzeile einzufügen, wählen Sie den Menübefehl **Einfügen > Zeile vor Markierung/Zeile nach Markierung**.

Um eine oder mehrere Symbolzeilen aus der Zwischenablage einzufügen, haben Sie folgende Möglichkeiten:

- Klicken Sie mit der linken Maustaste auf die Zeile (nicht auf den Zeilenkopf), ab der die Symbolzeile(n) eingefügt werden soll(en).
- Wählen Sie den Menübefehl **Bearbeiten > Einfügen**.
- Klicken Sie auf die entsprechende Schaltfläche in der Symbolleiste.
- Betätigen Sie CTRL + V.

6.1.7.4 Löschen von Symbolzeilen

Um die markierten Symbolzeilen zu löschen und eine Kopie davon in der Zwischenablage zu behalten, haben Sie folgende Möglichkeiten:

- Wählen Sie den Menübefehl **Bearbeiten > Ausschneiden**.
- Klicken Sie auf die entsprechende Schaltfläche in der Symbolleiste.
- Betätigen Sie CTRL + X.

Um die markierten Symbolzeilen ohne Sicherungskopie zu löschen, haben Sie folgende Möglichkeiten:

- Wählen Sie den Menübefehl **Bearbeiten > Löschen**.
- Betätigen Sie DEL.

Beachten Sie, dass das Ausschneiden und das Löschen der speziellen Objekteigenschaften nicht wieder rückgängig gemacht werden kann.

6.1.7.5 Filtern der Symboltabelle

Um einen Filter für die Anzeige des aktuellen Fensters einzustellen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf die Schaltfläche "Filter bearbeiten...".
2. Wählen Sie im Dialogfeld "Filter bearbeiten" über die entsprechende Nummer einen vorhandenen Filter aus oder definieren Sie einen neuen Filter.
3. Klicken Sie dazu auf die Schaltfläche "Neuer Filter".
4. Geben Sie dem neuen Filter einen eindeutigen Namen.
5. Wählen Sie die gewünschten Einstellungen.
6. Klicken Sie auf die Schaltfläche "Übernehmen". Der angelegte Filter ist nun auch in der Klappliste anwählbar.

Es werden nur die Symbole angezeigt, die den aktiven Filterkriterien entsprechen. Sie können mehrere Kriterien gleichzeitig verwenden. Die angegebenen Filterkriterien werden miteinander verknüpft.

oder

Wählen Sie einen schon vorhandenen Filter aus der Klappliste zur Auswahl eines Filters.

Folgende vordefinierte Filter sind vorhanden:


- Alles anzeigen: zeigt alle Symbole an (Voreinstellung)
- Fehlerhafte: zeigt alle doppelten oder syntaktisch falschen Symbole an
- Unbenutzte: zeigt alle unbenutzten Symbole an
- Eingänge: zeigt alle Symbole für Eingänge an
- Ausgänge: zeigt alle Symbole für Ausgänge an
- Merker: zeigt alle Symbole für Merker an
- Bausteine: zeigt alle Symbole für Bausteine an
- Zeiten & Zähler: zeigt alle Symbole für Zeiten & Zähler an

Die vordefinierten Filter können weder verändert noch gelöscht werden. Die vordefinierten Filter können aber dupliziert und unter einem neuen Namen übernommen werden.

6.1.7.6 Nicht verwendete Symbole

Sie erhalten mit der Auswahl des vordefinierten Filters "unbenutzte" in der Symboltabelle einen Überblick über alle Symbole mit folgender Eigenschaft:

- Die Symbole sind in der Symboltabelle definiert.
- Die Symbole werden aber innerhalb der Anwenderprogrammteile nicht verwendet.

Das Fragezeichen  markiert ein unbenutztes Symbol. Die Symbolzelle wird zusätzlich blau hinterlegt.

6.1.7.7 Operanden ohne Symbol

Aktivieren Sie das Optionskästchen "Operanden ohne Symbol anzeigen" in der Symboltabelle, um nicht nur die Operanden mit Symbolen sondern alle im Programm verwendete Operanden anzuzeigen, die von dem aktuellen Filter erfasst werden.

6.1.7.8 Sortieren der Symboltabelle

Um das Sortierkriterium für die Anzeige der aktuellen Tabelle einzustellen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf die Spaltenüberschrift, so dass die Sortierung ausgelöst wird. Ein blauer senkrechter Pfeil am rechten Rand erscheint.
2. Wählen Sie die gewünschte Sortierrichtung, die durch die Pfeilrichtung dargestellt wird.

Die Symboltabelle wird nach den Einträgen in dieser Spalte sortiert. Durch Wiederholung der Aktion können Sie die Sortierreihenfolge umkehren.

6.1.7.9 Markieren von Symbolzeilen

Um die Symbolzeile zu markieren, in der sich der Mauszeiger befindet, haben Sie folgende Möglichkeiten:

- Klicken Sie auf den Zeilenkopf links neben der gewünschten Symbolzeile.
- Betätigen Sie SHIFT + LEERTASTE.

Um alle Zeilen der aktuellen Symboltabelle zu markieren, haben Sie folgende Möglichkeiten:

- Wählen Sie den Menübefehl **Bearbeiten > Markieren > Alles**.
- Betätigen Sie CTRL + A.

Um eine Markierung aufzuheben, wählen Sie den Menübefehl **Bearbeiten > Markierung aufheben**.

6.1.7.10 Kopieren von Symbolzeilen in die Zwischenablage

Um eine oder mehrere markierte Symbolzeilen in die Zwischenablage zu kopieren, haben Sie folgende Möglichkeiten:

- Wählen Sie den Menübefehl **Bearbeiten > Kopieren**.
- Klicken Sie auf die entsprechende Schaltfläche in der Symbolleiste.
- Betätigen Sie CTRL + C.

Der bisherige Inhalt der Zwischenablage wird dabei überschrieben.

6.1.7.11 Speichern einer Symboltabelle

Die Symboltabelle wird nicht explizit gespeichert. Nach dem Verlassen eines Feldes in der Symboltabelle wird der Inhalt implizit zwischengespeichert und steht unmittelbar z. B. beim Programmieren zur Verfügung. Die Symboltabelle wird zusammen mit dem Projekt gespeichert (Menübefehl **Datei > Speichern** bzw. **Datei > Speichern unter**).

6.1.8 Schritte zum Ändern der Fenster-Einstellungen

6.1.8.1 Ein-/Ausschalten der Symbolleiste

Um die Anzeige der Symbolleiste ein- oder auszuschalten, wählen Sie den Menübefehl **Ansicht > Symbolleiste**.

Bei eingeschalteter Symbolleiste wird der Menübefehl durch ein Häkchen gekennzeichnet.

6.1.8.2 Ein-/Ausschalten der Statusleiste

Um die Anzeige der Statusleiste ein- oder auszuschalten, wählen Sie den Menübefehl **Ansicht > Statusleiste**.

Bei eingeschalteter Statusleiste wird der Menübefehl durch ein Häkchen gekennzeichnet.

6.1.8.3 Anordnen der Symbolleiste

Um die Anordnung der angezeigten Symbolleiste zu ändern, gehen Sie folgendermaßen vor:

1. Bewegen Sie den Mauszeiger auf einen freien Bereich der jeweiligen Leiste.
2. Halten Sie die linke Maustaste gedrückt und ziehen Sie die Leiste zur gewünschten Position.
3. Lassen Sie die linke Maustaste wieder los.

6.1.8.4 Einstellen der Anzeigegröße eines Fensters

Um die Anzeigegröße in den Sichten Bausteineditor, Projektdetails und Programmdetails **schrittweise** zu vergrößern, haben Sie folgende Möglichkeiten:

- Wählen Sie den Menübefehl **Ansicht > Vergrößern**.
- Betätigen Sie CTRL + Num+.

Um die Anzeigegröße im aktuellen Fenster schrittweise zu verkleinern, haben Sie folgende Möglichkeiten:

- Wählen Sie den Menübefehl **Ansicht > Verkleinern**.
- Betätigen Sie CTRL + Num-.

Um die Anzeigegröße auf einen **bestimmten Wert** einzustellen, haben Sie folgende Möglichkeiten:

1. Wählen Sie den Menübefehl **Ansicht > Zoomfaktor**.
2. Wählen Sie im Dialogfeld "Zoomfaktor" den gewünschten Zoomfaktor.
3. Bestätigen Sie mit "OK".

6.2 Arbeiten mit Bausteinen

6.2.1 Bausteineditor

Der Bausteineditor ermöglicht Ihnen das Erstellen und Testen von Bausteinen für die CPUs der SIMATIC S7-300 in den Programmiersprachen KOP, FUP und AWL.

Mit diesen Programmiersprachen können Sie die Bausteine einzeln durch inkrementelle Eingabe erstellen und bearbeiten.

Neben der reinen Programmerstellung, d.h. dem Erzeugen und Bearbeiten von Codebausteinen, Datenbausteinen und anwenderdefinierten Datentypen werden Ihnen zusätzliche Funktionen zum Programmieren, Testen und Inbetriebnehmen des Programms angeboten:

- Programmieren mit Symbolen
- Auslesen von Statusinformationen und Betriebsdaten der CPU über den Menübefehl Baugruppenzustand (Menü Extras)
- Anzeigen und Ändern der Betriebsart der CPU (Menü Extras)
- Urlöschen von CPUs
- Anzeigen und Einstellen von Datum und Uhrzeit der CPU mit dem Menübefehl CPU-Einstellungen (Menü Extras)
- Beobachten eines einzelnen Codebausteins (Testen von AWL-/KOP-/FUP-Programmen im Programmstatus)
- Multiinstanzfähigkeit, d.h. ein Instanz-DB kann die Daten mehrerer FBs enthalten:
 - Wissenswertes zu Instanz-Datenbausteinen
 - Deklarieren von Multiinstanzen
- Variablendeklarationstabelle:
 - in ihr können mehrere Deklarationen zusammen bearbeitet werden (Kopieren, Ausschneiden, Einfügen)
 - Variablendeklarationstabelle und der Anweisungsteil von Codebausteinen werden zusammen in einem geteilten Arbeitsfenster dargestellt.

Voraussetzung für das Erzeugen und Bearbeiten von Bausteinen ist das Vorhandensein eines Projektes.

6.2.2 Auswahl der Programmiersprache

6.2.2.1 Programmiersprachen des Bausteineditors

In welcher Programmiersprache Sie einen Baustein erstellen wollen, legen Sie bei der Erzeugung des Bausteins im Dialogfeld "Neuer Baustein" fest. Sie können die Erstsprache auch bei geöffnetem Baustein im Register "Eigenschaften" ändern.

Bausteineditor starten

Den Bausteineditor starten Sie durch einen Doppelklick auf den entsprechenden Baustein oder über den Kontextmenübefehl **Öffnen**, der beim markierten Objekt mit der rechten Maustaste eingeblendet wird.

Zur Erstellung des Programms stehen Ihnen die in der Tabelle angegebenen Programmiersprachen zur Verfügung.

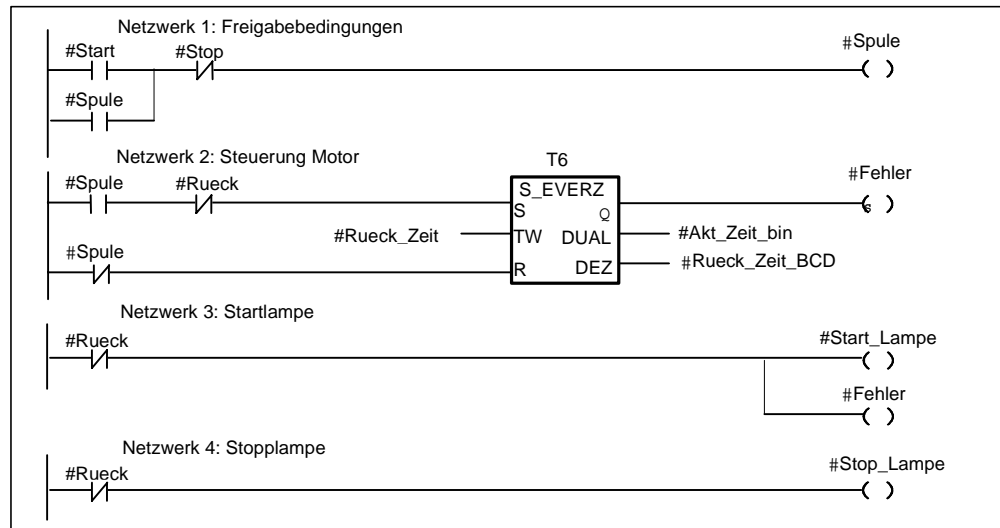
Programmiersprache	Anwendergruppe	Anwendungsfall	Inkrementelle Eingabe	Baustein aus der CPU rückdokumentierbar
Anweisungsliste AWL	Anwender, die maschinennah programmieren wollen.	Laufzeit- und speicherplatz-optimierte Programme	•	•
Kontaktplan KOP	Anwender, die Umgang mit Stromlaufplänen gewohnt sind.	Programmierung von Verknüpfungssteuerungen	•	•
Funktionsplan FUP	Anwender, die mit den logischen Boxen der Booleschen Algebra vertraut sind.	Programmierung von Verknüpfungssteuerungen	•	•

Bei fehlerfreien Bausteinen können Sie zwischen den Darstellungen des Bausteins in den Programmiersprachen KOP, FUP und AWL wechseln. Dabei werden Programmteile, die in der Zielsprache nicht darstellbar sind, in AWL dargestellt.

6.2.2.2 Programmiersprache KOP (Kontaktplan)

Die Darstellung in der grafischen Programmiersprache KOP (Kontaktplan) ist Stromlaufplänen nachempfunden. Die Elemente eines Stromlaufplans, wie z. B. Öffner und Schließer, werden zu Netzwerken zusammengefügt. Ein oder mehrere Netzwerke ergeben den gesamten Anweisungsteil eines Codebausteins.

Beispiel für Netzwerke in KOP:

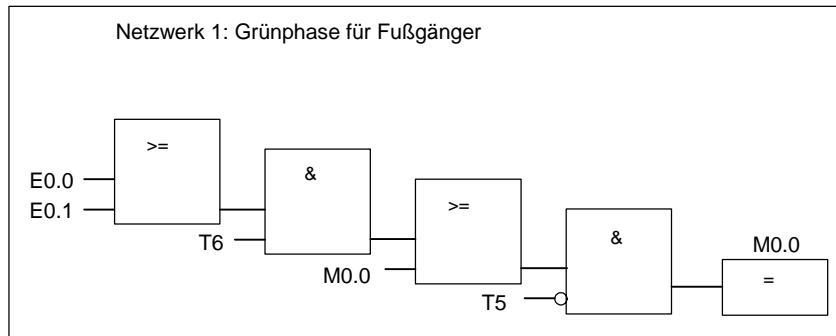


Die Programmiersprache KOP ist Bestandteil von STEP 7 Lite.

6.2.2.3 Programmiersprache FUP (Funktionsplan)

Die Programmiersprache FUP (Funktionsplan) benutzt zur Darstellung der Logik die von der Booleschen Algebra bekannten grafischen Logiksymbole. Außerdem können komplexe Funktionen, wie z. B. mathematische Funktionen, direkt in Verbindung mit den logischen Boxen dargestellt werden.

Beispiel für ein Netzwerk in FUP:

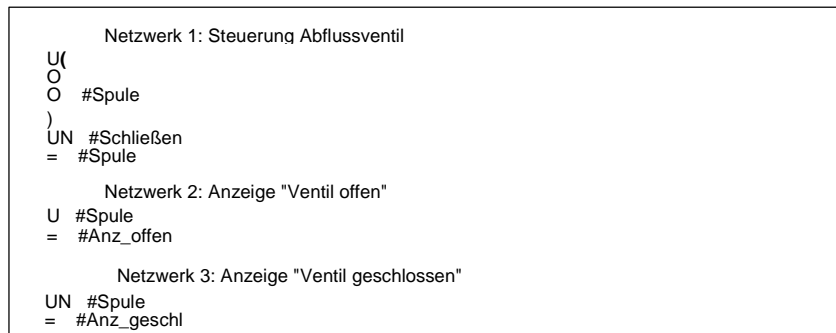


Die Programmiersprache FUP ist Bestandteil von STEP 7 Lite.

6.2.2.4 Programmiersprache AWL (Anweisungsliste)

Die Programmiersprache AWL (Anweisungsliste) ist eine maschinennahe textuelle Sprache. Die einzelnen Anweisungen entsprechen den Arbeitsschritten, mit denen die CPU die Abarbeitung des Programms durchführt. Mehrere Anweisungen können zu Netzwerken zusammengefügt werden.

Beispiel für Netzwerke in AWL:



Die Programmiersprache AWL ist Bestandteil von STEP 7 Lite.

6.2.3 Anlegen von Bausteinen

6.2.3.1 Anwenderdefinierte Datentypen (UDT)

Anwenderdefinierte Datentypen sind von Ihnen erzeugte spezielle Datenstrukturen, die Sie nach Ihrer Definition im gesamten Anwenderprogramm verwenden können.

- UDTs können wie elementare Datentypen oder zusammengesetzte Datentypen in der Variablendeklaration von Codebausteinen (FC, FB, OB) oder als Datentyp für Variablen in einem Datenbaustein (DB) verwendet werden. Damit haben Sie den Vorteil, dass Sie eine mehrmals zu verwendende, spezielle Datenstruktur nur einmal definieren und sie anschließend beliebig vielen Variablen zuweisen können.
- UDTs können als Vorlage für die Erstellung von Datenbausteinen mit gleicher Datenstruktur dienen, d. h. Sie erstellen nur einmal die Struktur und erzeugen anschließend die benötigten Datenbausteine durch einfache Zuweisung des UDTs (Beispiel Rezepturen: Die Struktur des DB ist immer gleich, nur die Mengenangaben sind unterschiedlich).

Aufbau eines UDT

Nach dem Öffnen eines UDT wird Ihnen in einem neuen Arbeitsfenster die Tabellendarstellung für diesen anwenderdefinierten Datentyp in Deklarationsicht angezeigt.

- Die erste und die letzte Zeile beinhalten bereits die Deklarationen STRUCT bzw. END_STRUCT für Beginn und Ende des anwenderdefinierten Datentyps. Diese Zeilen können Sie nicht editieren.
- Sie bearbeiten den anwenderdefinierten Datentyp, indem Sie Ihre Eingaben ab der zweiten Zeile der Deklarationstabelle in den entsprechenden Spalten durchführen. Die standardmäßig vom Programm eingetragene Variable "udt_bool" können Sie verändern bzw. überschreiben.
- Sie können anwenderdefinierte Datentypen strukturieren aus:
 - elementaren Datentypen
 - zusammengesetzten Datentypen
 - bereits vorhandenen anwenderdefinierten Datentypen

Die anwenderdefinierten Datentypen des Anwenderprogramms werden nicht auf die S7-CPU geladen. Sie werden direkt im Bausteineditor erstellt und bearbeitet.

6.2.3.2 Bausteineigenschaften

Mit Hilfe von Bausteineigenschaften können Sie die erstellten Bausteine besser identifizieren (z. B. bei der Versionspflege).

In der Sicht "Eigenschaften" können Sie die Bausteineigenschaften editieren, wenn Sie einen Baustein ausgewählt haben. Neben editierbaren Eigenschaften werden hier auch Daten nur zu Ihrer Information angezeigt: Diese können Sie nicht bearbeiten.

Achtung

Die Änderungen, die Sie im Feld "Symbol" bzw. "Symbolkommentar" vornehmen, werden sofort gespeichert. Auch wenn Sie die Eigenschaften-Sicht verlassen ohne zu speichern, werden die Änderungen in diesen Feldern trotzdem übernommen und in allen Sichten entsprechend angepasst.

Hinweis

Die Mnemonik, in der Sie Ihre Bausteine programmieren wollen, legen Sie unter **Extras > Einstellungen...** fest.

Tabelle der Bausteineigenschaften

Eigenschaft	Bedeutung	Editierbar bzw. Anwählbar	Beispiel
Projektspezifische Eigenschaften			
Name	Bausteinname (Typ und Nummer)	Nein	FB10
Erstelsprache	Aktuelle Erstel- bzw. Editiersprache	Ja	AWL
Symbol	Symbolischer Name des Bausteins	Ja	PID-Regler
Symbolkommentar	Kommentar zum Symbolischen Namen des Bausteins	Ja	Reglersteuerung
Zeitstempel			
Code erstellt	Zeitpunkt der Erstellung des Bausteins	Nein	24.08.2000 09:13:16
Code zuletzt geändert	Zeitpunkt der letzte Speicherung des Bausteins mit geändertem Code	Nein	24.08.2000 09:13:16
Schnittstelle zuletzt geändert	Zeitpunkt der letzte Speicherung des Bausteins mit geänderter Schnittstelle	Nein	24.08.2000 09:13:16
Kommentar	Kommentar zum Baustein: 1.Feld Titel 2.Feld Kommentar	Ja	Funktionsbaustein für Regelsteuerung

Eigenschaft	Bedeutung	Editierbar bzw. Anwählbar	Beispiel
Baustein-Header Eigenschaften			
Name (Header)	Bausteinname (max. 8 Zeichen und vom Autor festgelegt)	Ja	PID
Version	Versionsnummer des Bausteins (beide Zahlen zwischen 0..15, d. h. 0.0 - 15.15)	Ja	3.10
Familie	Name der Bausteinfamilie (max. 8 Zeichen, ohne Leerzeichen).	Ja	Regler
Autor	Name des Autors, Firmenname, Abteilungsname oder andere Namen (max. 8 Zeichen, ohne Leerzeichen)	Ja	Siemens
Längen			
Lokaldaten	Größe der Lokaldaten in Byte	Nein	10
Ladespeicherbedarf	Größe des Ladespeicherbedarfs in Byte	Nein	142
MC7	Größe des MC7 Codes in Byte	Nein	38
Arbeitsspeicherbedarf	Größe des Arbeitsspeicherbedarfs in Byte	Nein	74
Attribute			
DB ist schreibgeschützt im AS	Schreibschutz für Datenbausteine; seine Daten können von der AS nur gelesen und nicht vom Anwenderprogramm verändert werden.	Ja	
Bausteinschutz	Ein Baustein, der mit dieser Option übersetzt wurde, kann von keinem Erstellsystem (STEP 7 Lite, STEP 7) angezeigt oder verändert werden.	Nein	
Standard Baustein	Ein SIEMENS Standardbaustein, der einen Bausteinschutz besitzt. Eingabefelder für Name, Familie, Version und Autor sind grau dargestellt und können nicht editiert werden.	Nein	
Unlinked	Ein Datenbaustein mit der Eigenschaft UNLINKED wird nicht in das Programm eingebunden.	Ja	
Multiinstanzfähig	Nur Funktionsbausteine und Systemfunktionsbausteine können als multiinstanzfähige Bausteine erzeugt werden. Multiinstanzfähige FB/SFB können die Instanz-Datenbausteine anderer FB/SFB nutzen.	Nein	
Non-Retain	Datenbausteine mit diesem Attribut werden nach jedem Netz-Aus- und Netz-Einschalten und nach jedem STOP-RUN-Übergang der CPU auf die Ladewerte zurückgesetzt.	Ja	

Der Bausteinschutz hat folgende Konsequenzen:

- Wenn Sie später einen übersetzten Baustein im Bausteineditor ansehen wollen, kann der Anweisungsteil des Bausteins nicht eingesehen werden.
- In der Variablendeklarationstabelle des Bausteins werden nur die Variablen der Deklarationstypen var_in, var_out und var_in_out angezeigt. Die Variablen der Deklarationstypen var_stat und var_temp bleiben verborgen.

Zuordnung Bausteineigenschaften zu Bausteinararten

Die nachfolgenden Tabelle zeigt, welche Bausteineigenschaften bei welchen Bausteinararten vereinbart werden können.

Eigenschaft	OB	FB	FC	DB	UDT
Bausteinschutz	•	•	•	•	–
Autor	•	•	•	•	–
Familie	•	•	•	•	–
Name	•	•	•	•	–
Version	•	•	•	•	–
Unlinked	–	–	–	•	–
DB ist schreibgeschützt im AS	–	–	–	•	–
Non-Retain	–	–	–	•	–

6.2.3.3 Bausteinschutz setzen

Setzen Sie den Bausteinschutz ein, wenn Sie ihr Sach- und Fachwissen vor Nachahmern schützen wollen oder ungewollte Manipulationen an Bausteinen vermeiden wollen.

Der gesetzte Bausteinschutz bewirkt:

- Der geschützte Baustein kann weder im Projekt noch in der CPU geändert werden.
- Von geschützten Baustein kann lediglich der Deklarationsteil und der Name eingesehen werden; der Anweisungsteil bzw. der Datenteil bleibt verborgen. In der Variablendeklarationstabelle des Bausteins werden nur die Variablen der Deklarationstypen var_in, var_out und var_in_out angezeigt. Die Variablen der Deklarationstypen var_stat und var_temp bleiben verborgen.

Beim "Bausteinschutz setzen" wird eine ungeschützte Kopie des Bausteins in einer Exportdatei gesichert. Bei Bedarf kann die ungeschützte Kopie des Bausteins wieder importiert werden.

Gehen Sie wie folgt vor, um den Bausteinschutz zu setzen:

1. Markieren Sie im Projektfenster die zu schützenden Bausteine (beachten Sie, dass die zu schützenden Bausteine nicht geöffnet sein dürfen).
2. Wählen Sie den Menübefehl **Extras > Bausteinschutz** setzen.
3. Wählen Sie im Folgedialog den Namen und den Pfad der Exportdatei.
4. Starten Sie den Vorgang "Bausteinschutz setzen" mit Betätigung der Schaltfläche "Exportieren". Nach Abschluss des Vorgangs werden die Bausteine im Projektfenster mit einem kleinen Vorhängeschloss gekennzeichnet.

Hinweis

Wählen Sie jeweils eine neue Exportdatei, wenn Sie nachträglich weitere Bausteine schützen wollen.

6.2.3.4 Zulässige Bausteineigenschaften je Bausteinart

Die nachfolgenden Tabelle zeigt, welche Bausteineigenschaften bei welchen Bausteinarten vereinbart werden können.

Eigenschaft	OB	FB	FC	DB	UDT
Bausteinschutz	•	•	•	•	–
Autor	•	•	•	•	–
Familie	•	•	•	•	–
Name	•	•	•	•	–
Version	•	•	•	•	–
Unlinked	–	–	–	•	–
DB ist schreibgeschützt im AS	–	–	–	•	–

Schreibschutz auf Datenbausteine einrichten

Sie können für Datenbausteine einen Schreibschutz einrichten, indem Sie das Optionskästchen "DB ist schreibgeschützt im AS" aktivieren. Die Datenbausteine können beim Programmablauf nicht überschrieben werden.

6.2.3.5 Anzeige von Baustein-Längen

Baustein-Längen werden in der Einheit "Byte" angezeigt.

Anzeige in der Sicht "Programmstruktur"

In dieser Ansicht werden folgende Längen angezeigt:

- Summe aller Baustein-Längen ohne Systemdaten im Ladespeicher der CPU
- Summe aller Baustein-Längen ohne Systemdaten im Arbeitsspeicher der CPU

Anzeige in der Sicht "Eigenschaften" eines Bausteins

In dieser Ansicht werden angezeigt:

- Benötigte Anzahl an Lokaldaten: Größe der Lokaldaten in Byte
- MC7: Größe des MC7-Codes in Byte bzw. Größe der DB-Nutzdaten
- Größe im Ladespeicher der CPU (Ladespeicherbedarf)
- Größe im Arbeitsspeicher der CPU (Arbeitsspeicherbedarf)

Die Anzeigen sind unabhängig davon, OB der Baustein in der Ansicht Online oder einer Offline vorliegt.

6.2.3.6 Bausteine vergleichen

Gehen Sie folgendermaßen vor:

1. Markieren Sie den Baustein oder nur die Bausteine, die Sie vergleichen möchten.
2. Wählen Sie den Menübefehl **Extras > Vergleichen > Baustein**.
3. Die Ergebnisse des Vergleichs (ONLINE/offline) werden in einem Dialogfeld "Bausteine vergleichen-Ergebnisse" angezeigt.
4. Wählen Sie aus der Vergleichsliste einen Baustein aus.
5. Betätigen Sie die Schaltfläche "Details", damit die Bausteininformationen dargestellt werden.

Schritte zum Anlegen von Bausteinen

Gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Datei > Neu > Baustein**.
2. Geben Sie im Dialogfeld "Neuer Baustein" die gewünschten Einstellungen des neu zu erstellenden Bausteins an.
3. Bestätigen Sie mit "OK".

Der Baustein wird angelegt und im "Bausteineditor" geöffnet. Der obere Teil des Fensters dient zur Bearbeitung der Variablendeklarationstabelle, im unteren Teil programmieren Sie den Anweisungsteil.

Hinweis

Wenn Sie einen Funktionsbaustein (FB) erzeugen, so wird beim Anlegen bereits festgelegt, OB Sie in diesem FB Multiinstanzen deklarieren können.

Anlegen von Datenbausteinen (DB)

Datenbausteine können wie alle anderen Bausteine angelegt werden.

1. Wählen Sie den Menübefehl **Datei > Neu > Baustein** oder klicken Sie auf das entsprechende Symbol in der Symbolleiste.
2. Geben Sie im Dialogfeld den neu zu erstellenden Datenbaustein an. DB 0 dürfen Sie als DB-Nummer nicht vergeben, da diese Nummer für das System reserviert ist.
3. Wählen Sie im Dialogfeld "Neuer Baustein" aus, welche Art von Datenbaustein Sie erzeugen wollen:
 - Datenbaustein (Global-Datenbaustein)
 - Datenbaustein mit zugeordnetem UDT (Global-Datenbaustein)
 - Datenbaustein mit zugeordnetem FB (Instanz-Datenbaustein)

Beim letzten Punkt müssen Sie außerdem den FB auswählen, zu dem der Instanz-Datenbaustein gehören soll.

Hinweis

STEP 7 Lite bietet die Möglichkeit, unter bestimmten Voraussetzungen die Daten für verschiedene FBs in einem einzigen Datenbaustein abzulegen (Multiinstanz-Datenbaustein, siehe Eingeben der Multiinstanz in der Variablendeklarationstabelle).

Einstellen der Bausteineigenschaften

1. Öffnen Sie den Baustein durch Doppelklick oder blenden sie durch klicken der rechten Maustaste ein Kontextmenü ein. Mit dem Menübefehl **Öffnen** können Sie dann den markierten Baustein öffnen.
2. Wählen Sie bei geöffnetem Baustein das Register "Eigenschaften".
3. Tragen Sie Name, Familie, Symbol, Version und Autor des Bausteins ein. Die Angabe von Namen und Familie unterstützt Sie bei der Programmierung von Bausteinaufrufen in KOP. Die Änderungen, die Sie im Feld "Symbol" vornehmen, werden sofort übernommen und in allen Sichten entsprechend angepasst. Das Register enthält noch folgende Angaben:
 - Länge vom Baustein, MC7-Code und Lokaldaten
 - DB ist schreibgeschützt im AS
 - Bausteinschutz: Ein Baustein mit dieser Eigenschaft ist ein geschützter Baustein, der nicht bearbeitet werden kann.
 - Standardbaustein
 - Unlinked

6.2.4 Arbeiten mit Bibliotheken

6.2.4.1 Übersicht der Bausteinbibliotheken

Bibliotheken dienen zur Ablage von wiederverwendbaren Programmkomponenten für die SIMATIC S7. Standardbibliotheken, die z. B. die System- und Standardfunktionen der Familie S7-300 enthalten, sind Bestandteil von STEP 7 Lite.

Die Standardbibliotheken sind automatisch am rechten Rand des Fensters im Register "Bausteine" verfügbar, wenn Sie diese über den Menübefehl **Ansicht > Bibliotheken** aufgerufen haben. Sie können zwischen den Registern "Befehle" und "Bausteine" umschalten.

STEP 7 Lite enthält folgende Bausteinbibliotheken:

- **IEC Function Blocks:** Bausteine für IEC-Funktionen, wie z. B. für die Bearbeitung von Zeit- und Datumsangaben, für Vergleichsoperationen, für String-Bearbeitung und zur Auswahl von Maximum und Minimum
- **Organization Blocks:** Standard-Organisationsbausteine (OB)
- **PID Control Blocks:** Funktionsbausteine (FB) zur PID-Regelung
- **S5-S7 Converting Blocks:** Bausteine für die Konvertierung von STEP 5-Programmen
- **System Function Blocks:** Systemfunktionen (SFC) und Systemfunktionsbausteine (SFB)
- **TI-S7 Converting Blocks:** allgemein verwendbare Standardfunktionen

6.3 Erstellen von Codebausteinen

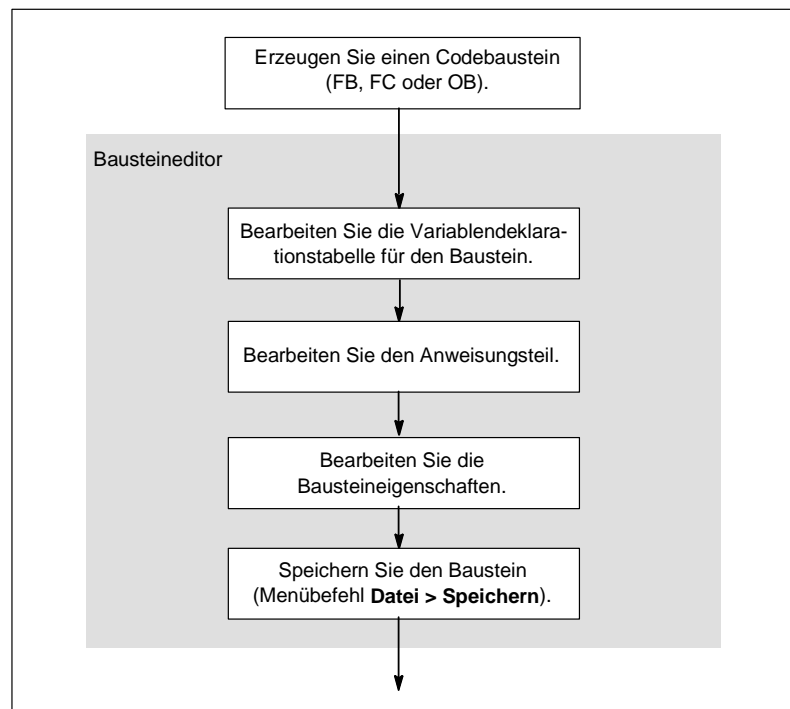
6.3.1 Grundlagen zum Erstellen von Codebausteinen

6.3.1.1 Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen

Codebausteine (OBs, FBs, FCs) bestehen aus einer Variablendeklarationstabelle, einem Anweisungsteil und besitzen zusätzlich Eigenschaften. Bei der Programmierung müssen Sie also folgende drei Teile bearbeiten:

- **Variablendeklarationstabelle:** In der Variablendeklarationstabelle legen Sie die Parameter sowie lokale Variablen des Bausteins fest.
- **Anweisungsteil:** Im Anweisungsteil programmieren Sie den Bausteincode, der vom Automatisierungssystem bearbeitet werden soll. Er besteht aus einem oder mehreren Netzwerken. Für die Erstellung der Netzwerke stehen Ihnen z. B. die Programmiersprachen AWL (Anweisungsliste), KOP (Kontaktplan) und FUP (Funktionsplan) zur Verfügung.
- **Bausteineigenschaften:** Die Bausteineigenschaften beinhalten zusätzliche Informationen wie Zeitstempel oder Pfadangabe, die vom System eingetragen werden. Daneben können Sie selbst Angaben zu Namen, Familie, Version und Autor machen.

Prinzipiell ist es gleich, in welcher Reihenfolge Sie die Teile eines Codebausteins bearbeiten. Selbstverständlich können Sie auch korrigieren/ergänzen.



Hinweis

Wenn Sie auf Symbole der Symboltabelle zurückgreifen wollen, sollten Sie diese zuerst auf Vollständigkeit überprüfen und ggf. ergänzen.

6.3.1.2 Voreinstellungen für Bausteineditor KOP/FUP/AWL

Bevor Sie mit der Programmierung beginnen, sollten Sie die Einstellmöglichkeiten kennenlernen, um möglichst komfortabel und Ihren Gewohnheiten entsprechend arbeiten zu können.

Über den Menübefehl **Extras > Einstellungen...** öffnen Sie einen Dialog. Sie können in diesem Dialogfeld Voreinstellungen für das Programmieren von Bausteinen einstellen, z. B. :

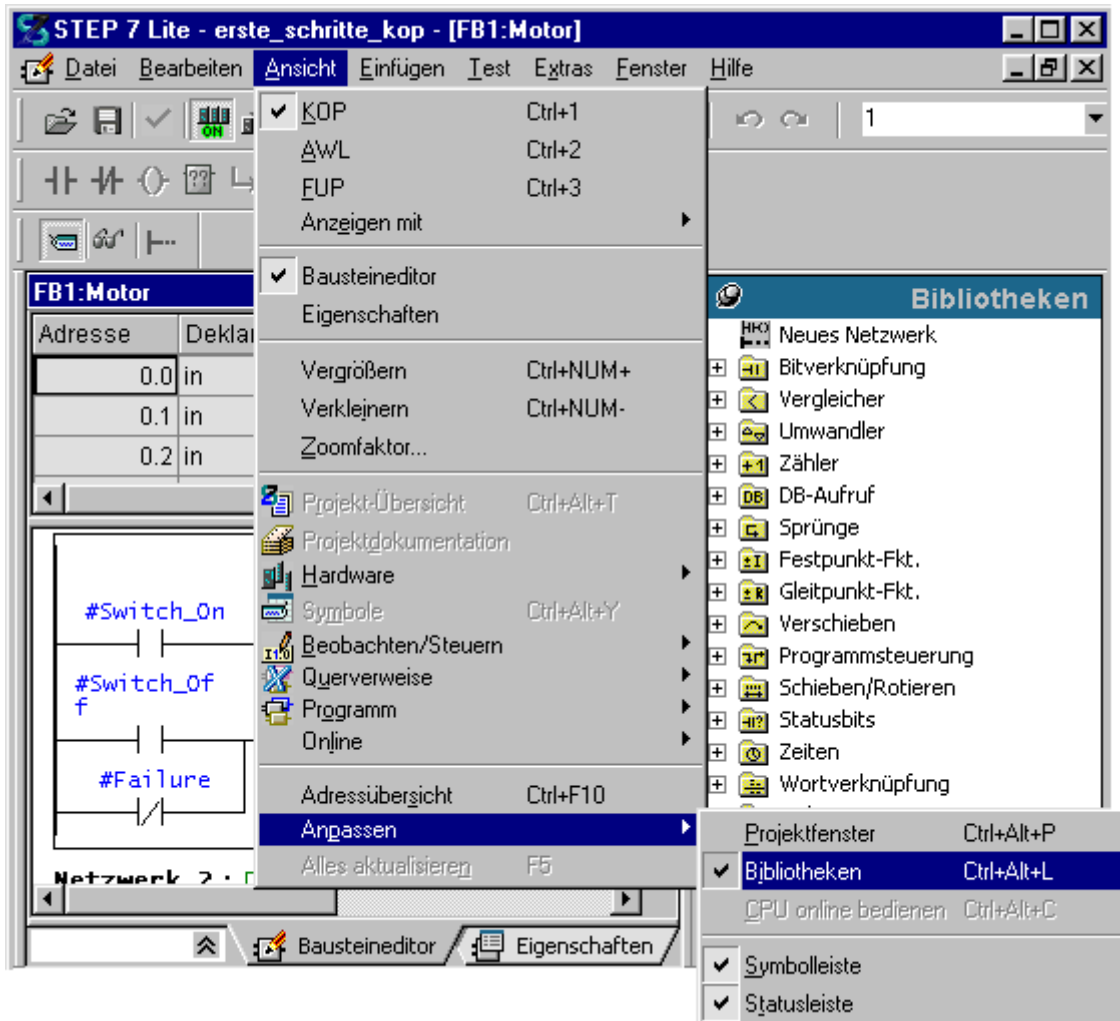
- die Schrift (Art und Größe) in Text und Tabellen.
- OB Sie bei einem neuen Baustein zunächst Symbolik und Kommentar angezeigt bekommen möchten.
- die Farbe, in der Markierungen von Netzwerken oder Anweisungszeilen dargestellt werden.

Die Einstellungen für Sprache, Kommentar und Symbolik können Sie aktuell während der Bearbeitung über Befehle des Menüs **Ansicht > ...** ändern.

6.3.1.3 Anweisungen aus den Befehlsbibliotheken

Die "Befehlsbibliotheken" stellen KOP- und FUP-Operationen sowie bereits deklarierte Multiinstanzen zur Verfügung. Sie lassen sich über den Menübefehl **Ansicht > Anpassen > Bibliotheken** aufrufen. Das gewünschte markierte Element im Register "Befehle" können Sie entweder mit dem Kontextmenübefehl "Einfügen", mit Doppelklick oder per Drag&Drop an die selektierte Stelle im Anweisungsteil einfügen.

Beispiel der Befehlsbibliothek bei KOP



6.3.1.4 Festlegen der Ansicht des Bausteineditors

Verkleinern/Vergrößern der Ansicht

Sie können für jedes Arbeitsfenster (Datenbaustein, Codebaustein) die Ansicht inklusive der Schriftart stufenweise verkleinern oder vergrößern.

Gehen Sie folgendermaßen vor:

1. Aktivieren Sie das Fenster, dessen Inhalt Sie um eine Stufe verkleinern oder vergrößern möchten.
2. Wählen Sie den Menübefehl **Ansicht > Verkleinern** bzw. **Ansicht > Vergrößern**. Die aktuelle Anzeige wird um eine Stufe verkleinert bzw. vergrößert, bis die minimale Größe erreicht ist.

Zoomen der Ansicht

Sie können für jedes Arbeitsfenster (Datenbaustein oder Codebaustein) die Ansicht inklusive der Schriftart durch Angabe eines Faktors vergrößern, verkleinern oder die Standardgröße wiederherstellen.

Gehen Sie folgendermaßen vor:

1. Aktivieren Sie das Fenster, für dessen Inhalt Sie die Größe der Darstellung ändern möchten.
2. Wählen Sie den Menübefehl **Ansicht > Zoomfaktor...**
3. Geben Sie im Dialogfeld die gewünschten Einstellungen an und bestätigen Sie mit "OK".

Einstellen der Fensterteilung

Codebausteine werden jeweils in geteilten Arbeitsfenstern angezeigt. Bei den Codebausteinen beinhaltet die obere Hälfte die Variablendeklarationstabelle, die untere den Anweisungsteil. Sie können die Größe der Bereichsaufteilung verstellen.

Gehen Sie folgendermaßen vor:

- Klicken Sie mit dem Mauszeiger auf die Teilungslinie und ziehen Sie den Mauszeiger bei gedrückter linker Maustaste in die Richtung, in der Sie die Teilungslinie verschieben möchten.

Einstellen von Spaltenbreiten

Die Breite der einzelnen Spalten der Variablendeklarationstabellen können Sie verstellen.

Gehen Sie folgendermaßen vor:

- Positionieren Sie den Mauszeiger zwischen zwei Spalten auf eine senkrechte Begrenzungslinie in der Spaltenüberschrift, so dass er als senkrechter Doppelpfeil dargestellt wird.
- Bei gedrückter linker Maustaste können Sie nun die Spaltenbreite durch horizontale Bewegungen der Maus grafisch bestimmen.

Hinweis

Wenn Sie auf die Überschrift der ausgewählten Spalte doppelklicken, wird die Spaltenbreite optimiert.

Umschalten der Programmiersprache

Zum Programmieren von Bausteinen stehen Ihnen die Programmiersprachen Anweisungsliste (AWL), Funktionsplan (FUP) und Kontaktplan (KOP) zur Verfügung.

1. Wählen Sie bei geöffnetem Baustein den Menübefehl **Ansicht > AWL/KOP/FUP...** oder das Register "Eigenschaften".
2. Wählen Sie im Feld "Erstsprache" die gewünschte Programmiersprache aus.

Hinweis

- Ein Umschalten der Programmiersprachen von KOP nach FUP und umgekehrt ist jederzeit möglich.
- Ein Umschalten der Programmiersprache von AWL nach KOP/FUP geht nur für die AWL-Anweisungen, die die komplette Versorgung von Parametern der entsprechenden KOP/FUP-Elemente nachbilden und die Reihenfolge einhalten. Die nicht verwendeten Parameter in AWL sind mit "NOP 0" zu versorgen.

6.3.2 Editieren der Variablendeklarationstabelle

6.3.2.1 Verwendung der Variablendeklaration in Codebausteinen

Nach dem Öffnen eines Codebausteins erscheint ein Fenster mit der Variablendeklarationstabelle des Bausteins im oberen Teil und dem Anweisungsteil im unteren Teil, in dem Sie den eigentlichen Bausteincode bearbeiten.

Beispiel: Variablendeklarationstabelle und Anweisungsteil in AWL

Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
0.0	in	Switch_On	BOOL	FALSE	Motor einschalten
0.1	in	Switch_Off	BOOL	FALSE	Motor ausschalten
0.2	in	Failure	BOOL	FALSE	Motorstörung, führt zum Ausschalten
2.0	in	Actual_Speed	INT	0	tatsächliche Motordrehzahl
4.0	out	Engine_On	BOOL	FALSE	Motor wird eingeschaltet
4.1	out	Preset_Speed	BOOL	FALSE	Solldrehzahl erreicht
	in_out				
6.0	stat	Preset_Speed	INT	1500	geforderte Motordrehzahl


```

U      #Switch_On
UN     "Automatikbetrieb"
S      #Engine_On
U(
O      #Switch_Off
ON     #Failure
)
R      #Engine_On
NOP    0
    
```

In der Variablendeklarationstabelle legen Sie die lokalen Variablen einschließlich der Formalparameter des Bausteins fest. Das hat u. a. folgende Auswirkungen:

- Durch das Deklarieren wird entsprechend Speicherplatz reserviert, für temporäre Variablen im Lokaldaten-Stack, im Fall von Funktionsbausteinen, für statische Variablen im später zugeordneten Instanz-DB.
- Durch die Festlegung von Eingangs-, Ausgangs- und Durchgangsparametern legen Sie auch die "Schnittstelle" für den Aufruf des Bausteins im Programm fest.
- Wenn Sie die Variablen in einem Funktionsbaustein deklarieren, bestimmen diese Variablen (mit Ausnahme der temporären) auch die Datenstruktur für jeden Instanz-DB, der dem FB zugeordnet wird.

6.3.2.2 Zusammenspiel zwischen Variablendeklarationstabelle und Anweisungsteil

Variablendeklarationstabelle und Anweisungsteil von Codebausteinen sind eng miteinander verbunden, da im Anweisungsteil die Namen aus der Variablendeklarationstabelle verwendet werden. Änderungen in der Variablendeklaration wirken sich daher im gesamten Anweisungsteil aus.

Aktion in der Variablendeklaration	Reaktion im Anweisungsteil
Korrekte Neueingabe	Falls ungültiger Code vorhanden, wird zuvor nicht deklarierte Variable nun gültig.
Korrekte Namensänderung ohne Typänderung	Symbol wird sofort überall mit neuem Namen dargestellt.
Korrechter Name wird in einen ungültigen Namen geändert	Code wird nicht geändert.
Ungültiger Name wird in einen korrekten geändert	Falls ungültiger Code vorhanden, wird dieser gültig.
Typänderung	Falls ungültiger Code vorhanden, wird dieser gültig, und falls gültiger Code vorhanden, wird dieser unter Umständen ungültig.
Löschen einer Variable (Symbol), die im Code verwendet wird.	Gültiger Code wird ungültig

Kommentaränderungen, die fehlerhafte Eingabe einer neuen Variablen, eine Anfangswertänderung oder das Löschen einer nicht verwendeten Variablen haben keine Auswirkung auf den Anweisungsteil.

6.3.2.3 Aufbau der Variablendeklarationstabelle

Die Variablendeklarationstabelle beinhaltet Einträge für Adresse, Deklarationstyp, Name, Datentyp, Anfangswert und Kommentar der Variablen. Jede Tabellenzeile steht für eine Variablendeklaration. Variablen vom Datentyp ARRAY oder STRUCT benötigen mehrere Zeilen.

Zulässige Datentypen für die Lokaldaten der verschiedenen Bausteinararten finden Sie in Zuordnen von Datentypen zu Lokaldaten von Codebausteinen.

Spalte	Bedeutung	Anmerkungen	Bearbeitung
Adresse	Adresse im Format BYTE.BIT	Bei Datentypen, für die mehr als ein Byte erforderlich ist, zeigt die Adresse die Zuordnung durch einen Sprung zur nächsten Byte-Adresse an. Zeichenlegende: * : Größe eines Feldelements in Byte. + : Anfangsadresse, rel. zum Strukturanfang = : gesamter Speicherbedarf einer Struktur	Systemeintrag: Die Adresse wird vom System vergeben und angezeigt, wenn Sie die Eingabe einer Deklaration beenden.
Name	Symbolischer Name der Variablen	Der Name muss mit einem Buchstaben beginnen. Reservierte Schlüsselwörter sind nicht erlaubt.	erforderlich
Deklaration	Deklarationstyp, "Verwendungszweck" der Variablen	Möglich sind je nach Bausteinarart: Eingangsparameter "in" Ausgangsparameter "out" Durchgangsparameter "in_out" Statische Variablen "stat" Temporäre Variablen "temp"	Systemvorgabe entsprechend Bausteinarart
Typ	Datentyp der Variable (BOOL, INT, WORD, ARRAY usw.).	Die Datentypen können Sie über das Menü der rechten Maustaste auswählen.	erforderlich
Anfangswert	Anfangswert, wenn die Software nicht den Default-Wert übernehmen soll	Muss mit Datentyp kompatibel sein. Der Anfangswert wird beim erstmaligen Speichern eines DBs als aktueller Wert für die Variable übernommen, falls Sie nicht explizit einen aktuellen Wert festlegen.	optional
Kommentar	Kommentar zur Dokumentation		optional

Vorbelegung

Nachdem Sie einen neuerstellten Codebaustein geöffnet haben, wird eine voreingestellte Variablendeklarationstabelle angezeigt. Diese listet nur die für die gewählte Bausteinarart zulässigen Deklarationstypen (in, out, in_out, stat, temp) auf und zwar in der vorgeschriebenen Reihenfolge.

Beim Anlegen eines neuen Organisationsbausteins wird die Variablendeklarationstabelle mit den Lokaldaten vorbelegt, die für den jeweiligen Organisationsbaustein vorgesehen sind. Diese 20 Bytes beinhalten eine vom Betriebssystem zur Verfügung gestellte, systemeinheitliche Startinformation. Diese Information beinhaltet sowohl Einstellungen für das Verhalten der Organisationsbausteine, als auch Informationen wie z. B. Prioritätsklasse, Nummer des OBs, Kennungen für verursachende Ereignisse usw. Diese Informationen werden zur Laufzeit des OBs eingetragen und können z. B. zu Diagnosezwecken ausgelesen werden.

Nicht editierbare Spalten in der Variablendeklarationstabelle

Spalte	Eintrag
Adresse	Der Eintrag wird vom System vergeben und angezeigt, wenn Sie die Eingabe einer Deklaration beenden.
Deklarationstyp	Der Deklarationstyp wird durch die Position der Deklaration innerhalb der Tabelle bestimmt. Dadurch wird sichergestellt, dass Variablen nur in der richtigen Reihenfolge der Deklarationstypen eingegeben werden können. Wollen Sie den Deklarationstyp einer Deklaration ändern, schneiden Sie die Deklaration zunächst aus und fügen Sie diese danach unter dem neuen Deklarationstyp wieder ein.

6.3.2.4 Allgemeine Hinweise zu Variablendeklarationstabellen

Zur Bearbeitung der Tabelle stehen Ihnen die bekannten Funktionen des Menüs **Bearbeiten** zur Verfügung. Zur einfachen Bearbeitung können Sie auf das kontextsensitive Menü der rechten Maustaste zugreifen. Bei der Eingabe des Datentyps werden Sie ebenfalls durch das Menü der rechten Maustaste unterstützt.

Markieren in Variablendeklarationstabellen

Einzelne Zeilen markieren Sie, indem Sie auf das zugehörige, schreibgeschützte Adressfeld klicken. Zusätzliche Zeilen des selben Deklarationstyps markieren Sie bei gedrückter SHIFT-Taste. Die markierten Zeilen werden schwarz hinterlegt.

ARRAYs markieren Sie durch Klicken auf das Adressfeld der betreffenden Zeile.

Wollen Sie eine **Struktur markieren**, dann führen Sie einen Mausklick auf das Adressfeld der ersten oder letzten Zeile aus (in der also das Schlüsselwort STRUCT oder END_STRUCT steht). Einzelne Deklarationen innerhalb einer Struktur markieren Sie durch einen Mausklick auf das entsprechende Adressfeld der Zeile.

Wenn Sie Strukturen innerhalb anderer Strukturen eingeben, wird Ihnen die Hierarchie durch entsprechende Einrückung der Variablennamen angezeigt.

Rückgängig machen von Aktionen

In der Variablendeklarationstabelle können Sie mit dem Menübefehl **Bearbeiten > Rückgängig** jeweils die letzte Ausschneide- oder Lösch-Operation rückgängig machen.

6.3.2.5 Schritte zum Arbeiten mit der Variablendeklarationstabelle

Einfügen von Leerzeilen in Variablendeklarationstabellen

Vor einer Zeile

1. Stellen Sie den Mauszeiger in die betreffende Zeile der Tabelle.
2. Wählen Sie den Menübefehl Einfügen > Zeile vor Markierung.

Nach einer Zeile

- Stellen Sie den Mauszeiger in das Feld "Kommentar" dieser Zeile und betätigen Sie RETURN oder
- wählen Sie den Menübefehl Einfügen > Zeile Nach Markierung.

Eingeben von einfachen Datentypen in die Variablendeklarationstabelle

Bei der Neueingabe einer Deklaration gehen Sie folgendermaßen vor:

1. Geben Sie hinter dem gewünschten Deklarationstyp den Variablennamen ein.
2. Bewegen Sie die Einfügemarke nach der Eingabe mit der TAB-Taste in das benachbarte Feld.
3. Geben Sie ebenso ein:
 - Datentyp,
 - Anfangswert (optional),
 - Kommentar (optional).

Nach Abschluss einer Zeile wird der Variablen eine Adresse zugewiesen.

Nach jeder Bearbeitung eines Tabellenfeldes erfolgt eine Syntaxprüfung, die eventuell vorhandene Fehler rot anzeigt. Sie müssen diese Fehler nicht sofort beheben, sondern können mit der weiteren Bearbeitung fortfahren und die Korrekturen zu einem späteren Zeitpunkt durchführen.

Eingeben von Datenelementen des Datentyps STRUCT

1. Zum Vereinbaren des Datentyps haben Sie folgende Möglichkeiten:
 - Positionieren Sie den Mauszeiger in das Feld der Spalte "Typ" und wählen Sie den Menübefehl **Einfügen > Datentyp > Zusammengesetzte Typen > STRUCT**.
 - Wählen Sie das Feld der Spalte "Typ" aus und betätigen Sie die rechte Maustaste. Wählen Sie dann im kontextsensitiven Menü den entsprechenden Datentyp.
 - Geben Sie im Feld der Spalte "Typ" das Schlüsselwort STRUCT ein.
2. Tragen Sie in der Spalte "Name" einen Namen ein und beenden Sie die Eingabe in der Tabellenzeile mit der TAB- oder RETURN-Taste. Es werden Ihnen daraufhin zunächst eine Leerzeile und die Abschlusszeile der Deklaration (END_STRUCT) eingeblendet.

- Geben Sie die Elemente der Struktur ein, indem Sie deren Name, Datentyp, Anfangswert (optional) und ggf. einen Kommentar festlegen. Weitere Zeilen können Sie über die Funktionen des Menüs "Einfügen" oder durch Betätigung der RETURN-Taste erzeugen, bzw. können Sie Variablen über das Menü "Bearbeiten" vervielfältigen oder wieder löschen.

Adresse	Deklaration	Name	Typ	Anfangswert	Komm.
0.0	in	New_Struct	STRUCT		
+0.0	in	var1	BOOL	FALSE	
+2.0	in	var2	INT	0	
+4.0	in	var3	WORD	W#16#0	
=6.0	in		END_STRUCT		
6.0	in	field1	ARRAY[1..20,1..40]		
*0.1	in		BOOL		

Eingabe des Datentyps ARRAY in die Variablendeklarationstabelle

- Positionieren Sie den Mauszeiger in das Feld der Spalte "Typ" in der Variablendeklarationstabelle.
- Wählen Sie den Menübefehl **Einfügen > Datentyp > Zusammengesetzte Typen > ARRAY**. Daraufhin wird ARRAY in das markierte Feld eingetragen. Sie können den Begriff auch direkt über die Tastatur eingeben.
- Geben Sie unmittelbar nach ARRAY eine öffnende eckige Klammer, die Index-Untergrenzen, zwei Punkte, die Index-Obergrenzen und eine schließende eckige Klammer ein, z. B. ARRAY[1..14] für ein eindimensionales Feld oder ARRAY[1..20,1..24] für ein zweidimensionales Feld.
- In das Feld der Spalte "Anfangswert" können Sie die Anfangswerte für die einzelnen Elemente des ARRAYS eintragen (siehe nachfolgende Beispiele).
- In das Feld der Spalte "Kommentar" können Sie Anmerkungen zum ARRAY eintragen.
- Beenden Sie die Eingabe in der Tabellenzeile mit der TAB- oder RETURN-Taste.
- Tragen Sie in der automatisch erzeugten zweiten Zeile den Datentyp der ARRAY-Elemente ein.

Beispiele zur Eingabe von Anfangswerten in ARRAYS

- **Einzelvorgabe:**
Sie geben den einzelnen Elementen einen eigenen Anfangswert. Die Werte werden durch Kommata getrennt aufgelistet.
- **Wiederholungsfaktor:**
Sie geben mehreren Elementen den gleichen Anfangswert vor. Die Angabe des Wertes erfolgt in runder Klammer, vorangestellt ist der Wiederholungsfaktor für die Anzahl der Elemente.

Typ	Anfangswert	Erklärung
ARRAY[1..14]	1234	Nur dem ersten ARRAY-Element wird der Anfangswert 1234 zugewiesen.
ARRAY[1..14]	1234, 56, 78, 90	Den ersten vier ARRAY-Elementen werden die Anfangswerte 1234, 56, 78, 90 in dieser Reihenfolge zugewiesen.
ARRAY[1..14]	14 (9876)	Allen 14 ARRAY-Elementen wird der Anfangswert 9876 zugewiesen.

Kopieren von Variablen in Variablendeklarationstabellen

1. Markieren Sie die zu kopierende Variablen folgendermaßen:
 - Klicken Sie auf das Feld "Adresse" (Anwahl einer Variablen).
 - Klicken Sie bei gleichzeitiger Betätigung der SHIFT-Taste wiederum mit der linken Maustaste auf das Feld "Adresse". Alle Zeilen zwischen der ersten markierten Zeile und der soeben markierten Variable sind dadurch mit ausgewählt (Anwahl mehrerer Variablen).
2. Wählen Sie den Menübefehl **Bearbeiten > Kopieren** oder das entsprechende Symbol der Symbolleiste.
3. Stellen Sie den Mauszeiger an die Stelle, hinter der Sie die kopierte Variable einfügen wollen und wählen Sie den Menübefehl **Bearbeiten > Einfügen** oder das entsprechende Symbol der Symbolleiste.

Die kopierten Variablen werden eingefügt. Damit die Namen der Variablen eindeutig bleiben, werden die Namen der kopierten Variablen automatisch mit einer laufenden Nummer ergänzt.

Löschen von Variablen in Variablendeklarationstabellen

1. Markieren Sie die zu löschenden Variablen folgendermaßen:
 - Klicken Sie auf das Feld "Adresse" (Anwahl einer Variablen).
 - Klicken Sie bei gleichzeitiger Betätigung der SHIFT-Taste wiederum mit der linken Maustaste auf das Feld "Adresse". Alle Zeilen zwischen der ersten markierten Zeile und der soeben markierten Variable sind dadurch mit ausgewählt (Anwahl mehrerer Variablen).
2. Wählen Sie den Menübefehl **Bearbeiten > Ausschneiden** oder den Menübefehl **Bearbeiten > Löschen** bzw. die entsprechenden Symbole der Symbolleisten.

Hinweis

Beachten Sie beim Löschen von ARRAYS und STRUCTs:

- Wenn Sie die erste Zeile eines ARRAY zum Löschen markieren, wird die zugehörige zweite Zeile mit markiert.
 - Wenn Sie die erste Zeile eines STRUCT zum Löschen markieren, werden auch alle Zeilen bis END STRUCT mit markiert.
-

Variieren der Spaltenbreite

Die Breite der Tabellenspalten können Sie variieren. Gehen Sie folgendermaßen vor:

1. Positionieren Sie den Mauszeiger zwischen zwei Spalten auf eine senkrechte Begrenzungslinie in der Spaltenüberschrift, so dass er als senkrechter Doppelpfeil dargestellt wird.
2. Bewegen Sie den Mauszeiger bei gedrückter linker Maustaste in horizontaler Richtung.

Wenn Sie also ganz auf die optionale Eingabe von Kommentar oder Anfangswerten verzichten wollen, können Sie diese Spalten auf diese Weise verändern, um sich ganz auf die anderen Spalten konzentrieren zu können.

6.3.3 Multiinstanzen in der Variablendeklarationstabelle

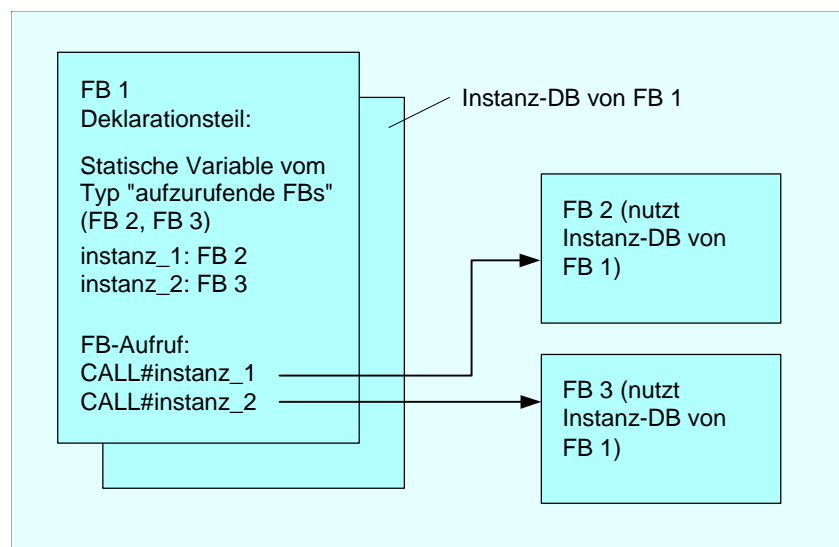
6.3.3.1 Verwendung von Multiinstanzen

Möglicherweise wollen oder können Sie aufgrund der Leistungsdaten (z. B. Speicherplatz) der verwendeten S7-CPU nur eine beschränkte Anzahl von Datenbausteinen für Instanzdaten spendieren. Wenn in Ihrem Anwenderprogramm in einem FB weitere, bereits vorhandene Funktionsbausteine aufgerufen werden (Aufrufhierarchie von FBs), dann können Sie diese weiteren Funktionsbausteine ohne eigene (d. h. zusätzliche) Instanz-DBs aufrufen.

Dazu bietet sich folgende Lösung an:

- Nehmen Sie die aufzurufenden FBs als statische Variablen in die Variablendeklaration des aufrufenden FBs auf.
- In diesem Funktionsbaustein rufen Sie weitere Funktionsbausteine auf ohne eigene (d. h. zusätzliche) Instanz-DBs.
- Damit erreichen Sie eine Konzentrierung der Instanzdaten in einem Instanz-Datenbaustein, d. h. Sie können die verfügbare Anzahl der DBs besser ausnutzen.

Das folgende Beispiel zeigt die beschriebene Lösung: FB 2 und FB 3 nutzen den Instanz-DB des Funktionsbausteins FB 1, von dem aus sie aufgerufen wurden.



Einzige Voraussetzung: Sie müssen dem aufrufenden Funktionsbaustein "mitteilen", welche Instanzen Sie aufrufen und von welchem (FB-) Typ diese Instanzen sind. Die Angaben sind im Deklarationsteil des aufrufenden FBs zu machen. Der zu verwendende FB muss mindestens eine Variable oder einen Parameter aus dem Datenbereich (also nicht VAR_TEMP) haben.

Verwenden Sie keine Multiinstanzdatenbausteine, solange Online-Änderungen bei laufender CPU zu erwarten sind. Ein stoßfreies Nachladen ist nur bei Verwendung von Instanzdatenbausteinen sichergestellt.

6.3.3.2 Regeln für die Bildung von Multiinstanzen

Für die Deklaration von Multiinstanzen gelten folgende Regeln:

- Für das Deklarieren von Multiinstanzen muss der Funktionsbaustein als multiinstanzfähiger FB erzeugt worden sein.
- Dem Funktionsbaustein, in dem eine Multiinstanz deklariert wird, muss ein Instanz-DB zugeordnet werden.
- Eine Multiinstanz kann nur als statische Variable deklariert werden (Deklarationstyp "stat").

Hinweis

Sie können auch Multiinstanzen zu Systemfunktionsbausteinen erzeugen.

6.3.3.3 Eingeben der Multiinstanz in der Variablendeklarationstabelle

1. Öffnen Sie den FB, von dem aus die unterlagerten FBs aufgerufen werden sollen.
2. Legen Sie in der Variablendeklarationstabelle des aufrufenden FBs eine statische Variable für jeden Aufruf eines Funktionsbausteins fest, für dessen Instanz Sie keinen Instanz-Datenbaustein angeben wollen:
 - Positionieren Sie in eine Leerzeile mit der Deklaration "stat" in der 2. Spalte.
 - Geben Sie hinter dem Deklarationstyp "stat" in Spalte "Name" eine Bezeichnung für den FB-Aufruf ein.
 - Geben Sie in Spalte "Typ" den aufzurufenden FB absolut oder mit seinem symbolischen Namen an.
 - Eventuelle Erläuterungen können Sie in die Kommentarspalte eintragen.

Aufrufe im Anweisungsteil

Wenn Sie Multiinstanzen deklariert haben, können Sie FB-Aufrufe ohne Angabe eines Instanz-DB verwenden.

Beispiel: Ist die statische Variable "Name: Motor_1 , Datentyp: FB20" definiert, lässt sich die Instanz folgendermaßen aufrufen:

```
Call Motor_1      // Aufruf des FB 20 ohne Instanz-DB
```

6.3.4 Allgemeine Hinweise zum Editieren von Anweisungen und Kommentaren

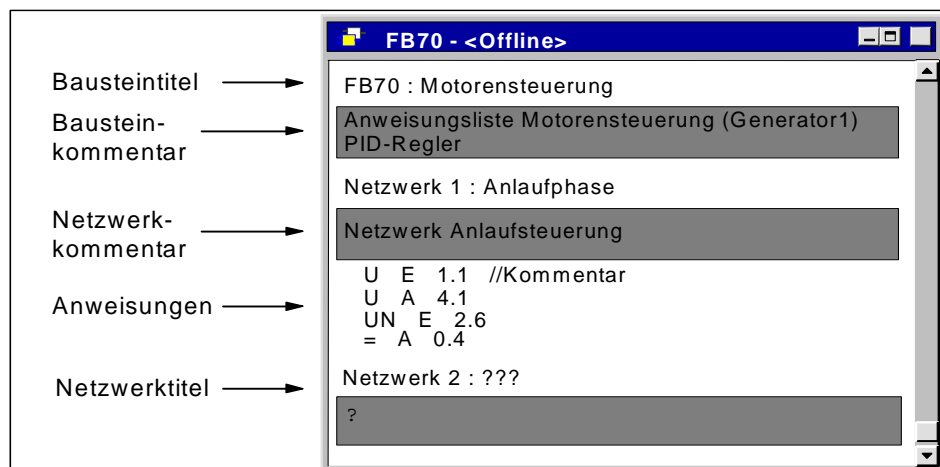
6.3.4.1 Aufbau des Anweisungsteils

Im Anweisungsteil beschreiben Sie den Programmablauf Ihres Codebausteins. Sie geben dazu die jeweiligen Anweisungen in Abhängigkeit der gewählten Programmiersprache in Netzwerken ein. Der Bausteineditor führt nach der Eingabe einer Anweisung sofort eine Syntaxprüfung durch und zeigt Ihnen Fehler rot und kursiv an.

Der Anweisungsteil für einen Codebaustein besteht in den häufigsten Fällen aus mehreren Netzwerken, die wiederum aus einer Liste von Anweisungen bestehen.

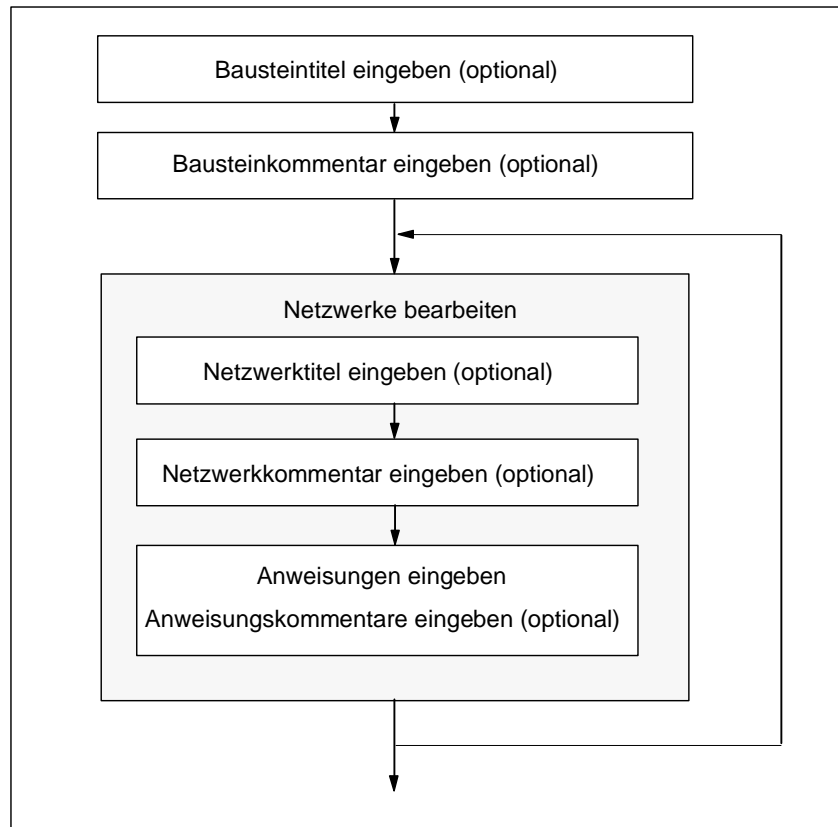
In einem Anweisungsteil können Sie Bausteintitel, Bausteinkommentare, Netzwerktitel, Netzwerkkommentare und Anweisungszeilen innerhalb der Netzwerke bearbeiten.

Aufbau des Anweisungsteils am Beispiel der Programmiersprache AWL



6.3.4.2 Vorgehensweise beim Eingeben von Anweisungen

Die Bestandteile des Anweisungsteils können Sie grundsätzlich in beliebiger Reihenfolge bearbeiten. Wenn Sie zum ersten Mal einen Baustein programmieren, empfiehlt sich folgendes Vorgehen:



Änderungen können Sie zusätzlich zum Einfügemodus auch im Überschreibmodus vornehmen. Zwischen beiden Modi wechseln Sie mit der INSERT-Taste.

6.3.4.3 Eingeben von globalen Symbolen in ein Programm

Mit dem Menübefehl **Einfügen > Symbol** können Sie Symbole in den Anweisungsteil Ihres Programms einfügen. Steht die Einfügemarke am Beginn, am Ende oder innerhalb einer Zeichenkette, so ist bereits das Symbol markiert, das mit dieser Zeichenkette beginnt - falls ein solches Symbol existiert. Wenn Sie die Zeichenkette ändern, wird die Markierung in der Liste nachgeführt.

Trennzeichen für Beginn und Ende einer Zeichenkette sind z. B. Leerzeichen, Punkt, Doppelpunkt. Innerhalb von globalen Symbolen werden keine Trennzeichen interpretiert.

Beim Einfügen von Symbolen können Sie folgendermaßen vorgehen:

1. Geben Sie die Anfangsbuchstaben des gewünschten Symbols im Programm ein.
2. Drücken Sie gleichzeitig die CTRL-Taste und die J-Taste, um die Liste der Symbole anzuzeigen. Das erste Symbol mit den eingegebenen Anfangsbuchstaben ist bereits markiert.
3. Übernehmen Sie das Symbol durch Drücken der RETURN-Taste oder wählen Sie ein anderes Symbol aus.

Daraufhin wird das in doppelte Hochkommas eingeschlossene Symbol anstelle der Anfangsbuchstaben eingefügt.

Generell gilt: Steht die Einfügemarke am Beginn, am Ende oder innerhalb einer Zeichenkette, so wird beim Einfügen eines Symbols diese Zeichenkette durch das in doppelte Hochkommas eingeschlossene Symbol ersetzt.

6.3.4.4 Titel und Kommentare zu Bausteinen und Netzwerken

Kommentare erhöhen die Lesbarkeit Ihres Anwenderprogramms und ermöglichen somit eine effektivere Inbetriebnahme und evt. Fehlersuche. Sie sind ein wichtiger Bestandteil der Programm-Dokumentation und sollten auf jeden Fall verwendet werden.

Kommentare für KOP, FUP und AWL-Programme

Folgende Kommentare stehen Ihnen zur Verfügung:

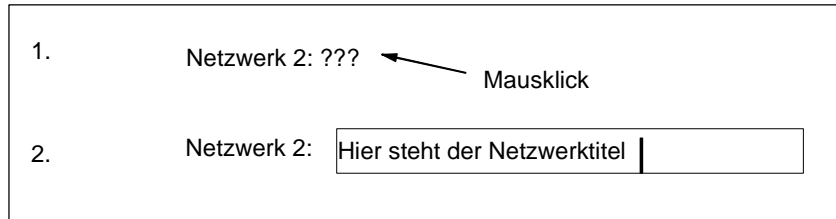
- Bausteintitel: Überschrift eines Bausteins (max. 64 Zeichen).
- Bausteinkommentar: Dokumentieren des gesamten Codebausteins, wie z. B. der Zweck des Bausteins.
- Netzwerktitel: Überschrift eines Netzwerks (max. 64 Zeichen).
- Netzwerkkommentar: Dokumentieren der Funktion einzelner Netzwerke.
- Kommentarspalte der Variablendeklarationstabelle: Kommentare für die deklarierten Lokaldaten.
- Symbolkommentar: Kommentare, die bei der Definition des Namens in der Symboltabelle für einen Operanden angegeben worden sind.
Sie können sich diese Kommentare mit dem Menübefehl **Ansicht > Anzeigen mit > Symbolinformation** anzeigen lassen.

Bausteintitel und Netzwerktitel sowie Bausteinkommentare oder Netzwerkkommentare können Sie im Anweisungsteil eines Codebausteins eingeben.

Baustein- oder Netzwerktitel

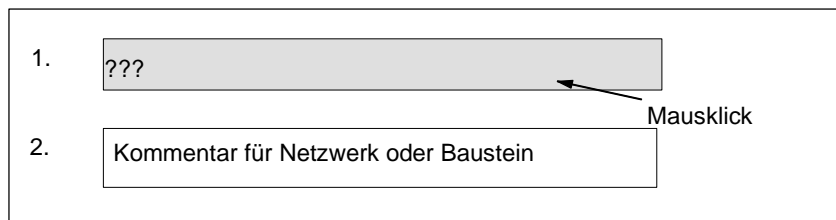
Um Baustein- oder Netzwerktitel einzugeben, positionieren Sie die Einfügemarke auf die drei Fragezeichen rechts neben dem Bausteinnamen bzw. Netzwerknamen (z. B. Netzwerk 1 : ???). Es öffnet sich ein Textfeld, in das Sie den Titel eingeben. Dieser kann maximal 64 Zeichen lang sein.

Bausteincommentare beziehen sich auf den gesamten Codebaustein. Dort können Sie die Funktion desselben beschreiben. Netzwerkkommentare beziehen sich auf einzelne Netzwerke und beschreiben deren Einzelheiten.



Baustein- und Netzwerkkommentare

Sie können die Ansicht der grauen Kommentarfelder über den Menübefehl **Ansicht > Anzeigen mit > Kommentar** ein- bzw. ausblenden. Durch einen Doppelklick auf ein solches Kommentarfeld wird das Textfeld geöffnet, in das Sie nun Ihre Erläuterungen eingeben können. Pro Baustein stehen Ihnen 64 KByte für Baustein- und Netzwerkkommentare zur Verfügung.



6.3.4.5 Eingeben von Baustein-/Netzwerkkommentaren

1. Schalten Sie mit dem Menübefehl **Ansicht > Anzeigen mit > Kommentar** die Anzeige der Kommentare ein (Häkchen vor dem Menübefehl).
2. Positionieren Sie den Mauszeiger mit einem Mausklick in das graue Feld unter dem Bausteinnamen bzw. Netzwerknamen. Das zuvor graue Kommentarfeld wird weiß und umrandet dargestellt.
3. Geben Sie im geöffneten Textfeld den Kommentar ein. Je Baustein stehen Ihnen insgesamt 64 kBytes für Baustein- und Netzwerkkommentare zur Verfügung.
4. Verlassen Sie das Textfeld durch einen Mausklick außerhalb des Textfeldes, durch Betätigen der TAB-Taste oder mit der Tastenkombination SHIFT + TAB.
5. Schalten Sie ggf. mit dem Menübefehl **Ansicht > Anzeigen mit > Kommentar** die Anzeige der Kommentare wieder aus.

6.3.4.6 Suchfunktion für Fehler im Anweisungsteil

Aufgrund der roten Darstellung sind Fehler im Anweisungsteil leicht zu erkennen. Um leichter zu Fehlern zu navigieren, die außerhalb des gerade sichtbaren Bereiches liegen, bietet der Bausteineditor die zwei Suchfunktionen **Bearbeiten > Gehe zu > Vorheriger Fehler/Nächster Fehler** an.

Die Fehlersuche ist netzwerkübergreifend. Das heißt, es wird im gesamten Anweisungsteil gesucht, nicht nur innerhalb eines Netzwerkes oder des gerade sichtbaren Bereichs.

In der Ausgabeleiste werden Ihnen Hinweise zu den Fehlern angezeigt.

Fehlerkorrekturen und Änderungen können Sie auch im Überschreibmodus vornehmen. Zwischen Einfüge- und Überschreibmodus wechseln Sie mit der INSERT-Taste.

6.3.4.7 Umverdrahten

Ähnlich wie beim Umverdrahten von Adern an einer Klemmleiste, können Sie in STEP 7 Lite Operanden und Bausteine umverdrahten.

Sie können in Bausteinen folgende absolute Operanden umverdrahten:

- Eingänge, Ausgänge (z. B. E 1.3 nach E 10.4)
- Merker, Zeiten, Zähler (z. B: M 50.2 nach M 60.1)
- Peripherie Eingänge / Ausgänge (z. B. PAB 0 nach PAB 10)

Der Ort der Umverdrahtung wird durch die Auswahl der Codebausteine im Projektfenster bestimmt. Beim Umverdrahten werden alle Verwendungen der Operanden in den ausgewählten Codebausteinen umverdrahtet.

Folgende Baustein-Namen und Bausteinaufrufe können umverdrahtet werden:

- FCs, FBs

Beim Umverdrahten von Bausteinen werden diese umbenannt und ihre Aufrufe bzw. ihre Verwendung in Codebausteinen wird umverdrahtet.

Gehen Sie beim Umverdrahten folgendermaßen vor:

1. Markieren Sie entweder das Symbol "Programm" oder ein bzw. mehrere Bausteine im Projektfenster.
2. Wählen Sie den Menübefehl **Extras > Umverdrahten**. Die Ausführung dieser Funktion ist nur offline und bei Operandenvorrang "Absolut" möglich (Einstellung über Menübefehl **Extras > Einstellungen**, Abschnitt "Allgemein" im Dialogfeld). Zusätzlich müssen alle Codebausteinfenster geschlossen sein.
3. Tragen Sie im angezeigten Dialogfeld "Umverdrahten" die gewünschten Ersetzungen (Alter Operand / Neuer Operand) in die Tabelle ein.
4. Lassen Sie das Optionskästchen "Operandenbereiche" deaktiviert, wenn Sie ausschließlich den angegebenen Operanden umverdrahten möchten.

Aktivieren Sie das Optionskästchen "Operandenbereiche", wenn Sie alle enthaltenen Operanden im Bereich des Operanden umverdrahten möchten. Wird z. B. ein DWORD-Operand umverdrahtet und das Optionskästchen "Operandenbereiche" ist aktiviert, dann werden die Zugriffe auf den Operanden selbst und alle Word-, Byte-, und Bit-Zugriffe auf diesen Adressbereich umverdrahtet.

5. Bestätigen Sie mit "OK".

Mit der Betätigung der "OK"-Schaltfläche starten Sie das Umverdrahten. Nach der Umverdrahtung wird eine Protokolldatei aufgeblendet, die Auskunft über die getätigten Änderungen gibt. Die Protokolldatei enthält die Operandenliste, "bisheriger Operand" und "neuer Operand". Dazu werden die einzelnen Bausteine aufgeführt mit der Anzahl der Umverdrahtungen pro Codebaustein.

Sie können die Protokolldatei drucken (Menübefehl **Datei > Drucken**) oder zur Sicherung speichern (Menübefehl **Datei > Speichern unter**). Verwenden Sie beim Speichern einen anderen als den vorgegebenen Namen, da die Datei sonst beim nächsten Umverdrahten überschrieben wird.

Beim Umverdrahten ist folgendes zu beachten:

- Wenn Sie einen Baustein umverdrahten, dann darf der neue Baustein nicht bereits existieren. Falls der Baustein bereits existiert, wird der Eintrag pastellrot hinterlegt und der Umverdrahtungsvorgang wird nicht gestartet.
- Wenn Sie einen Funktionsbaustein (FB) umverdrahten, dann wird automatisch sein Instanz-DB dem umverdrahteten FB zugeordnet. Der Instanz-DB ändert sich nicht (d. h. die DB-Nummer bleibt erhalten).

6.3.5 Editieren von KOP-Anweisungen im Anweisungsteil

6.3.5.1 Einstellungen für die Programmiersprache KOP

Einstellen des Layouts für KOP

Sie können für die Programmerstellung in der Programmiersprache KOP das Layout festlegen. Das ausgewählte Format (DIN A4 Hochformat/Querformat/maximale Größe) wirkt sich auf die Anzahl der in einem Zweig darstellbaren KOP-Elemente aus.

1. Wählen Sie den Menübefehl **Extras > Einstellungen...**
2. Wählen Sie im nachfolgenden Dialogfeld im Abschnitt "KOP/FUP" unter Listenfeld "Layout" das gewünschte Format. Geben Sie die gewünschte Formatgröße an.

Einstellungen zum Drucken:

Wenn Sie den KOP-Anweisungsteil ausdrucken möchten, so sollten Sie bereits vor der Erstellung des Anweisungsteils das geeignete Seitenformat einstellen.

Grundeinstellungen unter **Extras > Einstellungen...**

Sie können über **Extras > Einstellungen...** Grundeinstellungen vornehmen, z. B. zum Layout und zur Operandenfeldbreite.

6.3.5.2 Regeln für die Eingabe von KOP-Anweisungen

Die Beschreibung der Sprache "KOP" finden Sie im Handbuch *KOP für S7-300/400 – Bausteine programmieren* oder in der Online-Hilfe zu KOP.

Ein KOP-Netzwerk kann aus mehreren Elementen in mehreren Zweigen bestehen. Alle Elemente und Zweige müssen untereinander verbunden sein, wobei die linke Stromschiene nicht als Verbindung zählt (IEC 1131-3).

Bei der Programmierung in KOP müssen Sie einige Regeln beachten. Auf eventuelle Fehler werden Sie durch Fehlermeldung hingewiesen.

Abschluss eines KOP-Netzwerks

Jedes KOP-Netzwerk muss einen Abschluss in Form einer Spule oder einer Box haben. Folgende KOP-Elemente dürfen Sie jedoch nicht als Abschluss eines Netzwerkes verwenden:

- Vergleichboxen
- Spulen für Konnektoren `_(#)_/`
- Spulen für positive `_(P)_/` oder negative `_(N)_/` Flankenbewertung

Platzierung von Boxen

Der Ausgangspunkt des Zweiges für einen Boxanschluss muss immer die linke Stromschiene sein. Im Zweig vor der Box dürfen jedoch logische Verknüpfungen oder andere Boxen vorhanden sein.

Platzierung von Spulen

Spulen werden automatisch an den rechten Rand des Netzwerks platziert, wo sie den Abschluss eines Zweiges bilden.

Ausnahmen: Spulen für Konnektoren `_(#)_/` und positive `_(P)_/` oder negative `_(N)_/` Flankenbewertung können weder ganz links noch ganz rechts im Zweig platziert werden. Auch in Parallelzweigen sind sie nicht zulässig.

Unter den Spulen gibt es solche, die eine boolesche Verknüpfung verlangen, und solche, die keine boolesche Verknüpfung haben dürfen.

- Spulen mit erforderlicher boolescher Verknüpfung:
 - Ausgang `_()`, Ausgang setzen `_(S)`, Ausgang rücksetzen `_(R)`
 - Konnektor `_(#)_/`, positive Flanke `_(P)_/`, negative Flanke `_(N)_/`
 - alle Zähler- und Timerspulen
 - Springe wenn 0 `_(JMPN)`
 - Master Control Relais einschalten `_(MCR<)`
 - Verknüpfungsergebnisse ins BIE-Register laden `_(SAVE)`
 - Springe zurück `_(RET)`

- Spulen, die keine boolesche Verknüpfung erlauben:
 - Master Control Relais Anfang _(MCRA)
 - Master Control Relais Ende _(MCRD)
 - Datenblock Öffnen _(OPN)
 - Master Control Relais ausschalten _(MCR>)

Alle anderen Spulen können wahlweise eine boolesche Verknüpfung haben oder auch nicht.

Folgende Spulen dürfen Sie **nicht als Parallelausgang** verwenden:

- Springe wenn 0 _(JMPN)
- Springe wenn 1 _(JMP)
- Bausteinaufruf _(CALL)
- Rücksprung _(RET)

Freigabeeingang/ Freigabeausgang

Die Beschaltung des Freigabeeingangs "EN" bzw. des Freigabeausgangs "ENO" von Boxen kann erfolgen, ist aber nicht zwingend erforderlich.

Entfernen und Ändern

Besteht eine Verzweigung aus nur einem Element, so wird mit dem Entfernen dieses Elements der gesamte Zweig entfernt.

Mit dem Entfernen einer Box werden mit Ausnahme des Hauptzweiges auch alle Zweige, die mit booleschen Eingängen der Box verbunden sind, entfernt.

Zum einfachen Austauschen von Elementen gleichen Typs steht Ihnen der Überschreibmodus zur Verfügung.

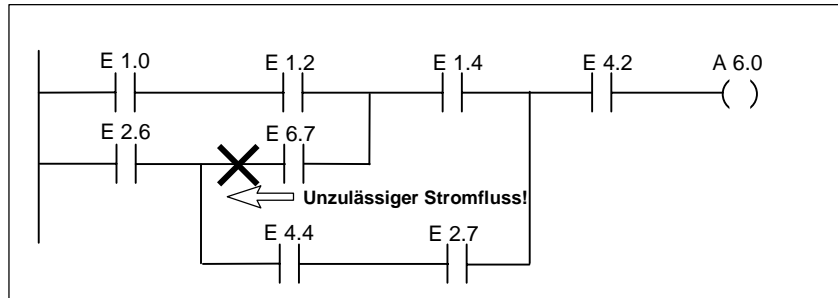
Parallele Verzweigungen

- Ergänzen Sie ODER-Verzweigungen von links nach rechts.
- Parallelzweige werden nach unten geöffnet und nach oben geschlossen.
- Parallelzweige werden immer hinter dem markierten KOP-Element geöffnet.
- Parallelzweige werden immer hinter dem markierten KOP-Element geschlossen.
- Um einen Parallelzweig zu löschen, müssen Sie alle KOP-Elemente dieses Zweiges löschen. Beim Entfernen des letzten KOP-Elements der Verzweigung wird der Rest der Verzweigung ebenfalls entfernt.

6.3.5.3 Unzulässige Verschaltungen in KOP

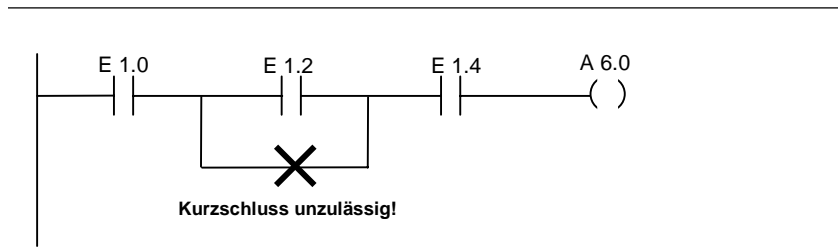
Stromfluss von rechts nach links

Es können keine Zweige editiert werden, die einen Stromfluss in umgekehrter Richtung verursachen könnten. Nachfolgendes Bild zeigt ein Beispiel. Beim Signalzustand "0" an E 1.4 käme es an E 6.7 zu einem Stromfluss von rechts nach links. Dieses ist nicht erlaubt.



Kurzschluss

Es können keine Zweige editiert werden, die einen Kurzschluss verursachen. Nachfolgendes Bild zeigt ein Beispiel:



6.3.5.4 Schritte zum Eingeben von KOP-Elementen

Eingeben von KOP-Elementen

1. Markieren Sie die gewünschte Stelle in einem Netzwerk, hinter der ein KOP-Element eingefügt werden soll.
2. Fügen Sie das gewünschte Element ein, indem Sie:
 - das entsprechende Symbol in der Symbolleiste betätigen (für Schließer, Öffner).
 - die entsprechende Funktionstaste F2 oder F3 (für Schließer, Öffner) betätigen.
 - auf das markierte Element in der Befehlsbibliothek doppelklicken oder es per Drag&Drop in den Bausteineditor ziehen.

Das ausgewählte KOP-Element wird eingefügt, wobei für die Adressen und Parameter der Platzhalter ??? eingefügt wird.

Hinweis

Sie können den Anweisungsteil auch durch Markieren vorhandener KOP-Elemente und anschließendem Auslösen der Menübefehle **Bearbeiten > Ausschneiden**, **Bearbeiten > Kopieren** oder **Bearbeiten > Einfügen** bearbeiten.

Eingeben und Editieren von Adressen oder Parametern in KOP-Elementen

Beim Einfügen eines KOP-Elements werden die Zeichenfolgen "???" und "..." als Platzhalter für Adressen bzw. Parameter eingesetzt.

Die rot dargestellte Zeichenfolge "???" kennzeichnet Adressen und Parameter, die beschaltet werden müssen.

Die schwarz dargestellte Zeichenfolge "..." kennzeichnet Adressen und Parameter, die beschaltet werden können.

1. Positionieren Sie mit Mausclick oder mit der TAB-Taste den Mauszeiger auf den Platzhalter.
2. Geben Sie die Adresse bzw. den Parameter in den Platzhalter ein (direkte oder indirekte Adressierung). Falls die Anzeige der Symbolauswahl eingeschaltet ist (Menübefehl **Ansicht > Anzeigen mit > Symbolauswahl**), wird eine Liste der bekannten Symbole angezeigt. Das mit den eingegebenen Zeichen beginnende Symbol ist markiert und kann durch Drücken der RETURN-Taste übernommen werden.
3. Drücken Sie die RETURN-Taste.
 - Ist die Syntax fehlerfrei, wird die Adresse schwarz und formatiert dargestellt und der Bausteineditor öffnet von sich aus das nächste Textfeld, in das noch keine Adresse bzw. Parameter eingegeben worden ist.
 - Ist ein Fehler in der Syntax wird das Eingabefeld nicht verlassen und es wird eine entsprechende Fehlermeldung in der Statuszeile angezeigt. Betätigen Sie nochmals die RETURN-Taste, wird das Eingabefeld geschlossen, die falsche Eingabe wird jedoch rot und kursiv dargestellt.

Überschreiben von Adressen oder Parametern in KOP-Elementen

1. Wechseln Sie mit der INSERT-Taste in den Überschreibmodus. Der aktuelle Modus wird in der Statuszeile unten rechts angezeigt.
2. Positionieren Sie mit Mausclick oder mit der TAB-Taste den Mauszeiger auf das Textfeld der Adresse oder des Parameters.
3. Überschreiben Sie die Adresse bzw. den Parameter.
4. Drücken Sie die RETURN-Taste.
 - Ist die Syntax fehlerfrei, wird die Adresse schwarz und formatiert dargestellt und der Bausteineditor öffnet von sich aus das nächste Textfeld, in das noch keine Adresse bzw. Parameter eingegeben worden ist.
 - Ist ein Fehler in der Syntax, wird das Eingabefeld nicht verlassen und es wird eine entsprechende Fehlermeldung in der Statuszeile angezeigt. Betätigen Sie nochmals die RETURN-Taste, wird das Eingabefeld geschlossen, die falsche Eingabe wird jedoch rot und kursiv dargestellt.

Überschreiben von KOP-Elementen

Der Überschreibmodus ermöglicht ein einfaches Austauschen von KOP-Elementen gleichen Typs. Der Vorteil liegt darin, dass Sie die Adressen und Parameter nicht nochmals eingeben müssen. Das zu überschreibende KOP-Element kann nur mit einem KOP-Element gleichen Typs überschrieben werden. So lassen sich z. B. Schließer und Öffner, RS-FlipFlop und SR-FlipFlop, Zeitglieder oder Zähler gegeneinander austauschen.

1. Wechseln Sie mit der INSERT-Taste in den Überschreibmodus. Der aktuelle Modus wird in der Statusleiste unten rechts angezeigt.
2. Markieren Sie das zu überschreibende KOP-Element.
3. Fügen Sie das gewünschte Element ein, indem Sie:
 - das entsprechende Symbol in der Symbolleiste betätigen (für Schließer, Öffner, Spule).
 - die entsprechende Funktionstaste F2, F3 oder F4 (für Schließer, Öffner) betätigen.
 - auf das markierte Element in der Befehlsbibliothek doppelklicken oder es per Drag&Drop in den Bausteineditor ziehen.

Das zuvor markierte KOP-Element wird von dem neu ausgewählten überschrieben.

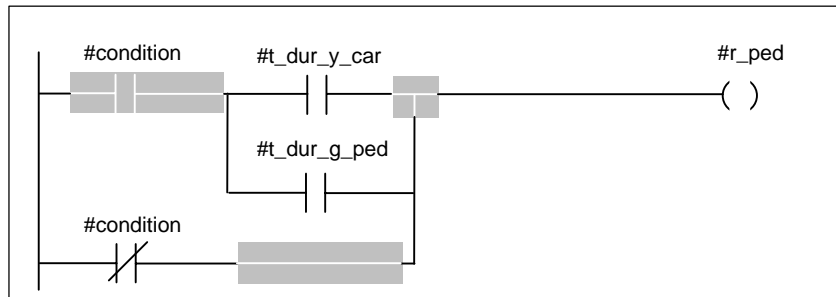
- Wechseln Sie ggf. mit der INSERT-Taste wieder in den Einfügemodus. Der aktuelle Modus wird in der Statusleiste unten rechts angezeigt.

Markieren in KOP-Netzwerken

In ein Netzwerk gelangen Sie durch einen Mausklick auf ein zugehöriges KOP-Element. Innerhalb eines Netzwerkes können Sie prinzipiell drei Bereiche, jeweils durch einen einfachen Mausklick, markieren:

- KOP-Elemente, z. B. einen Kontakt oder eine Box
- Kreuzungspunkte
- Leerelemente (Drahtstück oder offener Zweig)

Sie können gleichzeitig jeweils nur einen Bereich markieren. Nachfolgendes Bild zeigt Beispiele für Markierungen, dabei sind jedoch mehrere Markierungen gleichzeitig dargestellt.



Die Farbe der Markierung können im Dialogfeld "Einstellungen" selbst bestimmen. Sie erhalten dieses Dialogfeld über den Menübefehl **Extras > Einstellungen....**

Einfügen von weiteren KOP-Netzwerken

Um ein neues Netzwerk zu erstellen:

- Wählen Sie den Menübefehl **Einfügen > Netzwerk**.
- Klicken Sie auf das entsprechende Symbol in der Symbolleiste.
- Doppelklicken Sie auf "Neues Netzwerk" in der Befehlsbibliothek oder ziehen es per Drag&Drop in den Bausteineditor.
- Wählen Sie den Menübefehl Netzwerk über das Kontextmenü. Positionieren Sie den Mauszeiger und klicken Sie auf die rechte Maustaste, um das Kontextmenü aufzurufen.

Das neue Netzwerk wird unterhalb des markierten Netzwerkes eingefügt. Es enthält nur einen Zweig.

Wenn Sie mehr Elemente eingeben, als auf Ihrem Bildschirm angezeigt werden können, wird das Netzwerk auf dem Bildschirm nach links verschoben. Mit den Menübefehlen **Ansicht > Verkleinern/Vergrößern/Zoomfaktor...** können Sie die Darstellung anpassen, um sich so einen besseren Überblick zu verschaffen.

In ein Netzwerk gelangen Sie durch einen Mausklick auf ein zugehöriges KOP-Element. Innerhalb eines Netzwerkes können Sie prinzipiell drei Bereiche mit Mausklick markieren.

Erstellen von parallelen Verzweigungen in KOP-Netzwerken

Um in der Programmiersprache KOP ODER-Verknüpfungen zu realisieren, müssen Sie Parallelzweige erstellen.

Gehen Sie folgendermaßen vor:

1. Markieren Sie das Element, vor dem Sie den Parallelzweig öffnen möchten.
2. Öffnen Sie den Parallelzweig, indem Sie:
 - den Kontextmenübefehl **Verzweigung auf** wählen. Positionieren Sie den Mauszeiger und klicken Sie auf die rechte Maustaste, um das Kontextmenü aufzurufen.
 - die Funktionstaste F8 betätigen.
 - das entsprechende Symbol in der Symbolleiste betätigen.
3. Fügen Sie in den offenen Parallelzweig die gewünschten KOP-Elemente ein.
4. Markieren Sie im "Hauptzweig" das KOP-Element, hinter dem Sie die Parallelverzweigung wieder schließen möchten.
5. Schließen Sie den Parallelzweig, indem Sie:
 - den Kontextmenübefehl **Verzweigung zu** wählen. Positionieren Sie den Mauszeiger und klicken Sie auf die rechte Maustaste, um das Kontextmenü aufzurufen.
 - die Funktionstaste F9 betätigen.
 - das entsprechende Symbol in der Symbolleiste betätigen.

Erzeugen von neuen Zweigen in KOP-Netzwerken

Innerhalb eines KOP-Netzwerkes können Sie mehrere Zweige parallel verzweigen:

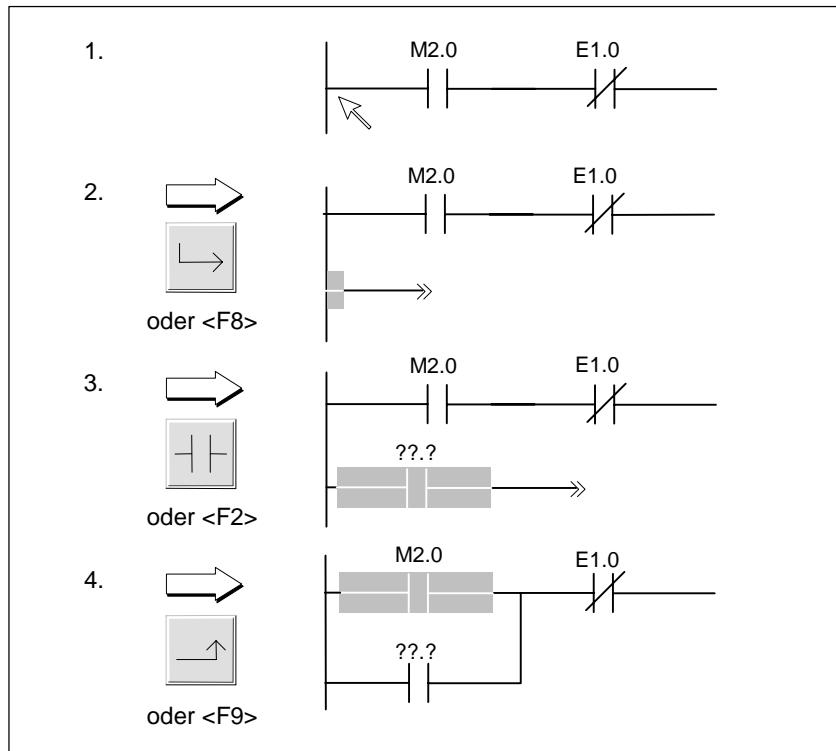
1. Markieren Sie mit dem Mauszeiger den Anfangspunkt des Zweiges, unter dem Sie einen neuen Zweig einfügen möchten.
2. Öffnen Sie den neuen Zweig, indem Sie:
 - den Kontextmenübefehl **Verzweigung auf** wählen. Positionieren Sie den Mauszeiger und klicken Sie auf die rechte Maustaste, um das Kontextmenü aufzurufen.
 - die Funktionstaste F8 betätigen.
 - das entsprechende Symbol in der Symbolleiste betätigen.

Erzeugen einer geschlossenen Verzweigung in KOP-Netzwerken

Um eine geschlossene Verzweigung zu erstellen, gehen Sie folgendermaßen vor:

1. Markieren Sie das Element, vor dem Sie den Parallelzweig öffnen wollen.
2. Öffnen Sie den Parallelzweig mit F8.
3. Fügen Sie KOP-Operation ein.
4. Schließen Sie den Zweig mit F9.

Nachfolgendes Beispiel zeigt, wie Sie eine Verzweigung nur mit Funktionstasten oder Symbolen der Symbolleiste erzeugen.

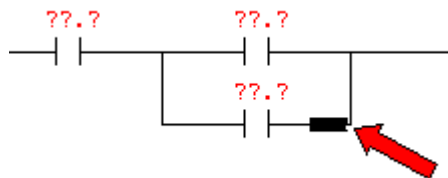


Beim Schließen von Parallelzweigen werden notwendige Leerelemente ergänzt. Wenn erforderlich, werden die Zweige so angeordnet, dass Überkreuzungen von Zweigen vermieden werden. Wenn Sie die Verzweigung direkt aus dem Parallelzweig schließen, wird die Verzweigung hinter dem nächstmöglichen KOP-Element geschlossen.

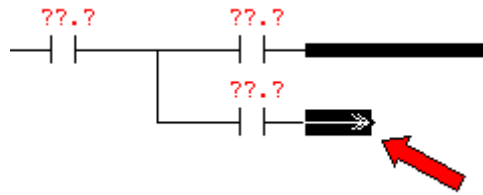
Auftrennen geschlossener Parallelzweige in KOP

Einen geschlossenen Parallelzweig können Sie folgendermaßen auftrennen:

1. Markieren Sie den Parallelzweig an der angegebenen Stelle vor dem Kreuzungspunkt, an dem der Parallelzweig wieder mit dem Hauptzweig zusammentrifft.



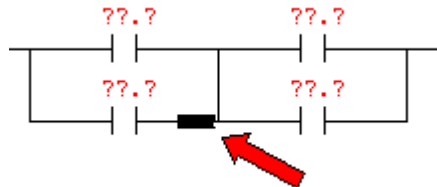
- Drücken Sie die Taste DEL (Entf). An der aufgetrennten Stelle können Sie anschließend neue KOP-Elemente einfügen.



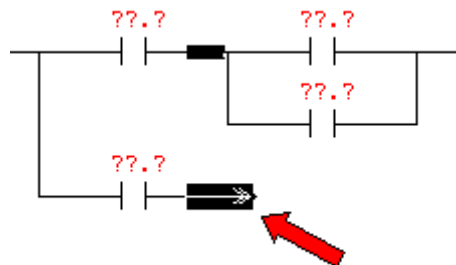
Aufspalten einer Kreuzung in KOP-Netzwerken

Wenn sich an einer Stelle eines KOP Netzwerkes gleichzeitig ein Parallelzweig schließt und sich ein weiterer öffnet, sprechen wir von einer Kreuzung. Eine solche Kreuzung können Sie folgendermaßen aufspalten:

- Markieren Sie die Kreuzung an der angegebenen Stelle im Netzwerk.



- Drücken Sie die Taste DEL (Entf). An der aufgetrennten Stelle können Sie anschließend neue KOP-Elemente einfügen.



- Fügen Sie ein KOP-Element ein, indem Sie
 - das entsprechende Symbol in der Symbolleiste betätigen (für Schließer, Öffner).
 - die entsprechende Funktionstaste F2 oder F3 (für Schließer, Öffner) betätigen.
 - auf das markierte Element in der Befehlsbibliothek doppelklicken oder es per Drag & Drop in den Bausteineditor ziehen.

Erzeugen von Abzweigen in KOP-Netzwerken

Der Kontextmenübefehl **Verzweigung auf** öffnet einen Parallelzweig ohne Spule, beginnend vor dem markierten Programmelement. In den neuen Abzweig können Sie weitere Verknüpfungen einfügen.

1. Markieren Sie mit dem Mauszeiger das Programmelement, vor dem Sie einen neuen Abzweig einfügen möchten.
2. Öffnen Sie den neuen Abzweig folgendermaßen:
 - Wählen Sie den Menübefehl **Verzweigung auf** über das Kontextmenü. Positionieren Sie den Mauszeiger und klicken Sie auf die rechte Maustaste, um das Kontextmenü aufzurufen.
 - Klicken Sie auf das entsprechende Symbol in der Symbolleiste.
 - Betätigen Sie die Funktionstaste F8.
3. Wählen Sie nun ein KOP-Element aus, das in den Abzweig eingefügt werden soll.

6.3.6 Editieren von FUP-Anweisungen im Anweisungsteil

6.3.6.1 Einstellungen für die Programmiersprache FUP

Einstellen des Layout für FUP

Sie können für die Programmerstellung in der Programmiersprache FUP das Layout festlegen. Das ausgewählte Format (DIN A4 Hochformat/Querformat/maximale Größe) wirkt sich auf die Anzahl der in einem Zweig darstellbaren FUP-Elemente aus.

1. Wählen Sie den Menübefehl **Extras > Einstellungen...**
2. Wählen Sie im nachfolgenden Dialogfeld unter Listenfeld "Layout" das gewünschte Format. Geben Sie die gewünschte Formatgröße an.

Hinweise zum Drucken

Wenn Sie den FUP-Anweisungsteil ausdrucken möchten, so sollten Sie bereits vor der Erstellung des Anweisungsteils das geeignete Seitenformat einstellen.

Grundeinstellungen unter **Extras > Einstellungen...**

Sie können über **Extras > Einstellungen...** Grundeinstellungen vornehmen, z. B. zum Layout und zur Operandenfeldbreite.

6.3.6.2 Regeln für die Eingabe von FUP-Anweisungen

Die Beschreibung der Sprache "FUP" finden Sie im Handbuch *FUP für S7-300/400 – Bausteine programmieren* oder in der Online-Hilfe zu FUP.

Ein FUP-Netzwerk kann aus mehreren Elementen bestehen. Alle Elemente müssen untereinander verbunden sein (IEC 1131–3).

Bei der Programmierung in FUP müssen Sie einige Regeln beachten. Auf eventuelle Fehler werden Sie durch eine Fehlermeldung hingewiesen.

Einfügen und Editieren von Adressen und Parametern

Beim Einfügen eines FUP-Elements werden die Zeichenfolgen "???" und "..." als Platzhalter für Adressen bzw. Parameter eingesetzt.

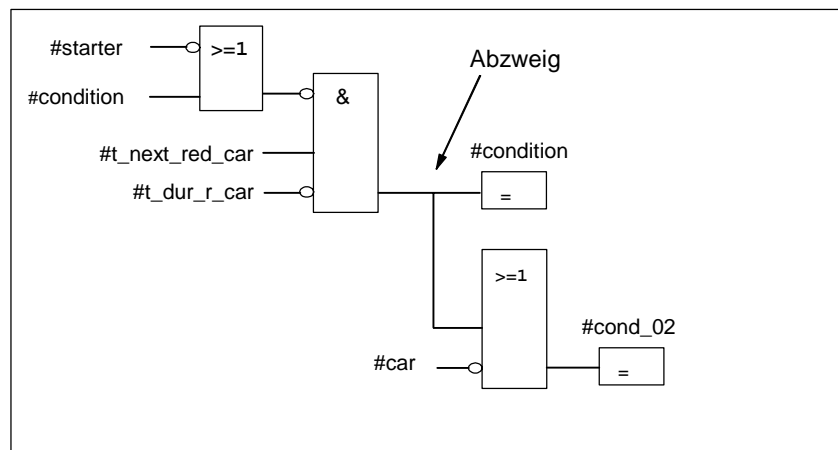
- Die rot dargestellte Zeichenfolge "???" kennzeichnet Adressen und Parameter, die beschaltet werden müssen.
- Die schwarz dargestellte Zeichenfolge "..." kennzeichnet Adressen und Parameter, die beschaltet werden können.

Wenn Sie den Mauszeiger über den Platzhalter bewegen, wird der erwartete Datentyp angezeigt.

Platzierung von Boxen

An Boxen mit binären Verknüpfungen (&, >=1, XOR) können Standard-Boxen (Flipflops, Zähler, Zeiten, Rechenoperationen usw.) angefügt werden. Ausgenommen von dieser Regelung sind Vergleicherverboxen.

In einem Netzwerk dürfen keine voneinander getrennten Verknüpfungen mit separaten Ausgängen programmiert werden. Sie können jedoch mit Hilfe des Abzweigs einer Verknüpfungskette mehrere Zuweisungen zuordnen. Nachfolgendes Bild zeigt ein Netzwerk mit zwei Zuweisungen.



Die folgenden Boxen dürfen nur am rechten Rand der Verknüpfungskette platziert werden, wo sie den Abschluss der Kette bilden:

- Zähleranfangswert setzen
- Vorwärtszählen, Rückwärtszählen
- Zeit als Impuls starten, Zeit als verlängerten Impuls starten
- Zeit als Einschaltverzögerung/Ausschaltverzögerung starten

Unter den Boxen gibt es solche, die eine boolesche Verknüpfung verlangen, und solche, die keine boolesche Verknüpfung haben dürfen.

Boxen mit erforderlicher boolescher Verknüpfung:

- Ausgang, Ausgang setzen, Ausgang rücksetzen `_[R]`
- Konnektor `_[#]_`, positive Flanke `_[P]_`, negative Flanke `_[N]_`
- alle Zähler- und Timerboxen
- Springe wenn 0 `_[JMPN]`
- Master Control Relais einschalten `_[MCR<]`
- Verknüpfungsergebnis ins BIE-Register laden `_[SAVE]`
- Springe zurück `_[RET]`

Boxen, die keine boolesche Verknüpfung erlauben:

- Master Control Relais Anfang [MCRA]
- Master Control Relais Ende [MCRD]
- Datenblock öffnen [OPN]
- Master Control Relais ausschalten [MCR>]

Alle anderen Boxen können wahlweise eine boolesche Verknüpfung haben oder auch nicht.

Freigabeeingang/-Freigabeausgang

Die Beschaltung des Freigabeeingangs "EN" bzw. des Freigabeausgangs "ENO" von Boxen kann erfolgen, ist aber nicht zwingend erforderlich.

Entfernen und Ändern

Mit dem Entfernen einer Box werden mit Ausnahme des Hauptzweiges auch alle Zweige, die mit booleschen Eingängen der Box verbunden sind, entfernt.

Zum einfachen Austauschen von Elementen gleichen Typs steht Ihnen der Überschreibmodus zur Verfügung.

6.3.6.3 Schritte zum Eingeben von FUP-Elementen

Eingeben von FUP-Elementen

Gehen Sie folgendermaßen vor:

1. Markieren Sie die gewünschte Stelle in einem Netzwerk, hinter der ein FUP-Element eingefügt werden soll.
2. Fügen Sie ein FUP-Element ein, indem Sie:
 - das entsprechende Symbol in der Symbolleiste betätigen (für ODER-Box, UND-Box).
 - die entsprechende Funktionstaste F2 oder F3 (für ODER-Box, UND-Box) betätigen.
 - auf das markierte Element in der Befehlsbibliothek doppelklicken oder es per Drag&Drop in den Bausteineditor ziehen.

Das ausgewählte FUP-Element wird eingefügt, wobei für die Adressen und Parameter der Platzhalter ??? eingefügt wird.

Hinweis

Sie können den Anweisungsteil auch durch Markieren vorhandener FUP-Elemente und anschließendem Auslösen der Menübefehle **Bearbeiten > Ausschneiden**, **Bearbeiten > Kopieren** oder **Bearbeiten > Einfügen** bearbeiten.

Eingeben von Adressen oder Parametern

1. Positionieren Sie mit Mausclick oder mit der TAB-Taste den Mauszeiger auf den Platzhalter.
2. Geben Sie die Adresse bzw. den Parameter in den Platzhalter ein (direkte oder indirekte Adressierung). Falls die Anzeige der Symbolauswahl eingeschaltet ist (Menübefehl **Ansicht > Anzeigen mit > Symbolauswahl**), wird eine Liste der bekannten Symbole angezeigt. Das mit den eingegebenen Zeichen beginnende Symbol ist markiert und kann durch Drücken der RETURN-Taste übernommen werden.
3. Drücken Sie die RETURN-Taste.
 - Ist die Syntax fehlerfrei, wird die Adresse schwarz und formatiert dargestellt und der Bausteineditor öffnet von sich aus das nächste Textfeld, in das noch keine Adresse bzw. noch kein Parameter eingegeben worden ist.
 - Ist ein Fehler in der Syntax, wird das Eingabefeld nicht verlassen und es wird eine entsprechende Fehlermeldung in der Statusleiste angezeigt. Betätigen Sie nochmals die RETURN-Taste, wird das Eingabefeld geschlossen, die falsche Eingabe wird jedoch rot und kursiv dargestellt.

Hinweis

Die Zeichenfolge ">>" an einem Ausgang bedeutet, dass dieser Ausgang vor dem Abspeichern bzw. Laden noch beschaltet werden muss.

Überschreiben von FUP-Elementen

Der Überschreibmodus ermöglicht ein einfaches Austauschen von FUP-Elementen gleichen Typs. Der Vorteil liegt darin, dass Sie die Adressen und Parameter nicht nochmals eingeben müssen. Das zu überschreibende FUP-Element kann nur mit einem FUP-Element gleichen Typs überschrieben werden. Es können auf diese Weise z. B. UND-Box und ODER-Box, RS-FlipFlop und SR-FlipFlop, Zeitglieder oder Zähler gegeneinander ausgetauscht werden.

Gehen Sie folgendermaßen vor:

1. Wechseln Sie mit der INSERT-Taste in den Überschreibmodus. Der aktuelle Modus wird in der Statusleiste unten rechts angezeigt.
2. Markieren Sie das zu überschreibende FUP-Element.
3. Fügen Sie das gewünschte Element ein, indem Sie:
 - das entsprechende Symbol in der Symbolleiste betätigen.
 - die entsprechende Funktionstaste F2 der F3 (für ODER-Box, UND-Box) betätigen.
 - auf das markierte Element in der Befehlsbibliothek doppelklicken oder es per Drag&Drop in den Bausteineditor ziehen.

Wechseln Sie ggf. mit der INSERT-Taste wieder in den Einfügemodus. Der aktuelle Modus wird in der Statusleiste unten rechts angezeigt.

Markieren in FUP-Netzwerken

Innerhalb eines Netzwerks können Sie folgende Bereiche, jeweils durch einen einfachen Mausklick, markieren:

- FUP-Elemente, z. B. eine UND-Box oder eine Standard-Box, z. B. Zähler,
- Verbindungslinien
- Operanden
- Ein- / Ausgangskontakte

Die Farbe der Markierung können Sie im Dialogfeld "Einstellungen" selbst bestimmen. Sie erhalten dieses Dialogfeld über den Menübefehl **Extras > Einstellungen....**

Um ein Netzwerk zu markieren, in das Sie FUP-Elemente eingeben können, gehen Sie folgendermaßen vor:

1. Klicken Sie auf die Netzwerkbezeichnung (z. B. "Netzwerk 1").
2. Das so markierte Netzwerk können Sie z. B. ausschneiden, wieder einfügen oder kopieren.

Einfügen von weiteren FUP-Netzwerken

Um ein neues Netzwerk zu erstellen:

- Wählen Sie den Menübefehl **Einfügen > Netzwerk**.
- Klicken Sie auf das entsprechende Symbol in der Symbolleiste.
- Doppelklicken Sie auf "Neues Netzwerk" in der Befehlsbibliothek oder ziehen es per Drag&Drop in den Bausteineditor.
- Wählen Sie den Menübefehl Netzwerk über das Kontextmenü.
Positionieren Sie den Mauszeiger und klicken Sie auf die rechte Maustaste, um das Kontextmenü aufzurufen.

Das neue Netzwerk wird unterhalb des markierten Netzwerks eingefügt.

Wenn Sie mehr Elemente eingeben, als auf Ihrem Bildschirm angezeigt werden können, wird das Netzwerk auf dem Bildschirm nach links verschoben. Mit den Menübefehlen **Ansicht > Verkleinern/Vergrößern/Zoomfaktor...** können Sie die Darstellung anpassen, um sich so einen besseren Überblick zu verschaffen.

Um ein Netzwerk zu markieren, klicken Sie auf die Netzwerkbezeichnung (z. B. "Netzwerk 1"). Das so markierte Netzwerk können Sie z. B. ausschneiden, wieder einfügen oder kopieren.

Erzeugen von Abzweigen in FUP-Netzwerken

Innerhalb eines FUP-Netzwerks können Sie mehrere Abzweige programmieren. Der Abzweig öffnet einen Parallelzweig, beginnend vor dem markierten binären Eingang. In den neuen Abzweig können Sie weitere Verknüpfungen einfügen:

1. Markieren Sie mit dem Mauszeiger den binären Eingang, an dem Sie einen neuen Abzweig einfügen möchten.
2. Öffnen Sie den neuen Abzweig, indem Sie
 - den Menübefehl **Abzweig** über das Kontextmenü wählen.
Positionieren Sie den Mauszeiger und klicken Sie auf die rechte Maustaste, um das Kontextmenü aufzurufen.
 - die Funktionstaste F11 betätigen.
 - das entsprechende Symbol in der Symbolleiste betätigen.

Erzeugen von Verbindungen in FUP-Netzwerken

Innerhalb eines FUP-Netzwerks können Sie zwei Verknüpfungspfade verbinden, wobei nur ein Verknüpfungspfad eine Zuweisung enthalten darf.

Gehen Sie folgendermaßen vor:

1. Markieren Sie mit dem Mauszeiger den binären Eingang und den binären Ausgang, welche Sie verbinden möchten.
2. Sie verbinden die binären Objekte folgendermaßen:
 - Wählen Sie den Menübefehl **Verbindung** über das Kontextmenü.
Positionieren Sie den Mauszeiger und klicken Sie auf die rechte Maustaste, um das Kontextmenü aufzurufen.
 - Betätigen Sie die Funktionstaste F12.
 - Klicken Sie auf das entsprechende Symbol in der Symbolleiste.

Verbindung trennen und neu verbinden in FUP-Netzwerken

1. Markieren Sie einen binären Eingang.
2. Trennen Sie die Verbindung durch Drücken der DEL-Taste.
3. Fügen Sie bei Bedarf an der Trennstelle neue FUP-Elemente ein.
4. Markieren Sie den binären Ausgang.
5. Halten Sie die Maustaste gedrückt und ziehen Sie eine Verbindung zu dem gewünschten binären Eingang.

Gegebenenfalls werden die Elemente grafisch neu angeordnet.

6.3.7 Editieren von AWL-Anweisungen im Anweisungsteil

6.3.7.1 Einstellungen für die Programmiersprache AWL

Einstellen der Mnemonik

Sie können zwischen zwei Einstellungen der Mnemonik wählen:

- Deutsch oder
- Englisch

Die Mnemonik stellen Sie vor dem Öffnen eines Bausteins im Dialogfeld **Extras > Einstellungen...** ein. Während der Bausteinbearbeitung kann die Mnemonik nicht geändert werden.

Die Bausteineigenschaften können Sie über das Register "Eigenschaften" verändern.

Im Bausteineditor können Sie mehrere Bausteine geöffnet haben und nach Wunsch abwechselnd bearbeiten.

6.3.7.2 Regeln für die Eingabe von AWL-Anweisungen

Die Beschreibung der Sprache "AWL" finden Sie im Handbuch *AWL für S7-300/400 – Bausteine programmieren* oder in der Online-Hilfe zu AWL (Sprachbeschreibungen)

Bei der inkrementellen Eingabe von AWL-Anweisungen müssen Sie folgende Grundregeln beachten:

- Achten Sie auf die Reihenfolge der Programmierung der Bausteine. Aufgerufene Bausteine müssen vor den aufrufenden Bausteinen programmiert werden.
- Eine Anweisung besteht aus der Angabe von Sprungmarke (optional), Operation, Operand und Kommentar (optional).
Beispiel: M001: U E1.0 //Kommentar
- Jede Anweisung steht in einer eigenen Zeile.
- Sie können je Baustein maximal 999 Netzwerke eingeben.
- Je Netzwerk können Sie etwa 2000 Zeilen eingeben. Bei vergrößerter oder verkleinerter Ansicht sind entsprechend mehr oder weniger Zeilen möglich.
- Bei der Eingabe von Operationen oder absoluten Adressen wird nicht zwischen Groß- und Kleinschreibung unterschieden.

6.3.7.3 Schritte zum Eingeben von AWL-Anweisungen

Eingeben von AWL-Anweisungen

Wenn Sie einen neuen Codebaustein erstellen, können Sie das erste Netzwerk sofort bearbeiten. In ein Netzwerk gelangen Sie durch einen Mausklick auf eine zugehörige Zeile. Die Anweisungen innerhalb der einzelnen Netzwerke geben Sie einfach Zeile für Zeile über die Tastatur ein. Zum Editieren stehen Ihnen alle üblichen Bearbeitungsfunktionen zur Verfügung.

1. Öffnen Sie das Textfeld für das Netzwerk, indem Sie auf die freie Fläche unterhalb des grünen Kommentarfelds klicken (bzw. unterhalb der Netzwerküberschrift, wenn die Anzeige der Kommentare ausgeschaltet ist).
2. Geben Sie die Operation ein und nach Betätigung der Leertaste die Adresse (direkte oder indirekte Adressierung).
3. Drücken Sie die Leertaste und geben Sie, beginnend mit zwei Schrägstrichen // den Anweisungskommentar ein (optional).
4. Schließen Sie die Eingabe der Anweisung - mit oder ohne //Kommentar - durch die RETURN-Taste ab.

Mit Abschluss der Eingabe einer Zeile wird eine Syntaxprüfung durchgeführt und die Anweisung formatiert ausgegeben. Kleinbuchstaben in der Operation oder absoluten Adresse werden dabei in Großbuchstaben gewandelt.

Syntaxfehler werden rot angezeigt. Sie müssen vor dem Übernehmen bzw. Speichern des Codebausteins behoben werden.

Markieren von Textbereichen in AWL-Anweisungen

So können Sie innerhalb eines AWL-Netzwerks zeichenorientiert den Text markieren:

1. Positionieren Sie die Einfügemarke auf das erste Zeichen.
2. Markieren Sie den Text, indem Sie bei gedrückter linker Maustaste über den zu markierenden Text ziehen.

Sie können mehrere Anweisungszeilen gleichzeitig markieren, indem Sie die Maus bei gedrückter Maustaste vertikal bewegen. Alternativ können Sie Textbereiche bei gedrückter SHIFT-Taste durch die Richtungspfeiltasten rechts, links oder auf- und abwärts markieren.

Hinweis

Die Farbe der Markierungen können Sie selbst festlegen. Öffnen Sie dazu mit dem Menübefehl **Extras > Einstellungen...** das Dialogfeld und bestimmen Sie die Farbe für "Angewähltes Element".

Einfügen von weiteren AWL-Netzwerken

Um ein neues Netzwerk zu erstellen:

- Wählen Sie den Menübefehl **Einfügen > Netzwerk**.
- Klicken Sie auf das entsprechende Symbol in der Symbolleiste.
- Doppelklicken Sie auf "Neues Netzwerk" in der Befehlsbibliothek oder ziehen es per Drag&Drop in den Bausteineditor.
- Wählen Sie den Menübefehl Netzwerk über das Kontextmenü.
Positionieren Sie den Mauszeiger und klicken Sie auf die rechte Maustaste, um das Kontextmenü aufzurufen.

Das neue Netzwerk wird unterhalb des markierten Netzwerks eingefügt.

Wenn Sie mehr Elemente eingeben, als auf Ihrem Bildschirm angezeigt werden können, wird das Netzwerk auf dem Bildschirm nach links verschoben. Mit den Menübefehlen **Ansicht > Verkleinern/Vergrößern/Zoomfaktor** können Sie die Darstellung anpassen, um sich so einen besseren Überblick zu verschaffen.

Um ein Netzwerk zu markieren, klicken Sie auf die Netzwerkbezeichnung (z. B. "Netzwerk 1"). Das so markierte Netzwerk können Sie z. B. ausschneiden, wieder einfügen oder kopieren.

Eingeben von Kommentaren in AWL-Anweisung

In der Programmiersprache AWL können Sie für jede Anweisung einen Kommentar eingeben.

1. Drücken Sie nach Eingabe der Adresse bzw. des Namens die Leertaste.
2. Beginnen Sie Ihren Anweisungskommentar mit zwei Schrägstrichen (//).
3. Schließen Sie die Kommentareingabe mit RETURN ab.

6.3.8 Aktualisieren von Bausteinaufrufen

Mit dem Menübefehl **Bearbeiten > Aufruf > Aktualisieren** im Bausteineditor können Sie ungültig gewordene Bausteinaufrufe oder UDT-Verwendungen nach folgenden Schnittstellenänderungen automatisch abgleichen:

- Einfügen neuer Formalparameter
- Löschen von Formalparametern
- Namensänderung von Formalparametern
- Typänderung von Formalparametern
- Änderung der Reihenfolge (Umkopieren) von Formalparametern

Bei der Zuordnung von Formal- und Aktualseite wird nach folgenden Regeln in der angegebenen Reihenfolge vorgegangen:

1. **Gleiche Parameternamen:**

Die Aktualparameter werden automatisch zugeordnet, wenn der Name des Formalparameters gleich geblieben ist.

Sonderfall: Die Vorverknüpfung von binären Eingangsparametern kann in KOP und FUP nur bei gleichem Datentyp (BOOL) automatisch zugeordnet werden.

Wurde der Datentyp in einem solchen Fall geändert, so bleibt die Vorverknüpfung als offener Zweig bestehen.

2. **Gleiche Parameterdatentypen:**

Nachdem die Parameter mit gleichen Namen zugeordnet wurden, werden nun noch nicht zugeordnete Aktualparameter zu Formalparametern mit gleichem Datentyp wie der "alte" Formalparameter zugeordnet.

3. **Gleiche Parameterposition:**

Die nach der Zuordnung gemäß Regel 1 und 2 noch nicht zugeordneten Aktualparameter werden nun gemäß ihrer Parameterposition in der "alten" Schnittstelle den neuen Formalseiten zugeordnet.

4. Können Aktualparameter nach den drei genannten Regeln nicht zugeordnet werden, so werden diese gelöscht bzw. bleiben im Falle von binären Vorverknüpfungen in KOP oder FUP als offene Zweige erhalten.

Überprüfen Sie nach Ausführung dieser Funktion die vorgenommenen Änderungen in der Variablendeklarationstabelle und im Anweisungsteil des Programms.

6.4 Erstellen von Datenbausteinen

6.4.1 Grundlagen zum Erstellen von Datenbausteinen

Der Datenbaustein ist ein Baustein, in dem Sie Daten hinterlegen, auf die Ihre Maschine oder Anlage zurückgreift. Im Gegensatz zum Codebaustein, der mit einer der Programmiersprachen KOP/FUP oder AWL programmiert wurde, enthält der Datenbaustein nur die Variablendeklarationstabelle. Der Anweisungsteil entfällt und damit auch das Programmieren von Netzwerken.

Datenbaustein erzeugen

1. Wählen Sie den Menübefehl **Datei > Neu > Baustein**.
2. Wählen Sie im Folgedialog als Bausteintyp "Datenbaustein" und geben Sie die Nummer ein.

Nach dem Öffnen eines Datenbausteins können Sie sich den Baustein entweder in der Deklarationssicht oder in der Datensicht anzeigen lassen. Umschalten können Sie mit den Menübefehlen **Ansicht > Deklarationssicht** und **Ansicht > Datensicht**.

Deklarationssicht

Wählen Sie die Deklarationssicht, um

- die Datenstruktur globaler DBs abzulesen oder festzulegen,
- die Datenstruktur von DBs mit zugeordnetem anwenderdefinierten Datentyp (UDT) abzulesen,
- die Datenstruktur von DBs mit zugeordnetem Funktionsbaustein abzulesen.

Die Struktur von Datenbausteinen, die einem FB oder einem anwenderdefinierten Datentyp zugeordnet sind, kann nicht geändert werden. Dazu müsste zunächst der entsprechende FB bzw. UDT geändert und anschließend der DB neu angelegt werden.

Datensicht

Wählen Sie die Datensicht, um Daten zu ändern. Nur in der Datensicht können Sie sich für jedes Element den aktuellen Wert anzeigen lassen, bzw. ihn eingeben oder ändern. In der Datensicht von Datenbausteinen werden bei Variablen mit zusammengesetzten Datentypen die Elemente einzeln mit vollständigem Namen aufgelistet.

Unterschied zwischen Instanz-Datenbaustein und globalen Datenbaustein

Der globale Datenbaustein ist keinem Codebaustein zugeordnet. Er dient als Datenablage für die von der CPU gesteuerte Anlage oder Maschine und lässt sich an jeder Stelle des Programms aufrufen und bearbeiten.

Der Instanz-Datenbaustein ist ein Baustein, der einem Codebaustein, z. B. einem Funktionsbaustein, unmittelbar zugeordnet ist. Der Instanz-Datenbaustein enthält die Daten, die in einem Funktionsbaustein in der Variablendeklarationstabelle hinterlegt wurden.

6.4.2 Deklarations­sicht von Datenbausteinen

Bei nicht globalen Datenbausteinen kann die Deklarations­sicht nicht verändert werden.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 Lite für die Variable automatisch zuordnet, wenn Sie die Eingabe einer Deklaration beenden.
Deklaration	Diese Spalte wird nur für Instanz-Datenbausteine dargestellt. Dieser Spalte können Sie entnehmen, wie die Variablen in der Variablendeklaration des FBs vereinbart worden sind: <ul style="list-style-type: none"> • Eingangparameter ("in") • Ausgangparameter ("out") • Durchgangparameter ("in_out") • statische Daten ("stat")
Name	Geben Sie hier den Namen ein, den Sie jeder Variablen zuweisen müssen.
Typ	Geben Sie hier den Datentyp der Variablen ein (BOOL, INT, WORD, ARRAY, usw.). Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben.
Anfangswert	Geben Sie hier den Anfangswert ein, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen soll. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert für die Variable festlegen.
Kommentar	In diesem Feld können Sie einen Kommentar zur Dokumentation der Variablen eingeben. Der Kommentar darf 80 Zeichen lang sein.

6.4.3 Datensicht von Datenbausteinen

Die Datensicht zeigt Ihnen die aktuellen Werte aller Variablen des Datenbausteins. Sie können diese Werte nur in der Datensicht ändern. Die Tabellendarstellung dieser Sicht ist für alle Global-Datenbausteine gleich. Für Instanz-Datenbausteine wird zusätzlich die Spalte "Deklaration" angezeigt.

Für Variablen mit zusammengesetzten Datentypen oder anwenderdefinierten Datentypen werden in der Datensicht alle Elemente einzeln in einer eigenen Zeile mit vollständigem Namen dargestellt. Wenn die Elemente im In_out-Bereich eines Instanz-Datenbausteins liegen, wird in der Spalte "Aktualwert" der Zeiger auf den zusammengesetzten oder anwenderdefinierten Datentyp gesetzt.

Die Datensicht zeigt folgende Spalten:

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 Lite für die Variable automatisch vergibt.
Deklaration	Diese Spalte wird nur für Instanz-DBs dargestellt. Dieser Spalte können Sie entnehmen, wie die Variablen in der Variablendeklaration des FBs vereinbart worden sind: <ul style="list-style-type: none"> • Eingangparameter("in") • Ausgangsparameter ("out") • Durchgangsparameter ("in_out") • statische Daten ("stat")
Name	Hierbei handelt es sich um den Namen, der für die Variable festgelegt ist. Dieses Feld können Sie in der Datensicht nicht bearbeiten.
Typ	Hierbei handelt es sich um den Datentyp, der für die Variable festgelegt ist. Bei einem globalen Datenbaustein stehen hier nur noch elementare Datentypen, da in der Datensicht für Variablen mit zusammengesetzten oder anwenderdefinierten Datentypen die Elemente einzeln aufgelistet werden. Bei einem Instanz-Datenbaustein werden zusätzlich Parameterdatentypen angezeigt, bei Durchgangsparametern ("in_out") mit zusammengesetztem oder anwenderdefiniertem Datentyp wird in der Spalte "Aktualwert" ein Zeiger auf den Datentyp gesetzt.
Anfangswert	Hierbei handelt es sich um den Anfangswert, den Sie für die Variable festgelegt haben, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen sollte. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert der Variable festlegen.
Aktualwert	Offline: Hierbei handelt es sich um den Wert, den die Variable beim Öffnen des Datenbausteins bzw. nach Ihrer letzten gespeicherten Änderung hatte (auch wenn Sie den DB online geöffnet haben, wird diese Anzeige nicht aktualisiert!). Online: Der aktuelle Wert beim Öffnen des Datenbausteins wird angezeigt, jedoch nicht automatisch aktualisiert. Zum Aktualisieren der Anzeige drücken Sie die Taste F5. Dieses Feld können Sie editieren, wenn es nicht zu einem Durchgangsparameter ("in_out") mit einem zusammengesetzten oder anwenderdefinierten Datentyp gehört. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein.
Kommentar	Hierbei handelt es sich um den Kommentar, der in der zur Dokumentation der Variable vergeben wurde. Dieses Feld können Sie in der Datensicht nicht bearbeiten.

6.4.4 Eingeben in Datenbausteine und Speichern

6.4.4.1 Eingeben der Datenstruktur von globalen Datenbausteinen

Haben Sie einen Datenbaustein geöffnet, der keinem UDT oder FB zugeordnet ist, so können Sie in der Deklarationssicht des Datenbausteins seine Struktur festlegen. Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

1. Öffnen Sie einen globalen Datenbaustein, d. h. einen Baustein, der keinem UDT oder FB zugeordnet ist. Erkennbar ist ein globaler Datenbaustein an der Erstsprache "DB" (siehe Ansicht "Eigenschaften").
2. Wechseln Sie mit dem Menübefehl **Ansicht > Deklarationssicht** in die Deklarationssicht des Datenbausteins, falls diese nicht bereits angezeigt wird.
3. Legen Sie die Struktur fest, indem Sie die angezeigte Tabelle anhand der nachstehenden Angaben ausfüllen.

Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 Lite für die Variable automatisch zuordnet, wenn Sie die Eingabe einer Deklaration beenden.
Name	Geben Sie hier den Namen der Variablen ein.
Typ	Geben Sie hier den Datentyp der Variablen ein (BOOL, INT, WORD, ARRAY, usw.), oder wählen Sie den Datentyp aus dem Kontextmenü (Klick mit der rechten Maustaste) aus. Die Variablen können vom Typ elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen sein.
Anfangswert	Geben Sie hier den Anfangswert ein, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen soll. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein. Der Anfangswert wird beim erstmaligen Übernehmen oder Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert für die Variable festlegen.
Kommentar	In diesem Feld können Sie einen Kommentar zur Dokumentation der Variable eingeben. Der Kommentar darf 80 Zeichen lang sein.

6.4.4.2 Eingeben/Anzeige der Datenstruktur von Datenbausteinen mit zugeordnetem FB (Instanz-DBs)

Eingabe

Wenn Sie einen Datenbaustein einem FB zuordnen (Instanz-DB), dann definiert die Variablendeklaration des FBs die Struktur des DBs. Änderungen sind nur im zugeordneten FB möglich:

1. Öffnen Sie den zugeordneten Funktionsbaustein.
2. Bearbeiten Sie die Variablendeklarationstabelle des Funktionsbausteins.
3. Legen Sie den Instanz-Datenbaustein erneut an!

Anzeige

Sie können sich in der Deklarationssicht des Instanz-DBs anzeigen lassen, wie die Variablen im FB deklariert wurden.

1. Öffnen Sie den Datenbaustein.
2. Lassen Sie sich die Deklarationssicht des Datenbausteins anzeigen, falls diese nicht bereits angezeigt wird.
3. Erläuterungen zur angezeigten Tabelle finden Sie im Anschluss.

Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 Lite für die Variable automatisch vergibt.
Deklaration	Diese Spalte zeigt an, wie die Variablen in der Variablendeklaration des FBs vereinbart worden sind: <ul style="list-style-type: none"> • Eingangsparameter ("in") • Ausgangsparameter ("out") • Durchgangsparameter ("in_out") • statische Daten ("stat") Die deklarierten temporären Daten des FBs sind nicht im Instanz-Datenbaustein.
Name:	Anzeige des Namens, der durch die Variablendeklaration des FBs für die Variable festgelegt ist.
Typ	Anzeige des Datentyps, der durch die Variablendeklaration des FBs für die Variable festgelegt ist. Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben. Werden innerhalb des FBs weitere Funktionsbausteine aufgerufen, für deren Aufruf statische Variablen vereinbart worden sind, dann kann hier auch ein FB oder Systemfunktionsbaustein (SFB) als Datentyp angegeben sein.
Anfangswert:	Hierbei handelt es sich um den Vorbelegungswert, den Sie in der Variablendeklaration des FB für die Variable festgelegt haben, wenn die Software nicht den Defaultwert des Datentyps übernehmen sollte. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert der Variable festlegen.
Kommentar	Hierbei handelt es sich um den Kommentar, der in der Variablendeklaration des FBs zur Dokumentation des Datenelements vergeben wurde. Dieses Feld können Sie nicht bearbeiten.

Hinweis

Bei Datenbausteinen, die einem FB zugeordnet sind, können Sie nur die aktuellen Werte für die Variablen bearbeiten. Das Eingeben von aktuellen Werten für die Variablen erfolgt in der Datensicht von Datenbausteinen.

6.4.4.3 Eingeben der Struktur von anwenderdefinierten Datentypen (UDT)

1. Öffnen Sie den anwenderdefinierten Datentyp (UDT).
2. Lassen Sie sich die Deklarationsansicht anzeigen, falls diese nicht bereits angezeigt wird.
3. Legen Sie die Struktur des anwenderdefinierten Datentyps fest, indem Sie die Reihenfolge der Variablen, ihren Datentyp und ggf. einen Anfangswert anhand der nachstehenden Angaben festlegen.
4. Sie beenden die Eingabe einer Variablen, indem Sie die Zeile mit der TAB- oder RETURN-Taste verlassen.

Spalte	Erläuterung
Adresse:	Anzeige der Adresse, die STEP 7 Lite für die Variable automatisch zuordnet, wenn Sie die Eingabe einer Deklaration beenden.
Name	Geben Sie hier den Namen ein, den Sie jeder Variablen zuweisen müssen.
Typ	Geben Sie hier den Datentyp der Variablen ein (BOOL, INT, WORD, ARRAY, usw.). Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder wiederum anwenderdefinierte Datentypen haben.
Anfangswert	Geben Sie hier den Anfangswert ein, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen soll. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein. Der Anfangswert wird beim erstmaligen Übernehmen bzw. Speichern einer Instanz des UDTs (einer Variablen bzw. eines Datenbausteins) für die Variable als aktueller Wert übernommen, sofern Sie nicht explizit einen anderen aktuellen Wert vorgeben.
Kommentar	In diesem Feld können Sie einen Kommentar zur Dokumentation der Variablen eingeben. Der Kommentar darf 80 Zeichen lang sein.

6.4.4.4 Eingeben/Anzeige der Struktur von Datenbausteinen mit zugeordnetem UDT

Eingabe

Wenn Sie einen Datenbaustein einem UDT zuordnen, dann definiert die Datenstruktur des UDT die Struktur des DB. Änderungen sind nur im zugeordneten UDT möglich:

1. Öffnen Sie den anwenderdefinierten UDT.
2. Bearbeiten Sie die Struktur des anwenderdefinierten Datentyps.
3. Legen Sie den Datenbaustein neu an!

Anzeige

Sie können sich in der Deklarationssicht des DBs nur anzeigen lassen, wie die Variablen im UDT deklariert wurden.

1. Öffnen Sie den Datenbaustein.
2. Lassen Sie sich die Deklarationssicht des Datenbausteins anzeigen, falls diese nicht bereits angezeigt wird.
3. Erläuterungen zur angezeigten Tabelle finden Sie im Anschluss.

Die Deklarationssicht kann nicht verändert werden. Änderungen sind nur im zugeordneten UDT möglich.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 Lite für die Variable automatisch vergibt.
Name	Anzeige des Namens, der durch den UDT für die Variable festgelegt ist.
Typ:	Anzeige der im UDT festgelegten Datentypen. Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben.
Anfangswert	Anzeige des Vorbelegungswerts, den Sie im UDT für die Variable festgelegt haben, wenn die Software nicht den Defaultwert des Datentyps übernehmen sollte. Dieser Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert der Variable festlegen.
Kommentar	Hierbei handelt es sich um den Kommentar, der durch den UDT zur Dokumentation des Datenelements festgelegt ist.

Hinweis

Bei Datenbausteinen, die einem UDT zugeordnet sind, können Sie nur die aktuellen Werte für die Variablen bearbeiten. Das Eingeben von aktuellen Werten für die Variablen erfolgt in der Datensicht von Datenbausteinen.

6.4.4.5 Ändern von Datenwerten in der Datensicht

Das Bearbeiten von aktuellen Werten ist nur in der Datensicht von Datenbausteinen möglich.

1. Schalten Sie ggf. mit dem Menübefehl **Ansicht > Datensicht** die Tabellendarstellung in die Datensicht um.
2. Geben Sie die gewünschten aktuellen Werte für die Datenelemente in den Feldern der Spalte "Aktualwert" ein. Die Aktualwerte müssen kompatibel zum Datentyp der Datenelemente sein.

Fehlerhafte Eingaben bei der Bearbeitung werden sofort erkannt und rot dargestellt (z. B. wenn ein eingegebener aktueller Wert nicht kompatibel zum Datentyp ist). Diese Fehler müssen vor dem Speichern behoben sein.

Achtung

Änderungen der Datenwerte werden erst beim Speichern von Datenbausteinen gesichert.

6.4.4.6 Rücksetzen von Datenwerten auf die Anfangswerte

Das Rücksetzen ist nur in der Datensicht von Datenbausteinen möglich.

1. Schalten Sie ggf. mit dem Menübefehl **Ansicht > Datensicht** die Tabellendarstellung in die Datensicht um.
2. Wählen Sie hierfür den Menübefehl **Bearbeiten > Datenbaustein initialisieren**.

Alle Variablen werden wieder mit ihrem vorgesehenen Anfangswert versorgt, d. h. die aktuellen Werte aller Variablen werden durch den jeweiligen Anfangswert überschrieben.

Achtung

Änderungen der Datenwerte werden erst beim Speichern von Datenbausteinen gesichert.

6.5 Anzeigen von Referenzen

6.5.1 Übersicht der möglichen Referenzen

Nutzen Sie die Querverweise "Querverweisliste", "Verwendete Operanden" und "Programmstruktur, um einen Überblick über die Nutzung und Verwendung von Operanden, Speicherbereiche, Bausteine etc. zu erhalten. Zu den Querverweisen gelangen Sie durch Doppelklicken des Symbols "Querverweise" im Projektfenster.

- Nutzen Sie die Querverweise um bei der Programmerstellung und bei Änderungen die Übersicht über verwendete Operanden und Bausteinaufrufe, zu behalten.
- Nutzen Sie die Querverweise um bei einem Programmtest oder der Fehlersuche zu erfahren, welcher Operand in welchem Baustein, mit welchem Befehl bearbeitet wird oder welcher Baustein von welchem aufgerufen wird.
- Nutzen Sie die Querverweise als Teil der Projektdokumentation, um den Endanwendern einen umfassenden Überblick über alle verwendeten Operanden, Speicherbereiche und Bausteine zu liefern.

Die folgende Tabelle gibt Auskunft darüber, welche Informationen Sie den einzelnen Registern finden können:

Ansicht	Anwendung
Querverweisliste	Überblick über die Verwendung von Operanden der Speicherbereiche E, A, M, P, T, Z und DB-, FB-, FC-, SFB- und SFC-Aufrufen innerhalb des Anwenderprogramms. Verwenden Sie die Filterfunktion (vordefinierte Filter oder selbstdefinierte Filter) um die Auswahl der angezeigten Operanden und Speicherbereiche einzuschränken.
Verwendete Operanden	Der Überblick, welche Bits, Bytes, Worte oder Doppelworte der Operanden und Speicherbereiche E, A, und M innerhalb des Anwenderprogramms bereits verwendet sind, ist eine wichtige Grundlage zur Programmierung und Erweiterung des Anwenderprogramms. Das Register "Verwendete Operanden" gibt weiterhin Aufschluss über die verwendeten Zeiten und Zähler.
Programmstruktur	Zeigt die Aufrufhierarchie der Bausteine innerhalb eines Anwenderprogramms und bietet einen Überblick über die verwendeten Bausteine und deren Abhängigkeiten.

6.5.2 Adressübersicht

Wenn Sie die Eingangs- und Ausgangsadressen aller konfigurierten Baugruppen bzw. Module anzeigen wollen, dann wählen Sie den Menübefehl **Ansicht > Adressübersicht**.

Daraufhin blendet STEP 7 Lite die Adressübersicht in Form einer Tabelle ein. In den Spalten der Tabelle finden Sie neben der Adresse und dem Typ der Adresse (E,A) auch Angaben zum Ort (Rack, Steckplatz...) und zur Identifikation der Baugruppe (Name, Bestellnummer), der die Adresse zugewiesen ist.

Die Adressübersicht bleibt im Vordergrund, auch wenn Sie in eine andere Anwendung von STEP 7 Lite wechseln.

Spalten der Adressübersicht ein- und ausblenden

Über Kontextmenü können Sie Spalten ein- und ausblenden.

Beispiel: Klicken Sie mit der rechten Maustaste in die Adressübersicht und wählen Sie den Menübefehl **Spalte Einblenden > Bestellnummer**.

Filtern der Adressübersicht

Mit Hilfe der Optionskästchen "Eingänge" und "Ausgänge" in der Adressübersicht können Sie die Adressübersicht filtern. Wenn Sie z. B. das Optionskästchen "Ausgänge" deaktivieren, werden nur noch Eingangsadressen angezeigt.

6.5.3 Querverweisliste

Die Querverweisliste bietet einen Überblick über die Verwendung von Operanden innerhalb des Anwenderprogramms.

Mit der Anzeige der Querverweisliste erhalten Sie eine Liste der im Anwenderprogramm verwendeten Operanden der Speicherbereiche Eingang (E), Ausgang (A), Merker (M), Zeiten (T), Zähler (Z), Funktionsbaustein (FB), Funktion (FC), Systemfunktionsbaustein (SFB), Systemfunktion (SFC), Peripherie (P) und Datenbaustein (DB), ihrer Adressen (absolute Adresse, Name) und ihrer Verwendung. Sie wird in einem Arbeitsfenster angezeigt.

Jede Zeile in der Sicht entspricht einem Querverweislisteneintrag. Eine Suchfunktion erleichtert Ihnen das gezielte Auffinden bestimmter Operanden und Symbole.

Die Querverweisliste wird mit einem Doppelklick auf das Symbol "Querverweise" im Projektfenster geöffnet. Sie können hier unter den Registern "Querverweisliste", "Verwendete Operanden" und "Programmstruktur" wählen.

Aufbau

Ein Querverweislisteneintrag besteht aus folgenden Spalten:

Spalte	Inhalt/Bedeutung
Operand	absolute Adresse des Operanden
Symbol	Name des Operanden
Baustein	Angabe des Bausteins, in dem der Operand verwendet wird
Baustein Sym.	symbolischer Bezeichner des Bausteins
Netzwerk	Angabe der Nummer des Netzwerks, in dem der Operand verwendet wird
Zeile	Die Positionsangabe innerhalb des Netzwerks, in dem der Operand verwendet wird
Zugriff	Angabe, OB es sich um einen lesenden (R) und/oder schreibenden (W) Zugriff auf den Operanden handelt
Sprache	Angabe der Sprache, in der der Zugriff programmiert wird
Operation	Angabe der Operation, mit der der Operand verwendet wird

Die Spaltenbreite können Sie in der auf dem Bildschirm angezeigten Querverweisliste über Mausbedienung an ihre Anforderungen anpassen.

Sortierung

Voreingestellt in der Querverweisliste ist die Sortierung nach Speicherbereichen. Wenn Sie mit der Maus auf eine Spaltenüberschrift klicken, wird nach den Einträgen dieser Spalte sortiert.

Filtern

Die Querverweisliste können Sie filtern. Sie können vordefinierte Filter verwenden oder eigene Filter erstellen.

Die Filter wählen Sie über die Klappliste "Filter".

Um neue Filter zu erstellen oder bestehende Filter zu ändern, klicken Sie auf die Schaltfläche "Filter". Im Folgedialog können Sie die Eigenschaften der Filter bestimmen und anschließend den Filter übernehmen. Nicht übernommene Filter haben anschließend in der Klappliste zur Auswahl der Filter ein Sternchen (*); die werden beim Speichern des Projekts nicht mitgespeichert.

Beispiel für den Aufbau der Querverweisliste

Operand	Symbol	Baustein	Baustein Sym.	Netzwerk	Zeile	Zugriff	Sprache	Operation
E 1.0	Motor ein	OB 2	Cycle	1		R	AWL	CALL
M1.2	MerkerBit	FC 2	Motor	2	3	RW	KOP	-()-
Z2	Zähler2	FB2	Multiinstanz	5	1		FUP	

6.5.4 Verwendete Operanden

Drei Listen zeigen Ihnen, welche Operanden innerhalb des Anwenderprogramms bereits belegt sind. Diese Anzeige ist eine wichtige Grundlage für Fehlersuche oder Änderungen im Anwenderprogramm.

Verwendungsliste "Verwendete Bits und Bytes"

Mit der Liste "Verwendete Bits und Bytes" erhalten Sie eine Übersicht, welches Bit in welchem Byte der Speicherbereiche Eingang (E), Ausgang (A) und Merker (M) verwendet wird.

Jede Zeile beinhaltet ein Byte des Speicherbereichs, in dem die acht Bits je nach Zugriff gekennzeichnet werden. Zusätzlich wird noch angegeben, OB der Zugriff durch einen Byte-, Wort- oder Doppelwortzugriff erfolgt (Zeile wird blau hinterlegt).

Kennzeichen in der Liste "Verwendete Bits und Bytes":

X	der Operand ist direkt verwendet
Blauer Balken	der Operand ist indirekt bearbeitet (Byte-, Wort- oder Doppelwortzugriff), die Zellen sind blau hinterlegt

Spalten in der Liste "Verwendete Bits und Bytes"

Spalte	Inhalt/Bedeutung
7	Bitnummer des entsprechenden Bytes
6	
5	
4	
3	
2	
1	
0	
B	das Byte ist durch einen Bytezugriff belegt
W	das Byte ist durch einen Wortzugriff belegt
D	das Byte ist durch einen Doppelwortzugriff belegt

Beispiel

Folgendes Beispiel zeigt den typischen Aufbau einer Verwendungsliste für Eingänge, Ausgänge und Merker (E/A/M).

	7	6	5	4	3	2	1	0	B	W	D
EB0		X	X	X	X	X	X				
EB1		X	X	X			X	X	X		
AB4						X	X	X			
AB5		X	X	X		X	X	X			
MB2											
MB3											
MB4											
MB5											

In der ersten Zeile ist die Belegung vom Eingangsbyte EB 0 aufgeführt. Eingänge des Operanden EB 0 werden direkt verwendet (Bitzugriff). In den Spalten "1", "2", "3", "4", "5", und "6" steht jeweils ein "X" als Kennzeichen für den Bitzugriff. Es erfolgt aber auch ein Wortzugriff auf die Merkerbytes 2 und 3 bzw. 4 und 5. Daher steht in der Spalte "W" jeweils ein "Balken" und die Zellen sind noch zusätzlich blau hinterlegt. Die schwarze Spitze der Balken zeigt den Beginn des Wortzugriffs an.

Listen "Verwendete Zeiten" und "Verwendete Zähler"

Mit den Listen "Verwendete Zeiten" und "Verwendete Zähler" erhalten Sie eine Übersicht, welche Zeiten (T) und Zähler (Z) verwendet sind.

Kennzeichen in der Liste "Verwendete Zeiten/Zähler":

- Blaues Feld: Der Timer/Zähler wird verwendet
- Weißes Feld: Der Timer/Zähler wird nicht verwendet

Beispiel für "Verwendete Zeiten"

Zeiten	0	1	2	3	4	5	6	7	8	9
T-		T1	T2							
T1-									T18	
T2-	T20									
T17-										T179
T22-					T224					

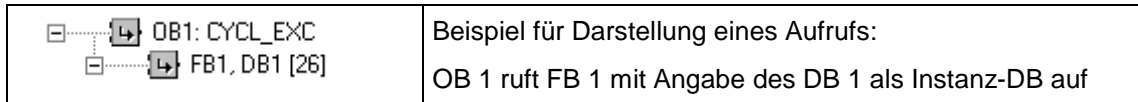
In diesem Beispiel sind die Zeiten T1, T2, T18, T20, T179 und T224 belegt.

6.5.5 Programmstruktur

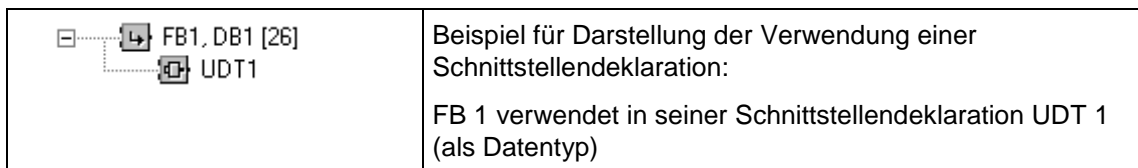
Die Programmstruktur beschreibt die Beziehungen oder Abhängigkeiten der verwendeten Bausteine innerhalb eines Anwenderprogramms.

Eine Beziehung bzw. eine Abhängigkeit ist gegeben

- durch Aufruf (z. B. Baustein A ruft Baustein B über die Anweisung CALL auf)



- durch Verwendung einer Schnittstellendeklaration (z. B. Baustein A verwendet UDT B oder FB C in seiner Schnittstellendeklaration)

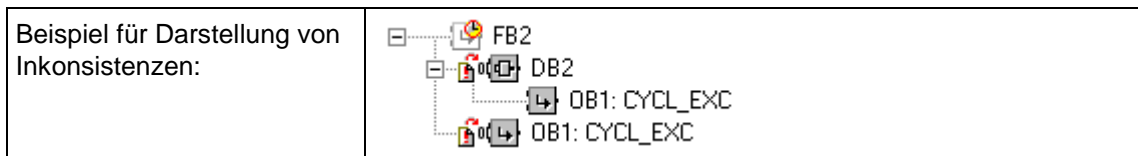


Lokaldatenbedarf

Zusätzlich erhalten Sie in der Programmstruktur einen Überblick über den Lokaldatenbedarf. Wenn z. B. ein OB 1 existiert, ist neben dem OB 1 in spitzen Klammern der maximale Lokaldatenbedarf der CPU bei Ausführung des OBs angeben. Der Lokaldatenbedarf eines Aufrufpfades wird durch Aufaddieren des Lokaldatenbedarfs jedes einzelnen Bausteins im Aufrufpfad ermittelt, ausgehend vom OB 1. Ebenso wird der maximale Lokaldatenbedarf für die Synchronfehler-OBs (OB 121 und OB 122) ermittelt und hinter den Text von OB 1 in spitzen Klammern angegeben.

Inkonsistenzen

Außerdem werden Inkonsistenzen visualisiert, wie sie z. B. durch Schnittstellenänderungen an einem Baustein entstehen. Über die Programmstruktur können Sie die Bausteine öffnen (Funktion "Gehe zu...") und die Inkonsistenzen sukzessiv beseitigen oder Sie können mit dem Menübefehl **Extras > Programmkonsistenz herstellen** die Beseitigung der Inkonsistenzen automatisch vornehmen lassen.



Programmstruktur anzeigen lassen

Die Programmstruktur wird mit einem Doppelklick auf das Symbol "Querverweise" im Projektfenster geöffnet. Wählen Sie hier unter den Registern "Querverweisliste", "Verwendete Operanden" und "Programmstruktur" am unteren Rand des Fensters das Register "Programmstruktur".

Anzeigemöglichkeiten und Einstellungen der Programmstruktur

Die Einstellungen in den Feldern "Startbaustein" und "Anzeigen" haben Einfluß auf die Darstellung der Programmstruktur. Im folgenden sind die Einstellungen erläutert. Um die Beschreibung einfacher zu halten, wird im folgenden unter "Baustein" auch ein Datentyp (UDT) verstanden.

Feld	Mögliche Einstellungen	Bedeutung
Startbaustein	"System" und vorhandene Bausteine im Projekt	Der ausgewählte Baustein wird in der Programmstruktur markiert, und zwar dort, wo er das erste Mal in der angezeigten Programmstruktur vorkommt. "System" ist die oberste Hierarchiestufe, die das CPU-Betriebssystem repräsentiert. Vom System werden alle OBs aufgerufen.
Anzeigen	<ul style="list-style-type: none"> • Aufrufstruktur • Aufrufstruktur mit Mehrfachaufrufen • Verwendungsstruktur • Verwendungsstruktur (nur Konflikte) 	<p>Die Aufrufstruktur zeigt die aufgerufenen Bausteine und die Beziehung zwischen diesen Bausteinen an, beginnend bei OB1. Ganz links stehen die OBs, die nur durch das CPU-Betriebssystem aufgerufen werden können. Darunter eingerückt die Bausteine, die vom jeweiligen OB aufgerufen oder verwendet werden. Diese Hierarchie setzt sich je nach Schachtelungstiefe der Aufrufe fort. Angezeigt wird nur der erste Aufruf bzw. die erste Verwendung.</p> <p>Die Aufrufstruktur mit Mehrfachaufrufen zeigt alle Aufrufe bzw. Verwendungen von Bausteinen an.</p> <p>Die Verwendungsstruktur zeigt für jeden Baustein im Projekt die Abhängigkeiten zu anderen Bausteinen an. Ganz links steht der Baustein und darunter eingerückt die Bausteine, die diesen Baustein aufrufen oder verwenden.</p> <p>Die Verwendungsstruktur (nur Konflikte) zeigt nur Beziehungen mit Konflikten bezüglich des Schnittstellenzeitstempels oder bezüglich der Symboltabelle:</p> <ul style="list-style-type: none"> • Der Schnittstellenzeitstempel des aufgerufenen (bzw. verwendeten) Bausteins hat sich seit dem letzten Speichern des aufrufenden Bausteins geändert (kann auch beim Laden von Bausteinen in das PG passieren). • Im Projekt ist "Operandenvorrang: Symbol" eingestellt und Symbole wurden nach Speichern der Bausteine geändert. <p>Wenn keine Konflikte bestehen, wird nur das "System"-Symbol angezeigt.</p>

Hinweis

Die Anzeigen (Aufrufstruktur etc.) können Sie auch über Menübefehle im Menü Ansicht (**Menü Ansicht > Aufrufstruktur**) ändern.

Darstellung von Bausteinabhängigkeiten in der Programmstruktur

Symbol	Bedeutung	Beispiel
	Baustein regulär aufgerufen mit CALL	
	Baustein unbedingt aufgerufen mit UC	
	Baustein bedingt aufgerufen mit CC	
	Datenbaustein geöffnet und auf Inhalte zugegriffen (z. B. L DB 1.DBW 10)	
	Baustein verwendet Schnittstellendeklaration z. B. FB, SFB, UDT	

Darstellung von nicht verwendeten Bausteinen

Bausteine, die im Programm existieren, aber nicht verwendet werden, sind durchgekreuzt:

Symbol	Bedeutung
	Nicht verwendeter Datenbaustein
	Nicht verwendeter FB, FC, SFB, SFC, UDT

Im folgenden Beispiel sind sowohl Baustein-Aufruf als auch Verwendung der Schnittstellendeklaration eines Baustein in einer Aufrufstruktur erläutert.

	<p>FB 1 wird nicht aufgerufen</p> <p>FB 1 verwendet FB 2 als Multiinstanz in der eigenen Schnittstellendeklaration</p> <p>FB 2 verwendet UDT 3 in der eigenen Schnittstellendeklaration. Dass FB 2 den FB 3 aufruft, wird nicht hier angezeigt, sondern unter dem Aufruf-Symbol, siehe (3)</p> <p>FB 1 ruft Multiinstanz FB 2 auf (Call) und FB 2 verwendet UDT 3 in seiner Schnittstellendeklaration und FB 2 ruft FC 3 auf (Call)</p>
--	---





Darstellung von Aufrufen nicht vorhandener Bausteine

Wenn ein Baustein gelöscht wurde, dieser Baustein aber von einem anderen Baustein verwendet wird, dann wird der Text solcher Bausteine rot eingefärbt und durch "???" am Ende ergänzt.

Darstellung von Rekursionen in Bausteinabhängigkeiten

Rekursionen werden verursacht durch folgende Bausteinabhängigkeiten:

- Baustein 1 ruft Baustein 2 auf **und** Baustein2 ruft Baustein 1 auf.
- Baustein 1 ruft Baustein 2 auf **und** Baustein 2 verwendet Schnittstellendeklaration des Bausteins 1, z. B.: FB1 ruft Instanz-DB zu FB 1 auf
- Baustein 1 verwendet Schnittstellendeklaration des Bausteins 2 **und** Baustein 2 verwendet Schnittstellendeklaration des Bausteins 1. Diese Konstellation ist nicht erlaubt und kann nur durch Einkopieren von Bausteinen in ein bestehendes Projekt auftreten. Die betroffenen Bausteine sind nicht mehr übersetzbar.
- Rekursionen in Bausteinabhängigkeiten werden durch einen überlagerten Pfeil dargestellt:

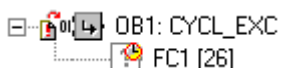
Symbol	Bedeutung
	Rekursion und Bausteinaufruf mit CALL
	Rekursion und Bausteinaufruf mit UC
	Rekursion und Bausteinaufruf mit CC
	Rekursion durch Schnittstellendeklaration in verwendetem Baustein

Darstellung von Zeitstempelkonflikten in Bausteinabhängigkeiten

Unterschiedliche Zeitstempel von Schnittstellen können zu Konflikten führen, wenn die Schnittstelle des aufgerufenen Bausteins sich geändert hat (z. B. weniger Variablen hat), der aufrufende Baustein aber noch die "alte" Schnittstellendeklaration für den Aufruf verwendet.

Als Indikator für möglicherweise inkonsistente Schnittstellen wird der Schnittstellen-Zeitstempel verwendet. Wenn der Schnittstellen-Zeitstempel des aufgerufenen Bausteins jünger ist als der des aufrufenden Bausteins, wird das in der Programmstruktur durch ein überlagertes Uhrensymbol dargestellt.

Dem aufrufende Baustein wird bei Konflikten ein Status-Symbol vorangestellt, das darauf hinweist, dass der Baustein neu übersetzt werden muss.

Beispiel für Darstellung von Zeitstempelkonflikten	
--	---

Darstellung von Symbolkonflikten in Bausteinabhängigkeiten

Wenn in den allgemeinen Einstellungen des Projekts 'Operandenvorrang: Symbol' eingestellt ist, können Inkonsistenzen (Symbolkonflikte) entstehen, wenn Sie einen Baustein speichern und nachträglich in der Symboltabelle ein im Baustein verwendetes Symbol ändern.

Die Symbole für Bausteinabhängigkeit ändern sich folgendermaßen:

Symbol	Bedeutung
	Symbolkonflikt und Bausteinaufruf mit CALL
	Symbolkonflikt und Bausteinaufruf mit UC
	Symbolkonflikt und Bausteinaufruf mit CC
	Symbolkonflikt durch Schnittstellendeklaration in verwendetem Baustein

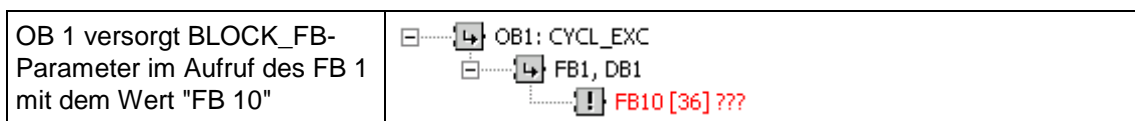
Darstellung von Bausteinnummern bei indirekter Angabe bzw. bei Verwendung des Paramertyps BLOCK

Die Bausteinnummer von indirekt angegebener Bausteinnummer im Aufruf (z. B. UC FC[MW 10]) wird als Reihe von Fragezeichen angezeigt:



Wenn in der Schnittstellendeklaration eines Bausteins ein Eingangsparameter vom Paramertyp BLOCK verwendet wurde (z. B. BLOCK_FB), dann legt der aufrufende Baustein die Bausteinnummer fest. Wenn STEP 7 Lite die Nummer rekursiv ermitteln kann, wird sie auch angezeigt. Der Text neben dem Baustein-Symbol hat in der Programmstruktur dann eine rote Farbe. Wenn die Bausteinnummer nicht ermittelt werden kann, z. B. weil der Baustein mit Paramertyp BLOCK gar nicht aufgerufen wird, werden wie oben Fragezeichen angezeigt.

Beispiel:



6.5.6 Arbeiten mit Referenzdaten

6.5.6.1 Schnelles Positionieren auf Verwendungsstellen im Programm

Sie können Referenzdaten benutzen, um beim Programmieren auf die Verwendungsstellen eines Operanden zu positionieren.

Prinzipielle Vorgehensweise

1. Markieren Sie in einem geöffneten Baustein den Operanden.
2. Wählen Sie den Menübefehl **Bearbeiten > Gehe zu > Verwendungsstelle**. Nun wird ein Dialogfeld angezeigt, das eine Liste mit den Verwendungsstellen des Operanden im Programm enthält.
3. Wählen Sie die Option "Überlappender Zugriff auf Speicherbereiche", wenn zusätzlich die Verwendungsstellen der Operanden angezeigt werden sollen, deren Adressen oder Adressbereich die Adresse bzw. den Adressbereich des aufgerufenen Operanden überlappen. Die Tabelle wird um die Spalte "Operand" erweitert.
4. Markieren Sie eine Verwendungsstelle in der Liste und klicken Sie auf die Schaltfläche "Gehe zu".

Liste der Verwendungsstellen

Die Liste der Verwendungsstellen im Dialogfeld enthält folgende Angaben:

- Baustein, in dem der Operand verwendet wird
- Bausteinsymbol, falls vorhanden
- Details, d.h. von der Erstsprache des Bausteins/der Quelle (SCL) abhängige Information zur Verwendungsstelle und ggf. der Operationssprachabhängige Informationen
- Zugriffsart auf den Operanden: Lesend (R), schreibend (W), lesend und schreibend (RW), nicht ermittelbar (?)
- Bausteinsprache

Sie können die Anzeige der Verwendungsstellen filtern und sich beispielsweise nur die schreibenden Zugriffe auf einen Operanden anzeigen lassen. Ausführliche Information zu den Eingabemöglichkeiten und Anzeigen finden Sie in der Online-Hilfe zu diesem Dialogfeld.

Achtung

Referenzdaten sind nur offline vorhanden. Diese Funktion arbeitet daher stets mit den Querverweisen der Bausteine offline, auch wenn Sie die Funktion in einem Baustein online aufrufen.

6.5.6.2 Beispiel zum Arbeiten mit Verwendungsstellen

Sie möchten feststellen, an welchen Stellen der Ausgang A1.0 (direkt/indirekt) gesetzt wird. Als Beispiel dient der folgende AWL-Code im Baustein OB1:

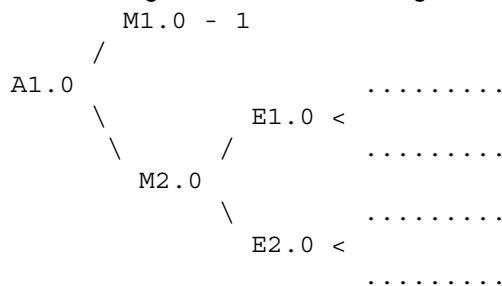
```
Netzwerk1: .....
U A 1.0 // in diesem Beispiel
= A 1.1 // nicht relevant
```

```
Netzwerk 2:
U M1.0
U M2.0
= A 1.0 // Zuweisung
```

```
Netzwerk3:
//nur Kommentarzeile
SET
= M1.0 // Zuweisung
```

```
Netzwerk 4:
U E 1.0
U E 2.0
= M2.0 // Zuweisung
```

Für A1.0 ergibt sich daraus der folgende Zuweisungsbaum:



Dann gehen Sie so vor:

1. Im Bausteineditor im OB 1 auf A1.0 (NW 1, Anw 1) positionieren.
2. Über **Bearbeiten > Gehe zu > Verwendungsstelle** oder mit der rechten Maustaste "Verwendungsstelle" aufrufen.
Im Dialogfeld werden unter anderem alle Zuweisungen auf A1.0 angezeigt:

OB1	Cycle Execution	NW 2	Anw 3	/=	W	AWL
OB1	Cycle Execution	NW 1	Anw 1	/U	R	AWL
3. Mittels "Gehe zu" aus dem Dialogfeld zu "NW 2 Anw 3" in den Editor springen:

```
Netzwerk 2:
U M1.0
U M2.0
= A 1.0
```


Sowohl die Zuweisungen an M1.0 als auch an M2.0 müssen nun kontrolliert werden. Also im Bausteineditor zuerst auf M1.0 positionieren.

4. Über **Bearbeiten > Gehe zu > Verwendungsstelle** oder mit rechter Maustaste "Verwendungsstelle" aufrufen. Im Dialogfeld werden unter anderem alle Zuweisungen auf M1.0 angezeigt:

```
OB1      Cycle Execution      NW 3  Anw 2  /=      W      AWL
OB1      Cycle Execution      NW 2  Anw 1  /U      R      AWL
```

5. Mittels "Gehe zu" zu "NW 3 Anw 2" in den Bausteineditor springen.
6. Im Bausteineditor in Netzwerk 3 wird festgestellt, daß die Belegung von M1.0 uninteressant ist (da immer TRUE) und stattdessen die Belegung von M2.0 untersucht werden müßte.
7. Entweder das noch geöffnete Dialogfeld "Gehe zu Verwendungsstelle" nach vorne holen oder aus der aktueller Position im Bausteineditor "Gehe zur Verwendungsstelle" aufrufen.
8. Mittels "Gehe zu" (wie in Punkt 3) aus dem Verwendungsstellendialog zu "NW 2 Anw 3" in den Editor zu springen:

```
Netzwerk 2:
U M1.0
U M2.0
= A 1.0
```

9. In Punkt 4ff. wurde die Zuweisung an M1.0 kontrolliert. Jetzt müssen alle (direkten/indirekten) Zuweisungen auf M2.0 kontrolliert werden. Also im Editor auf M2.0 positionieren und "Gehe zur Verwendungsstelle" aufrufen: Es werden u.a. alle Zuweisungen auf M2.0 angezeigt:

```
OB1      Cycle Execution      NW 4  Anw 3  /=      W      AWL
OB1      Cycle Execution      NW 2  Anw 2  /U      R      AWL
```

10. Mittels "Gehe zu" zu "NW 4 Anw 3" in den Bausteineditor springen:

```
Netzwerk 4:
U E 1.0
U E 2.0
= M2.0
```

11. Jetzt müssen die Zuweisungen an E1.0 und E2.0 überprüft werden. Dies wird in diesem Beispiel nicht mehr erläutert, da die weitere Vorgehensweise sich von der bisherigen (Punkt 4ff.) nicht unterscheidet

Durch abwechselndes Springen zwischen Bausteineditor und dem Verwendungsstellen-Dialog können Sie so die relevanten Stellen Ihres Programms ermitteln und untersuchen.

6.5.6.3 Schritte zum Arbeiten mit Referenzdaten

Springen aus der Querverweisliste in den Programmteil

Gehen Sie folgendermaßen vor, um aus der Querverweisliste in den entsprechenden Programmabschnitt zu springen:

1. Markieren Sie die Zeile des gewünschten Operanden.
2. Betätigen Sie die Schaltfläche "Gehe zu Verwendungsstelle".

Die Funktion ist auch über den Menübefehl **Bearbeiten > Gehe zu > Verwendungsstelle** ausführbar.

Alternative Vorgehensweise:

1. Markieren Sie in der Querverweisliste einen Operanden.
2. Klicken Sie auf die rechte Maustaste, um ein kontextsensitives Menü aufzublenden.
3. Wählen Sie daraus den Menübefehl "Gehe zur Verwendungsstelle".

Springen aus der Programmstruktur in Programmteil

Gehen Sie folgendermaßen vor, um aus der Programmstruktur in den entsprechenden Programmteil zu springen:

1. Markieren Sie in der "Programmstruktur" einen Baustein.
2. Drücken Sie auf die rechte Maustaste. Ein Kontextmenü wird aufgeblendet.
3. Wählen Sie den Eintrag "Gehe zu Baustein", um den Baustein selbst zu öffnen oder wählen Sie den Eintrag "Gehe zum Aufruf", um den übergeordneten Baustein zu öffnen und darin auf den Aufruf des markierten Bausteins zu positionieren.

Der Eintrag "Gehe zum Aufruf" ist nur wählbar, wenn es zu dem markierten Baustein einen übergeordneten Baustein gibt.

Die über das Kontextmenü aufrufbaren Befehle stehen auch in der Menüleiste zur Verfügung:

Bearbeiten > Gehe zu > Baustein bzw.

Bearbeiten > Gehe zu > Aufruf

Anzeigen überlappender Zugriffe

Um Querverweise zu Operanden anzuzeigen, deren Adressbereiche sich überlappen, gehen Sie folgendermaßen vor:

1. Markieren Sie einen Operanden in der Querverweisliste der Referenzdaten.
2. Drücken Sie die rechte Maustaste und wählen Sie aus dem dann angezeigten Kontextmenü "**Querverweise für Operand**".

In einem weiteren Fenster werden daraufhin auch die Querverweise zu Operanden angezeigt, deren Adressbereiche sich mit dem markierten Operanden überlappen.

Im Bausteineditor gehen Sie folgendermaßen vor:

1. Markieren Sie den Operand im Anweisungsteil.
2. Wählen Sie den Menübefehl Bearbeiten > Gehe zu > Verwendungsstelle.
3. Wählen Sie im Dialogfeld "Gehe zu Verwendungsstelle" die Option "Überlappender Zugriff auf Speicherbereiche".

6.6 Programmkonsistenz herstellen und Zeitstempel als Bausteineigenschaft

6.6.1 Programmkonsistenz herstellen

Einführung

Müssen während oder nach der Programmerstellung die Schnittstellen bzw. der Code einzelner Bausteine angepasst oder ergänzt werden, so kann dies zu Programminkonsistenzen zwischen aufrufenden und aufgerufenen Bausteinen bzw. Referenzbausteinen führen.

Die Funktion "Programmkonsistenz herstellen" nimmt Ihnen einen großen Teil dieser Korrekturarbeit ab. Die Funktion " Programmkonsistenz herstellen" bereinigt selbständig einen großen Teil aller Zeitstempelkonflikte und Programminkonsistenzen. Bei Bausteinen, deren Programminkonsistenzen nicht automatisch bereinigt werden konnten, führt Sie die Funktion an die zu ändernden Positionen im Bausteineditor. Dort nehmen Sie die notwendigen Änderungen vor. Schritt für Schritt werden alle Programminkonsistenzen bereinigt und die Bausteine übersetzt.

Herstellen der Programmkonsistenz

Mit dem Start der Programmkonsistenz-Prüfung werden die Zeitstempel der Bausteinschnittstellen geprüft und Bausteine, bei denen es zu Programminkonsistenzen führen kann, werden markiert.

1. Wählen Sie den Menübefehl **Extras > Programmkonsistenz herstellen**.
STEP 7 Lite bereinigt (soweit möglich) automatisch Zeitstempelkonflikte und Programminkonsistenzen und die Bausteine werden übersetzt. Konnte der Zeitstempelkonflikt bzw. die Inkonsistenz in einem Baustein nicht automatisch bereinigt werden, so erfolgt eine Fehlermeldung.
Die betreffenden Bausteine werden im Projektfenster rot (= fehlerhaft) und fett markiert.
2. Doppelklicken Sie im Projektfenster auf einen rot markierten Baustein.
Der Baustein wird geöffnet. Die Fehler werden in der Ausgabeleiste des Bausteineditors angezeigt.
3. Doppelklicken Sie auf einen Eintrag in der Ausgabeleiste.
Die fehlerhafte Stelle wird angezeigt.
4. Beheben Sie alle Fehler und übernehmen Sie die Änderungen. Schließen Sie den Baustein. Wiederholen Sie den Ablauf für alle fehlerhaften Bausteine. Sind alle Fehler bereinigt, so wird der entsprechende Baustein im Projektfenster zur Kontrolle in schwarzer Fettschrift markiert.
5. Speichern Sie das Projekt.

Automatisches Erzeugen von Instanz-DBs beim Herstellen der Programmkonsistenz

Wenn Sie einen FB ohne den zugehörigen Instanz-DB sowie den FB-aufrufenden Baustein von der CPU in das PG laden, dann wird beim Herstellen der Programmkonsistenz automatisch ein Instanz-DB zum FB erzeugt.

6.6.2 Zeitstempel und Zeitstempelkonflikte

Bausteine enthalten einen Code-Zeitstempel und einen Schnittstellen-Zeitstempel. Diese Zeitstempel werden im Register "Eigenschaften" des Bausteineditors angezeigt. Anhand von Zeitstempeln wird die Konsistenz von STEP 7 Lite-Programmen überwacht.

STEP 7 Lite zeigt einen Zeitstempelkonflikt an, wenn beim Vergleich von Zeitstempeln ein Regelverstoß erkannt wird. Folgende Regelverstöße können auftreten:

- Ein aufgerufener Baustein ist jünger als der aufrufende Baustein (CALL).
- Ein referenzierter Baustein ist jünger als der Baustein, der ihn verwendet.
Beispiele:
 - Ein UDT ist jünger als der Baustein, der ihn verwendet, z. B. ein DB oder ein anderer UDT, oder ein FC, FB, OB, der den UDT in der Variablendeklarationstabelle verwendet.
 - Ein FB ist jünger als der zugehörige Instanz-DB.
 - In FB1 ist ein FB2 als Multiinstanz definiert und FB2 ist jünger als FB1.

Hinweis

Auch wenn die Relation der Schnittstellen-Zeitstempel korrekt ist, können Inkonsistenzen vorkommen:

- Die Definition der Schnittstelle des referenzierten Bausteins stimmt nicht mit der an seiner Verwendungsstelle verwendeten Schnittstelle überein.

Solche Inkonsistenzen werden Schnittstellenkonflikte genannt. Sie können z. B. durch Kopieren von Bausteinen verschiedener Programme entstehen.

6.6.3 Zeitstempel in Codebausteinen

Code-Zeitstempel

Hier wird der Erstellungszeitpunkt des Bausteins eingetragen. Der Zeitstempel wird aktualisiert

- bei Änderung des Programmcodes
- bei Änderungen der Schnittstellenbeschreibung
- bei Änderung des Kommentars
- bei Änderungen der Bausteineigenschaften

Schnittstellen-Zeitstempel

Der Zeitstempel wird aktualisiert bei

- Änderungen der Schnittstellenbeschreibung (Änderung von Datentypen oder Initialwerten, neue Parameter)

Der Zeitstempel wird nicht aktualisiert bei:

- Änderung von Symbolen
- Änderung von Kommentaren in der Variablendeklarationstabelle
- Änderungen im TEMP-Bereich

Regel für Bausteinaufrufe

- Der Schnittstellen-Zeitstempel des aufgerufenen Bausteins muss älter sein als der Code-Zeitstempel des aufrufenden Bausteins.
- Ändern Sie die Schnittstelle eines Bausteins nur, wenn kein Baustein geöffnet ist, der diesen aufruft. Denn wenn Sie die aufrufenden Bausteine später als den geänderten speichern, so erkennen sie diese Inkonsistenz nicht am Zeitstempel.

Vorgehen bei Zeitstempelkonflikt

Ein Zeitstempelkonflikt wird beim Öffnen des Aufrufers angezeigt. Nach Änderung einer FC- oder FB-Schnittstelle werden alle Aufrufe dieses Bausteins in aufrufenden Bausteinen rot dargestellt.

Wird die Schnittstelle eines Bausteins geändert, so müssen alle Bausteine angepasst werden, die diesen Baustein aufrufen.

Nach Änderung einer FB-Schnittstelle müssen die vorhandenen Multiinstanzdefinitionen und Instanz-Datenbausteine aktualisiert werden.

6.6.4 Zeitstempel bei globalen Datenbausteinen

Code-Zeitstempel

Der Zeitstempel wird aktualisiert

- bei der Ersterstellung,
- bei Änderungen in der Deklarationssicht oder in der Datensicht des Bausteins.

Schnittstellen-Zeitstempel

Der Zeitstempel wird aktualisiert

- bei Änderungen der Schnittstellenbeschreibung in der Deklarationssicht (Änderung von Datentypen oder Initialwerten, neue Parameter).

6.6.5 Zeitstempel bei Instanz-Datenbausteinen

Ein Instanz-Datenbaustein speichert die Formalparameter und statischen Daten von Funktionsbausteinen.

Code-Zeitstempel

Hier wird der Erstellungszeitpunkt der Instanz-Datenbausteine eingetragen. Der Zeitstempel wird aktualisiert, wenn Sie Aktualwerte in der Datensicht des Instanz-Datenbausteins eintragen. Änderungen der Struktur eines Instanz-Datenbausteins durch den Benutzer sind nicht möglich; da die Struktur vom zugehörigen Funktionsbaustein (FB) oder Systemfunktionsbaustein (SFB) abgeleitet ist.

Schnittstellen-Zeitstempel

Beim Erzeugen eines Instanz-Datenbausteins wird der Schnittstellen-Zeitstempel des zugehörigen FB oder SFB eingetragen.

Regeln für konfliktfreies Öffnen

Die Schnittstellen-Zeitstempel des FB/SFB und des zugeordneten Instanz-Datenbausteins müssen übereinstimmen.

Vorgehen bei Zeitstempelkonflikt

Wenn Sie die Schnittstelle eines FB ändern, wird der Schnittstellen-Zeitstempel des FB aktualisiert. Beim Öffnen eines zugehörigen Instanz-Datenbausteins wird ein Zeitstempelkonflikt angezeigt, da die Zeitstempel des Instanz-Datenbausteins und des FB nicht mehr übereinstimmen. Im Deklarationsteil des DB wird die Schnittstelle mit vom Compiler generierten Symbolen (Pseudosymbolik) dargestellt. Der Instanz-Datenbaustein ist nur noch zur Ansicht nutzbar.

Um solche Zeitstempelkonflikte zu beheben, müssen Sie den zu einem geänderten FB gehörigen Instanz-DB neu erzeugen.

6.6.6 Zeitstempel bei UDTs und von UDTs abgeleiteten DBs

Anwenderdefinierte Datentypen (UDT) können z. B. zur Erzeugung mehrerer Datenbausteine mit der gleichen Struktur genutzt werden.

Code-Zeitstempel

Der Code-Zeitstempel wird bei jeder Änderung aktualisiert.

Schnittstellen-Zeitstempel

Der Schnittstellen-Zeitstempel wird bei Änderungen der Schnittstellenbeschreibung (Änderung von Datentypen oder Initialwerten, neue Parameter) aktualisiert.

Regeln für konfliktfreies Öffnen

- Der Schnittstellen-Zeitstempel des anwenderdefinierten Datentyps muss älter sein als der Schnittstellen-Zeitstempel in Codebausteinen, in denen dieser Datentyp verwendet wird.
- Der Schnittstellen-Zeitstempel des anwenderdefinierten Datentyps muss identisch sein mit dem Zeitstempel eines aus einem UDT abgeleiteten DB.
- Der Schnittstellen-Zeitstempel des anwenderdefinierten Datentyps muss jünger sein als der Zeitstempel eines unterlagerten UDT.

Vorgehen bei Zeitstempelkonflikt

Wenn Sie eine UDT-Definition ändern, die in einem DB, FC, FB oder einer anderen UDT-Definition verwendet wird, zeigt STEP 7 Lite beim Öffnen eines solchen Bausteins ein Zeitstempelkonflikt an.

Die UDT-Komponente wird als Struktur aufgefüchert dargestellt. Alle Variablennamen werden durch vom System vorgegebene Werte überschrieben.

6.6.7 Vermeiden von Fehlern beim Aufrufen von Bausteinen

STEP 7 Lite überschreibt Daten in DB-Register

STEP 7 Lite ändert die Register der S7-300-CPU bei verschiedenen Operationen. Die Inhalte der DB- und DI-Register beispielsweise werden getauscht, wenn Sie einen FB aufrufen. Dadurch kann der Instanz-DB des aufgerufenen FB geöffnet werden, ohne die Adresse des vorherigen Instanz-DB zu verlieren.

Wenn Sie mit der absoluten Adressierung arbeiten, können Fehler beim Zugreifen auf Daten auftreten, die in den Registern gespeichert sind: in einigen Fällen werden die Adressen in dem Register AR1 (Adressregister 1) und in dem DB-Register überschrieben. Dadurch kann es sein, dass Sie falsche Adressen lesen oder in falsche Adressen schreiben.



Gefahr

Gefahr von Sachschäden und Personenschäden bei der Verwendung von:

1. CALL FC, CALL FB, CALL Multiinstanz
2. vollqualifizierten DB-Zugriffen (z. B. DB20.DBW10)
3. Zugriffen auf Variablen eines zusammengesetzten Datentyps

Es kann dabei geschehen, dass die Inhalte von DB-Registern (DB und DI), Adressregistern (AR1, AR2) und Akkus (AKKU1, AKKU2) verändert werden.

Ebenso kann beim FB-CALL/FC-CALL das Verknüpfungsergebnis VKE nicht als zusätzlicher (impliziter) Parameter verwendet werden.

Wenn Sie die oben genannten Programmiermöglichkeiten nutzen, müssen Sie selbst für eine Wiederherstellung der Inhalte Sorge tragen, da es sonst zu einem Fehlverhalten kommen kann.

Speichern von korrekten Daten

Kritisch wird es mit den Inhalten des DB-Registers, wenn Sie im abgekürzten Format der absoluten Adressen auf Daten zugreifen. Gehen Sie beispielsweise davon aus, dass DB 20 geöffnet ist (und seine Nummer im DB-Register gespeichert ist), können Sie DBX0.2 angeben, um auf die Daten zuzugreifen, die in Bit 2 von Byte 0 des DB gespeichert sind, dessen Adresse in dem DB-Register abgelegt ist (also DB20). Enthält das DB-Register allerdings eine andere DB-Nummer, greifen Sie auf die falschen Daten zu.

Fehler beim Zugreifen auf Daten des DB-Registers können Sie vermeiden, wenn Sie die folgende Methode zum Adressieren von Daten verwenden:

- Verwenden Sie symbolische Adressierung
- Verwenden Sie die vollständige absolute Adresse (z. B. DB20.DBX0.2)

Bei diesen beiden Adressierungsmethoden öffnet STEP 7 Lite automatisch den richtigen DB. Wenn Sie das Register AR1 für die indirekte Adressierung verwenden, müssen Sie immer die richtige Adresse in AR1 laden.

Situationen, in denen Register modifiziert werden

Die Manipulation der Adressregister zur indirekten Adressierung ist ausschließlich in AWL relevant. Die anderen Sprachen unterstützen keinen indirekten Zugriff auf die Adressregister.

Die Anpassung des DB-Registers durch den Compiler muss bei allen Programmiersprachen beachtet werden, um eine korrekte Parameterübergabe bei Bausteinaufrufen sicherzustellen.

In den folgenden Situationen werden die Inhalte des Adressregisters AR1 und des DB-Registers des aufrufenden Bausteins überschrieben:

Situation	Erläuterung
Bei Aktualparametern aus einem DB	<ul style="list-style-type: none"> Nachdem Sie einem Baustein einen Aktualparameter zugeordnet haben, der in einem DB gespeichert ist (z. B. DB20.DBX0.2), öffnet STEP 7 Lite diesen DB (DB 20) und passt dabei den Inhalt des DB-Registers an. Das Programm arbeitet im Anschluss an den Bausteinaufruf dann mit dem angepassten DB.
Bei Aufruf von Bausteinen in Zusammenhang mit höheren Datentypen	<ul style="list-style-type: none"> Nach einem Bausteinaufruf aus einem FC, der eine Komponente eines Formalparameters eines höheren Datentyps (String, Array, Struct oder UDT) an den aufgerufenen Baustein übergibt, wird der Inhalt von AR1 und des DB Registers des aufrufenden Bausteins modifiziert. Dasselbe gilt bei Aufruf aus einem FB, wenn der Parameter im var_in_out Bereich des Aufrufers liegt.
Bei Zugriff auf Komponenten höheren Datentyps	<ul style="list-style-type: none"> Beim Zugriff eines FB auf eine Komponente eines Formalparameters höheren Datentyps im var_in_out Bereich (String, Array, Struct oder UDT) verwendet STEP 7 Lite das Adressregister AR1 und das DB-Register. Dadurch werden die Inhalte der beiden Register modifiziert. Beim Zugriff eines FC auf eine Komponente eines Formalparameters höheren Datentyps (String, Array, Struct oder UDT) verwendet STEP 7 Lite das Adressregister AR1 und das DB-Register. Dadurch werden die Inhalte der beiden Register modifiziert.

Achtung

- Beim Aufruf eines FB aus einem Baustein mit Bausteinversion 1 wird der Aktualparameter für den ersten booleschen In- bzw. In_out-Parameter nicht korrekt übergeben, wenn der Befehl vor dem Call nicht VKE begrenzend ist. In diesem Fall wird er mit dem anstehenden VKE verknüpft.
 - Beim Aufruf eines FB (Single oder Multiinstanz) wird das Adressregister AR2 beschrieben.
 - Wird innerhalb eines FB das Adressregister AR2 modifiziert, ist die ordnungsgemäße Bearbeitung dieses FB nicht mehr gewährleistet.
 - Wird einem ANY-Parameter nicht die vollständige absolute DB-Adresse übergeben, enthält der ANY-Pointer nicht die DB-Nummer des aufgeschlagenen DBs, sondern immer die Nummer 0.
-

6.6.8 Hinweise zur Änderung von Register-Inhalten

Wenn Sie die im Anschluss beschriebenen Programmiermöglichkeiten und gleichzeitig die genannten Register/Akkus nutzen, müssen Sie selbst für eine Wiederherstellung der Register- bzw. Akku-Inhalte sorgen, da es sonst zu einem Fehlverhalten kommen kann.

Die Verwendung der folgenden höheren Sprachkonstrukte kann dazu führen, dass die Inhalte von **DB-Register** und **Adressregister AR1** verändert werden:

- vollqualifizierter DB-Zugriff (z. B. DB20.DBW10) als Aktualparameter für FC
- FB- und Multiinstanz-CALL
- Strukturkomponente eines Formalparameters als Operand innerhalb eines FC oder FB
- Strukturkomponente eines Formalparameters als Aktualparameter für FC oder FB

Beim FB-, FC-, Multiinstanz-CALL dürfen VKE oder **AKKU1** und **AKKU2** nicht als zusätzliche (implizite) Parameter verwendet werden.

Das **DI-Register** und das **Adressregister AR2** werden systemseitig für den FB- und Multiinstanz-CALL verwendet und dürfen deshalb innerhalb von FBs nicht verändert werden.

Das **Adressregister AR1** wird von einem Teil der ladbaren Standardbausteine verwendet.

Der Befehl "L P#Parametername" lädt innerhalb eines FB den Adressoffset des angegebenen Parameters, relativ zum **Adressregister AR2**. Um in multiinstanzfähigen FBs den absoluten Offset im Instanzdatenbaustein zu ermitteln, muss zu diesem Wert noch der bereichsinterne Zeiger (nur Adresse) des AR2-Registers addiert werden.

Weitere Information zu den Registern der CPU finden Sie in der Hilfe zur Programmiersprache (KOP/FUP/AWL).

7 Aufbau der Online-Verbindung und CPU-Einstellung

7.1 Aufbau von Online-Verbindungen

Eine Online-Verbindung zwischen PG/PC und CPU ist für das Laden von Anwenderprogrammen/Bausteinen, das Zurückladen von Bausteinen aus der CPU in das PG/PC und für folgende weitere Tätigkeiten erforderlich:

- Testen von Anwenderprogrammen
- Anzeigen und Ändern des Betriebszustands der CPU
- Anzeigen und Einstellen von Uhrzeit und Datum der CPU
- Anzeigen des Baugruppenzustands
- Baustein on-/offline vergleichen
- Hardware diagnostizieren

Um eine Online-Verbindung herstellen zu können, müssen PG/PC und CPU über die MPI Schnittstelle miteinander verbunden sein.

Wird von zwei Programmen (z. B. zwei PGs/PCs oder ein PG/PC mit STEP 7 Lite und STEP 7) auf die selbe CPU zugegriffen, so sollten gelegentlich mit "F5" die Onlineinformationen aktualisiert werden.

STEP 7 Lite sofort "Online"

STEP 7 Lite versucht unmittelbar nach dem Start, eine Online-Verbindung zur CPU aufzubauen.

Wenn **keine** CPU erreicht werden kann, dann bleibt STEP 7 Lite offline. Auch wenn Sie STEP 7 Lite nochmals starten, wird keine Online-Verbindung hergestellt. In diesem Fall müssen Sie die Ursache für den misslungenen Aufbau der Online-Verbindung beseitigen und auf die Schaltfläche "Online/Offline" klicken.

Ohne Online-Verbindung werden keine Synchronisations-Symbole im Projektfenster angezeigt, das CPU-Bedienpanel und die CPU-Sicht im Projektfenster sind nicht anwählbar.

Wechsel zwischen Online und Offline

In der Symbolleiste (und im Menü) befindet sich die Schaltfläche "Online/Offline", mit der Sie den Aufbau bzw. Abbau einer Verbindung zur CPU initiieren können. Die Schaltfläche wird gedrückt dargestellt, wenn eine Verbindung besteht und ist nicht-gedrückt dargestellt, wenn keine besteht. Gelingt der Aufbau einer Online-Verbindung, dann sind die abgeschalteten Fenster wieder anwählbar und die Synchronisations-Symbole werden angezeigt. In der Titelleiste des CPU-Bedienpanels wird die MPI-Adresse der CPU in eckigen Klammern angezeigt.

7.1.1 Passwortschutz für Zugriff auf Zielsysteme

Mit Hilfe des Passwortschutzes können Sie

- das Anwenderprogramm in der CPU mit seinen Daten vor ungewollten Änderungen schützen (Schreibschutz)
- das Know-How, das in Ihrem Anwenderprogramm enthalten ist, bewahren (Leseschutz)
- Online-Funktionen, die den Prozess stören würden, unterbinden

Sie können eine Baugruppe nur dann mit einem Passwort schützen, wenn die Baugruppe diese Funktionalität unterstützt.

Einstellen des Passwortschutzes

Wenn Sie eine Baugruppe mit einem Passwort schützen möchten, müssen Sie die Schutzstufe und das Passwort im Rahmen der CPU-Parametrierung definieren und anschließend die geänderte Parametrierung auf die CPU laden.

Um die CPU entsprechend zu parametrieren, doppelklicken Sie im Projektfenster auf "Hardware" und doppelklicken anschließend auf die CPU auf Steckplatz 2. Die Schutzstufe und das Passwort parametrieren Sie im Abschnitt "Schutz".

Passwort-Abfrage im Betrieb

Wenn für die Ausführung einer Online-Funktion die Eingabe eines Passworts erforderlich ist, wird das Dialogfeld "Passwort eingeben" angezeigt. Durch die Eingabe des korrekten Passworts erhalten Sie die Zugangsberechtigung zur CPU, für die im Rahmen der Parametrierung eine besondere Schutzstufe festgelegt wurde. Sie können dann Online-Verbindungen zur geschützten Baugruppe aufbauen und die zur Schutzstufe gehörigen Online-Funktionen ausführen.

Das Passwort können Sie auch über das erweiterte CPU-Bedienpanel eingeben (anschließend auf Schaltfläche "Anmelden" klicken). Ebenfalls über das erweiterte CPU-Bedienpanel können Sie die über Passwort legitimierte Online-Verbindung wieder abmelden. Ein erneuter Zugriff ist dann nur durch erneute Passwort-Eingabe möglich.

7.2 Anzeigen und Ändern des Betriebszustands

Mit dieser Funktion können Sie beispielsweise die CPU nach Korrektur eines Fehlers wieder in den Betriebszustand "RUN" setzen.

Anzeigen des Betriebszustands im CPU-Bedienpanel

Voraussetzung ist, dass eine Online-Verbindung zur CPU besteht.

Falls das CPU-Bedienpanel nicht geöffnet ist, klicken Sie auf den Doppelpfeil "CPU Online bedienen" über dem Projektfenster.

Der Doppelpfeil nach unten deutet an, dass das Fenster minimiert ist und über dieses Symbol geöffnet werden kann.

Das geöffnete CPU-Bedienpanel zeigt den aktuellen Betriebszustand sowie die aktuelle Stellung des Betriebsartenschalters an der Baugruppe an. Je nach Ausführung des Betriebsartenschalters der CPU wird er als Schlüsselschalter, Kippschalter oder Drehschalter dargestellt. Sie haben ein Abbild der Frontseite der CPU vor sich.

Ändern des Betriebszustands im CPU-Bedienpanel

Über die Schaltflächen RUN und STOP können Sie den Betriebszustand der CPU-Baugruppe ändern. Es sind nur die Schaltflächen aktivierbar, die im aktuellen Betriebszustand angewählt werden können.

7.3 Anzeigen und Einstellen von Uhrzeit und Datum

Gehen Sie folgendermaßen vor:

1. Öffnen Sie das CPU-Bedienpanel (CPU online bedienen).
2. Klicken Sie auf den Doppelpfeil nach rechts. Es öffnet sich die erweiterte Ansicht des CPU-Bedienpanels. Im Abschnitt "Uhrzeit stellen" wird Ihnen die Zeit Ihres PGs/PCs und die Zeit der CPU angezeigt.
 - Falls die CPU die Uhrzeit vom PG/PC übernehmen soll: Aktivieren Sie das Kontrollkästchen "Von PG/PC übernehmen" und klicken anschließend auf die Schaltfläche "Stellen".
 - Falls Sie die Uhrzeit unabhängig vom PG/PC übernehmen möchten: Deaktivieren Sie das Kontrollkästchen "Von PG/PC übernehmen", editieren die Uhrzeit der CPU und klicken anschließend auf die Schaltfläche "Stellen".

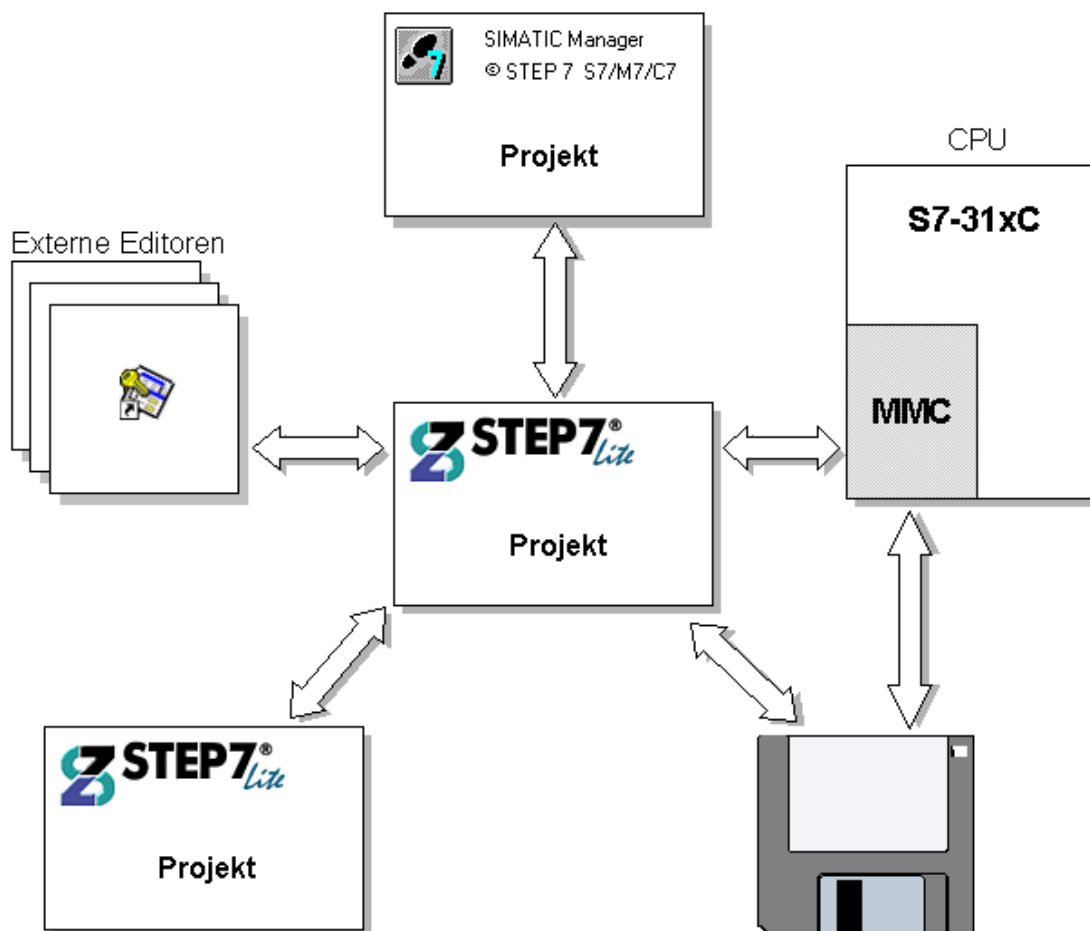
Hinweis

Bei einer Baugruppe ohne integrierte Echtzeituhr werden als Datum 00.00.00 und als Uhrzeit 00:00:00 ausgegeben.

8 Importieren, Exportieren, Speichern unter

8.1 Importieren, Exportieren, Speichern unter

In der folgenden Darstellung werden die möglichen Quellen und Ziele der Import-/Exportfunktionen und der Speicherfunktionen von STEP 7 Lite aufgezeigt.



8.2 Projekte auf Datenträger sichern

Sie haben die Möglichkeit, das gesamte STEP 7 Lite-Projekt z. B. auf Festplatte oder Diskette zu sichern.

Mit dem Menübefehl **Datei > Speichern** werden alle Änderungen, die Sie am Projekt vorgenommen haben, im ursprünglich geöffneten Projekt gesichert.

Mit dem Menübefehl **Datei > Speichern unter** können Sie das geöffnete Projekt mit allen Änderungen unter neuem Namen auf einem anderen Pfad oder anderen Datenträgern sichern.

Hinweis

Beachten Sie, dass der Menübefehl **Bearbeiten > Übernehmen** die Inhalte des Projektes nicht sichert. Der Menübefehl **Bearbeiten > Übernehmen** stellt nach Änderungen die Konsistenz unterschiedlicher geöffneter Sichten in STEP 7 Lite wieder her.

8.3 Projektdaten auf Micro Memory Card (MMC) ablegen

Mit STEP 7 Lite können Sie die Projektdaten Ihres STEP 7 Lite Projektes auf der SIMATIC Micro Memory Card (MMC) einer CPU 31xC ablegen. Sie haben damit den Vorteil auch mit Programmiergeräten, die das Projekt nicht gespeichert haben, auf die Projektdaten zugreifen zu können.

Welche Projektdaten können auf der MMC abgelegt werden?

In STEP 7 Lite können Sie folgende Projektdaten in folgenden Dateiformaten auf einer MMC ablegen:

- Das gesamte Projekt als *.k7p-Datei
- Ausgewählte Bausteine und die Symboltabelle als "S7Lite-Exportdatei" (*.k7e)
- Alle Bausteine des Anwenderprogramms als *.awl-Datei
- Die Symbolliste als *.sdf-Datei

Voraussetzungen

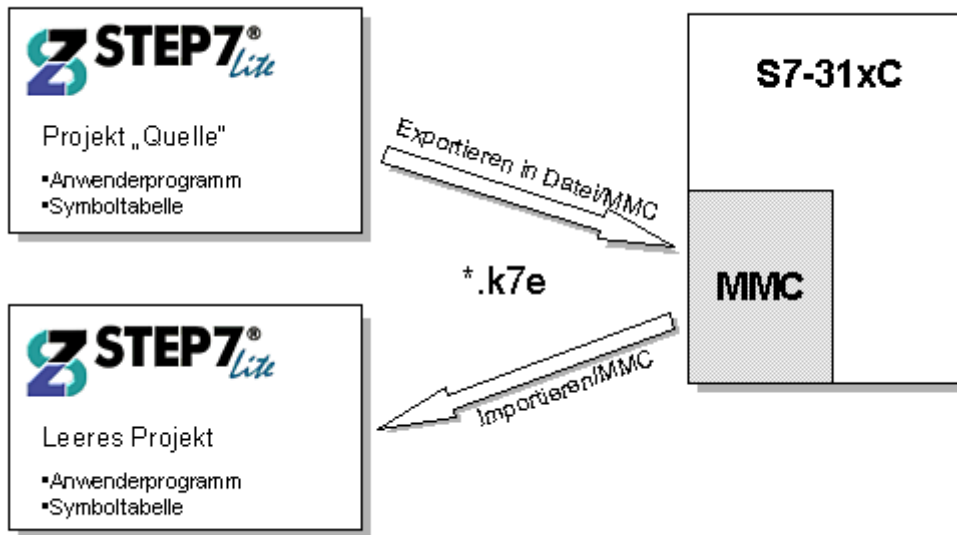
Das Ablegen von Projektdaten auf einer MMC ist nur möglich, wenn sich die MMC im Schacht der CPU 31xC befindet und zu dieser eine Onlineverbindung besteht.

Die Größe der MMC muss so bemessen sein, dass die abgelegten Daten auf dieser Platz finden.

Vorgehensweise zum Speichern des gesamten Projekts

1. Wählen Sie den Menübefehl **Datei > Speichern unter**.
2. Wählen Sie im angezeigten Dialogfeld das Register "Memory Card".
3. Tragen Sie im Feld "Dateiname" einen spezifischen Dateinamen ohne Namensweiterung ein.
4. Wählen Sie in der Klappliste "Dateityp" den Dateityp "Projekte (*.k7p)" aus.
5. Klicken Sie auf die Schaltfläche "Speichern".

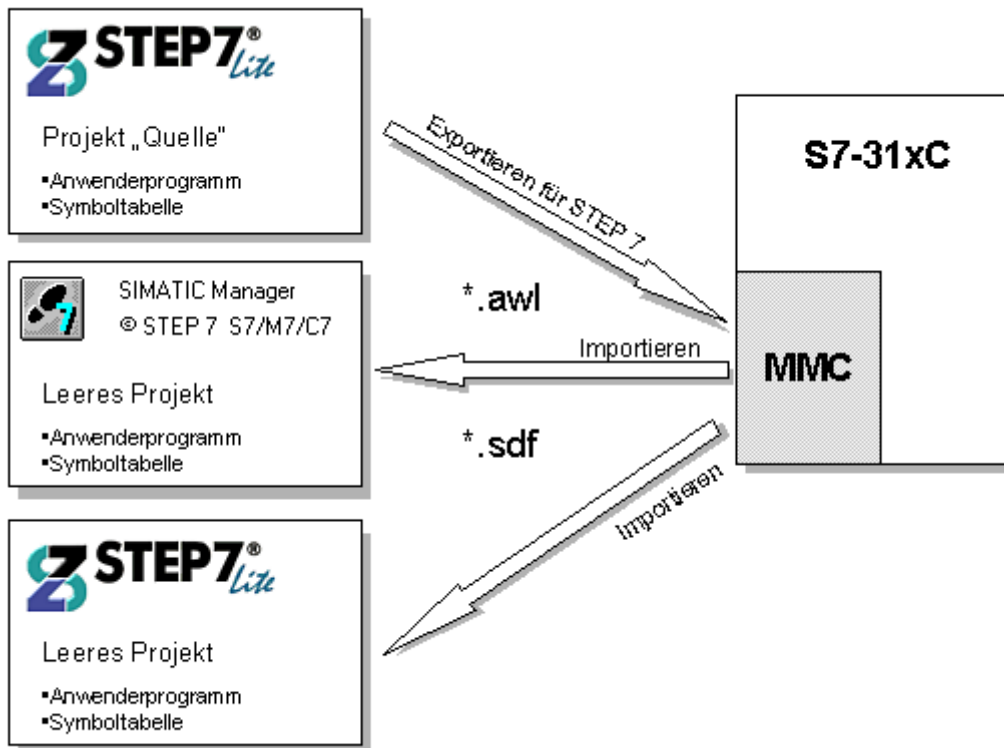
Vorgehensweise zum Speichern ausgewählter Bausteine und der Symboltabelle als "S7Lite-Exportdatei"



1. Wählen Sie den Menübefehl **Datei > Exportieren > In Datei**.
2. Wählen Sie im angezeigten Dialogfeld das Register "Memory Card".
3. Klicken Sie auf die Schaltfläche "Exportieren".

Vorgehensweise zum Speichern als *.awl -Datei und *.sdf -Datei

Das Ablegen der Projektdaten als *.awl-Datei und als *.sdf-Datei hat den Vorteil, dass zukünftige Versionen von STEP 7 diese Projektdaten ebenfalls importieren können.



1. Markieren Sie das Symbol "Programm" bzw. das Symbol "Symboltabelle" im Projektfenster.
(Falls Sie beide Elemente auf der MMC möchten, so markieren Sie das zweite Element bei gleichzeitiger Betätigung der CTRL-Taste).
2. Wählen Sie den Menübefehl Datei > Exportieren > Für STEP 7
3. Wählen Sie das Register "Memory Card"
4. Wählen Sie den Namen der Export-Dateien ohne Namensweiterung. Die zu erzeugenden Dateinamen werden mit der Pfadangabe im Feld "erzeugte Dateien:" angezeigt.
5. Klicken Sie auf die Schaltfläche "Exportieren".

8.4 Micro Memory Card als Datenträger verwenden

Die SIMATIC Micro Memory Card (MMC) einer CPU 31xC lässt sich mit STEP 7 Lite wie ein ganz normaler externer Datenträger einsetzen. Unter der Voraussetzung, dass die MMC ausreichend groß gewählt wurde, lassen sich alle im Datei-Explorer des Betriebssystems sichtbaren Dateien auf die MMC übertragen. Sie können somit an Ihrer Anlage zusätzliche Zeichnungen, Serviceanweisungen, Funktionsbeschreibungen etc. für andere Mitarbeiter zur Verfügung stellen.

Vorgehensweise: Dateien auf die MMC übertragen

1. Öffnen Sie die Sicht "Memory Card" im Register "Online-CPU" des Projektfensters.
2. Wählen Sie den Menübefehl Datei > Dateien übertragen > Auf Memory Card
3. Navigieren Sie im Auswahldialog zu den Dateien, die Sie auf MMC speichern wollen (Klappliste "Suchen in").
4. Markieren Sie in der angezeigten Liste die Dateien, die Sie auf MMC speichern wollen.
5. Klicken Sie auf die Schaltfläche "Öffnen".

Vorgehensweise: Dateien auf der MMC ins Dateisystem übertragen

1. Öffnen Sie die Sicht "Memory Card" im Register "Online-CPU" des Projektfensters.
2. Wählen Sie den Menübefehl Datei > Dateien übertragen > Auf PG
3. Wählen Sie im angezeigten Register "Memory Card" die Dateien aus, die Sie auf PG übertragen wollen.
4. Klicken Sie auf die Schaltfläche "Übertragen".
5. Wählen Sie im Folgedialog den Ordner, in den die Dateien übertragen werden sollen.
6. Klicken Sie auf die Schaltfläche "OK".

8.5 Projektdaten zwischen STEP 7 Lite und STEP 7 austauschen

Was kann nach STEP 7 exportiert, was von STEP 7 importiert werden?

Zwischen STEP 7 Lite und STEP 7 können folgende Projektdaten ausgetauscht werden:

- Das komplette Programm als *.awl -Datei.
- Die gesamte Symboltabelle als *.sdf -Datei.

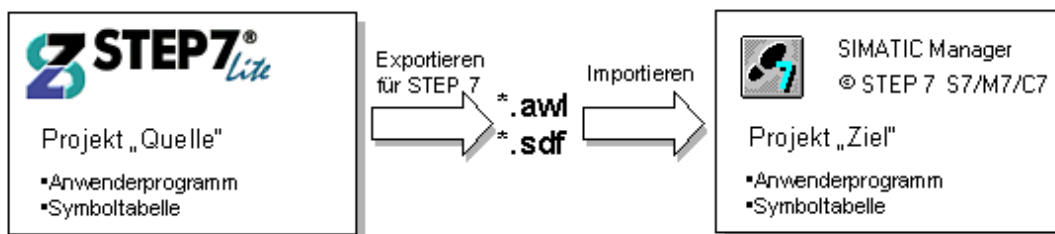
Beim Exportieren exportiert STEP 7 Lite alle Bausteine des Programms bzw. die gesamte Symboltabelle. Beim Importieren aus STEP 7 werden alle Bausteine der *.awl -Datei und alle Symbole der *.sdf -Datei importiert.

Voraussetzungen

Beim Export und Import eines Programms oder einer Symboltabelle sollten in STEP 7 Lite alle Objekte geschlossen sein.

Vorgehensweise beim Exportieren nach STEP 7

Exportieren Sie mit STEP 7 Lite das komplette Anwenderprogramm als *.awl -Datei, bzw. die gesamte Symboltabelle als *.sdf -Datei. Anschließend muss die *.awl -Datei mit der Funktion **Einfügen > Externe Quellen** und die *.sdf -Datei mit der Funktion **Tabelle > Importieren** in das STEP 7-Projekt importiert werden.



Gehen Sie beim Exportieren nach STEP 7 wie folgt vor:

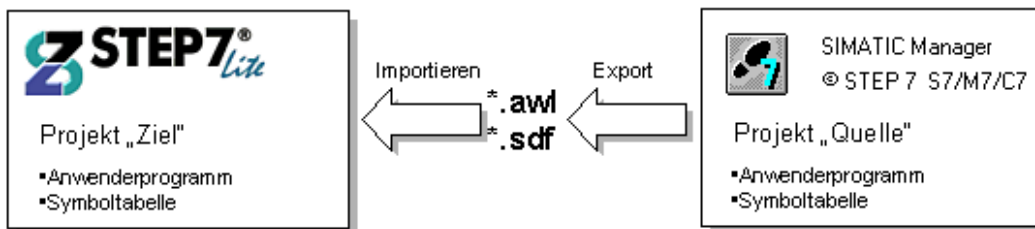
1. Markieren Sie das Symbol "Programm" bzw. das Symbol "Symboltabelle" im Projektfenster.
(Falls Sie beide Elemente exportieren möchten, so markieren Sie das zweite Element bei gleichzeitiger Betätigung der CTRL-Taste).
2. Wählen Sie den Menübefehl Datei > Exportieren > Für STEP 7
3. Wählen Sie im angezeigten Dialogfeld das Zielverzeichnis aus.
In welcher Form die angewählten Elemente exportiert werden, hängt von den gewählten Option ("absolute Bezeichner" oder "symbolische Bezeichner") ab.

4. Wählen Sie den Namen der Export-Dateien ohne Namensweiterung. Die zu erzeugenden Dateinamen werden mit der Pfadangabe im Feld "erzeugte Dateien:" angezeigt.
5. Klicken Sie auf die Schaltfläche "Exportieren".
6. Die Beschreibung zum Importieren in das STEP 7-Projekt entnehmen Sie der Hilfe zu STEP 7 (Themen "Einfügen externer Quellen" und "Importieren einer Symboltabelle").

Vorgehensweise beim Importieren von STEP 7

Exportieren Sie mit STEP 7 das komplette Anwenderprogramm oder einzelne Bausteine mit der Funktion "Quelle exportieren" als *.awl -Datei. Gleiches gilt für die Symboltabelle, diese muss in STEP 7 mit der Funktion "Tabelle > Exportieren" als *.sdf -Datei exportiert werden.

Importieren Sie anschließend die *.awl -Datei und die *.sdf -Datei in STEP 7 Lite.



Gehen Sie beim Importieren von STEP 7 wie folgt vor:

1. Exportieren Sie in STEP 7 die Quellen und die Symboltabelle. Die Beschreibung zum Exportieren von STEP 7-Quellen und einer STEP 7-Symboltabelle entnehmen Sie der Hilfe zu STEP 7 (Themen "Exportieren von Quellen" und "Exportieren einer Symboltabelle").
2. Wählen Sie in STEP 7 Lite den Menübefehl **Datei > Importieren**
3. Wählen Sie im angezeigten Dialogfeld das Quellverzeichnis und die entsprechenden *.awl -Datei und die *.sdf -Datei aus.
4. Bestimmen Sie mit dem Optionskästchen "Objekte überschreiben", OB bereits existierende Bausteine überschrieben werden sollen oder nicht.
5. Klicken Sie auf die Schaltfläche "Importieren" (im folgenden Dialog wird eine Liste mit möglichen Importfehlern angezeigt).

Hinweis

Sie können auch Projektdaten aus STEP 5 importieren. Verwenden Sie hierzu den im Lieferumfang beinhalteten Konverter "S5 Datei konvertieren" und die dazugehörige Onlinehilfe.

Fahren Sie nach der Konvertierung mit dem obig beschriebenen Schritt 2 fort. Als zu importierende Dateien sind die vom Konverter erzeugten Dateien <Name>AC.AWL und <Name>S7.SEQ auszuwählen.

8.6 Projektdaten für externe Editoren exportieren

8.6.1 Datenformat für den Import/Export einer Symboltabelle

Das Dateiformat System Data Format (SDF) kann in die Symboltabelle importiert bzw. aus ihr exportiert werden:

- SDF-Dateien können Sie in Microsoft Access öffnen, bearbeiten und wieder speichern.
- Benutzen Sie für den Import und Export von Daten der Microsoft-Applikation ACCESS das SDF-Format.
- Wählen Sie in ACCESS als Dateiformat "Text (mit Trennzeichen)".
- Benutzen Sie als Texttrennzeichen das doppelte Hochkomma (").
- Benutzen Sie als Feldtrennzeichen das Komma (,).

System Data Format (SDF)

Dateityp:	*.SDF
Aufbau:	Zeichenketten in Hochkommata, Teile durch Kommata getrennt
Beispiel:	"grün_phase_fußg","T 2","TIMER","Dauer der Grünphase für Fußgänger" "rot_fußg","A 0.0","BOOL","Rot für Fußgänger"

Zum Öffnen einer SDF-Datei in Microsoft Access wählen Sie das Dateiformat "Text (mit Trennzeichen)". Geben Sie als Texttrennzeichen das doppelte Hochkomma (") und als Feldtrennzeichen das Komma (,) an.

8.6.2 Texte mehrsprachig verwalten

STEP 7 Lite bietet die Möglichkeit, Texte, die in einem Projekt einsprachig abgelegt sind, zu exportieren, übersetzen zu lassen, wieder zu importieren und in der übersetzten Sprache anzeigen zu lassen.

Die folgenden Arten von Texten können mehrsprachig verwaltet werden.

- Kommentare und Titel
 - Titel und Kommentare zu Stationen und Baugruppen
 - Kategorietitel
 - Bausteintitel und Bausteinkommentare
 - Netzwerktitel und Netzwerkkommentare
 - Zeilenkommentare aus AWL-Programmen und Variablentabellen
 - Kommentare aus Symboltabellen, Variablendeklarationstabellen, anwenderdefinierten Datentypen und Datenbausteinen
- Anzeigetexte (nicht mit STEP 7 Lite)
 - Meldungstexte
 - System-Textbibliotheken

Export

Der Export wird für alle Texttypen ausgeführt, die zum angewählten Objekt gehören. Für jeden Texttyp wird eine Exportdatei erzeugt. Diese enthält je eine Spalte für die Quell- und für die Zielsprache. Die Texte in der Quellsprache dürfen nicht geändert werden.

Import

Beim Import wird der Inhalt der Spalten für die Zielsprache (rechte Spalte) in das angewählte Projekt übernommen. Es werden dabei nur die Texte übernommen, für die in der Spalte für die Quellsprache ein Übereinstimmung mit einem vorhandenen Text festgestellt wird.

Sprachwechsel

Beim Sprachwechsel können alle Sprachen ausgewählt werden, die beim Import in das angewählte Projekt angegeben wurden.

Sprache löschen

Beim Löschen einer Sprache werden alle Texte in dieser Sprache aus der internen Datenhaltung gelöscht.

Es sollte immer eine Sprache als Bezugssprache in Ihrem Projekt vorhanden sein. Das kann beispielsweise Ihre Landessprache sein. Diese Sprache sollten Sie nicht löschen. Beim Exportieren und Importieren stellen Sie diese Bezugssprache immer als Quellsprache ein. Die Zielsprache stellen Sie nach Wunsch ein.

Prinzipielle Vorgehensweise



8.6.2.1 Texttypen mehrsprachig verwalteter Texte

Für jeden Texttyp wird beim Export eine eigene Datei angelegt. Diese hat den Texttyp als Namen und das Exportformat als Endung (Texttyp.Format: z. B. SymbolComment.CSV). Dateien die nicht der Namenskonvention genügen, können nicht als Quelle oder Ziel angegeben werden.

Die übersetzbaren Texte innerhalb eines Projekts sind in folgende Texttypen aufgeteilt:

Texttyp	Bedeutung
HardwareTitle	Titel von Stationen und Baugruppen
HardwareComment	Kommentare zu Stationen und Baugruppen
CategoryTitle	Kategorietitel
VariableComment	Zeilenkommentare in Variablendeklarationstabellen
BlockTitle	Bausteintitel
BlockComment	Bausteinkommentar
NetworkTitle	Netzwerktitel
NetworkComment	Netzwerkkommentar
LineComment	Zeilenkommentar in AWL
InterfaceComment	Var_Section-Kommentar (Deklarationstabelle in Codebausteinen) UDT-Kommentar (anwenderdefinierte Datentypen) Datenbausteinkommentar
SymbolComment	Symbolkommentar

8.6.2.2 Aufbau der Export-Datei

Die Export-Datei ist prinzipiell folgendermaßen aufgebaut:

Beispiel:

\$ Languages		
7(1) Deutsch (Deutschland)	9(1) Englisch (USA)	
\$ Typ(SymbolComment)		
Erste zu übersetzende Zeichenfolge	Übersetzung	
Zweite zu übersetzender Zeichenfolge	Übersetzung	

Quellsprache

Zielsprache

Grundsätzlich gilt:

1. Nicht geändert, überschrieben oder gelöscht werden dürfen:
 - die Felder mit einem vorangestellten "\$_" (das sind Schlüsselwörter)
 - die Nummern für die Sprache (im Beispiel oben: 7(1) für die Quellsprache Deutsch (Deutschland) und 9(1) für die Zielsprache Englisch)
2. In der Datei sind immer nur Texte eines Texttyps hinterlegt. Im Beispiel ist der Texttyp Symbolkommentar (\$_Typ(SymbolComment)). Die Regeln für den Übersetzer, der in dieser Datei editiert, sind im einleitenden Text der Export-Datei selbst hinterlegt.
3. Zusatzinformationen zu den Texten oder Kommentare müssen immer vor der Typdefinition (\$_Typ...) oder hinter der letzten Spalte stehen.

Hinweis

Wenn die Spalte für die Zielsprache mit "\$_Undefined" überschrieben ist, dann wurde beim Exportieren keine Zielsprache angegeben. Zur besseren Übersichtlichkeit können Sie diesen Text ersetzen durch die Zielsprache; z. B. durch "Englisch". Beim Importieren müssen Sie dann die vorgeschlagene Zielsprache kontrollieren und ggf. neu wählen.

Format der Exportdatei

Die Exportdateien werden im CSV-Format abgelegt. Für die Bearbeitung mit Excel muss beachtet werden, dass EXCEL eine CSV-Datei nur dann korrekt öffnet, wenn dazu der Öffnen-Dialog verwendet wird. **Beim Öffnen einer CSV-Datei durch Doppelklick im Explorer wird diese Datei häufig unbrauchbar.** Durch folgendes Vorgehen kann das Arbeiten mit EXCEL erleichtert werden:

1. Öffnen Sie die Exportdateien mit EXCEL.
2. Speichern Sie die Dateien als XLS-Dateien.
3. Übersetzen Sie die Texte in den XLS-Dateien.
4. Speichern Sie die XLS-Dateien mit EXCEL im CSV-Format.

Achtung

Die Exportdateien dürfen nicht umbenannt werden!

8.6.2.3 Schritte zum Verwalten mehrsprachiger Texte

Exportieren von mehrsprachig verwalteten Texten

Voraussetzung

Es dürfen keine Exportdateien geöffnet sein.

Vorgehensweise

1. Wählen Sie den Menübefehl **Extras > Texte mehrsprachig verwalten > Exportieren**.
2. Bestimmen Sie im Dialog "Exportieren" Quell- und Zielsprache und Texttypen
Tipp 1: In der Regel alle Texttypen wählen. Für Nachübersetzungen (z. B. von geänderten Kommentaren) nur die relevanten Texttypen.
Tipp 2: Wenn Sie das Projekt in **mehrere Sprachen übersetzen** lassen wollen: Lassen Sie das Feld für die Zielsprache leer! In der Exportdatei wird dann für die Zielsprache "\$_Undefined" eingetragen. Kopieren Sie die Datei für mehrere Übersetzer und ändern Sie jeweils den Text "\$_Undefined" in die zu übersetzende Sprache. Beim Importieren der übersetzten Dateien müssen Sie dann explizit die Zielsprache wählen!
3. Falls Exportdateien (Exportziel) bereits vorhanden sind, können Sie im Folgedialog wählen, OB sie erweitert oder überschrieben werden sollen. Beim Erweitern bleiben die bereits übersetzten Texte erhalten und die neuen (noch zu übersetzenden) Texte werden hinzugefügt.
4. Quittieren Sie den Dialog mit "OK" und senden die erzeugten Textdateien an den Übersetzer.

Übersetzen von mehrsprachig verwalteten Texten

Wenn die Exportdateien mit MS-Excel bearbeitet werden, müssen sie über den Menübefehl **Datei > Öffnen** von MS-Excel geöffnet werden und nicht über Doppelklick auf die Exportdatei (CSV-Format), da sonst die Datei fehlerhaft geöffnet wird.

- Übersetzen Sie die Texte in der zweiten Spalte unter dem Schlüsselwort "\$_Typ(...)".

Achtung

Beachten Sie bitte, dass EXCEL bestimmte Zeichen als Formel interpretiert und deshalb beim Speichern die Quellsprache verändert. Die übersetzten Texte werden dann nicht importiert.

Importieren von mehrsprachig verwalteten Texten

1. Wählen Sie den Menübefehl **Extras > Texte mehrsprachig verwalten > Importieren**.
2. Bestimmen Sie im Dialog "Importieren" die Importquelle und das Format
Falls Sie beim Exportieren keine Zielsprache eingegeben haben, werden Sie durch einen Dialog aufgefordert, die Zielsprache zu wählen.
3. Quittieren Sie den Dialog mit "OK"
4. Werten Sie bei aufgetretenen Fehlern das angezeigte Protokoll aus.

Wählen der Sprache

1. Wählen Sie den Menübefehl **Extras > Texte mehrsprachig verwalten > Sprachwechsel**.
2. Wählen Sie im aufgeblendeten Sprachwechsel-Dialog die gewünschte Sprache für die Texttypen.
3. Quittieren Sie den Dialog mit "OK".

Löschen einer Sprache

1. Wählen Sie den Menübefehl **Extras > Texte mehrsprachig verwalten > Sprache löschen**.
2. Wählen Sie im aufgeblendeten Dialogfeld die gewünschte Sprache aus und geben Sie an, OB auch Titel und Kommentare gelöscht werden soll.
3. Quittieren Sie den Dialog mit "OK".

8.6.2.4 Tipps zum Übersetzen

Optimieren der Vorlage für die Übersetzung

Sie können das sprachliche "Quellmaterial" für die Übersetzung bereinigen, indem Sie unterschiedliche Begriffe bzw. Ausdrücke zusammenführen.

Beispiel

Vor der Bereinigung (Export-Datei):

\$_Languages		
7(1) Deutsch (Deutschland)	7(1) Deutsch (Deutschland)	
\$_Typ(SymbolComment)		
Auto-Freig.		
Automatische Freigabe		
Auto-Freigabe		

Quellsprache

Zielsprache

Zusammenführen zu einem Ausdruck:

\$_Languages		
7(1) Deutsch (Deutschland)	7(1) Deutsch (Deutschland)	
\$_Typ(SymbolComment)		
Auto-Freig.	Auto-Freigabe	
Automatische Freigabe	Auto-Freigabe	
Auto-Freigabe	Auto-Freigabe	

Quellsprache

Zielsprache

Nach der Bereinigung (d. h. nach Import und anschließendem Export):

\$_Languages		
7(1) Deutsch (Deutschland)	7(1) Deutsch (Deutschland)	
\$_Typ(SymbolComment)		
Auto-Freigabe	Auto-Freigabe	

Quellsprache

Zielsprache

Optimierung des Übersetzungsvorgangs

Bei Projekten, deren Aufbau und Texte ähnlich einem Vorgängerprojekt sind, können Sie den Übersetzungsvorgang optimieren.

Insbesondere für Projekte, die durch Kopieren und anschließendes Anpassen entstanden sind, ist das im folgenden beschriebene Vorgehen empfehlenswert.

Voraussetzung

Es liegt ein bereits übersetztes Exportziel (CSV-Dateien) vor.

Vorgehensweise

1. Kopieren Sie die Exportdateien in das Projektverzeichnis für das zu übersetzende neue Projekt.
2. Öffnen Sie das neue Projekt und exportieren Sie die Texte (Menübefehl **Extras > Texte mehrsprachig verwalten > Exportieren**).
Weil das Exportziel bereits existiert, werden Sie gefragt, OB das Exportziel erweitert oder überschrieben werden soll.
3. Klicken Sie auf die Schaltfläche "Erweitern".
4. Lassen Sie die Exportdateien übersetzen (es müssen nur die neuen Texte übersetzt werden).
5. Importieren Sie anschließend die übersetzten Dateien.

9 Laden in CPU und in PG

9.1 Laden aus dem PG in die CPU

9.1.1 Voraussetzungen für das Laden

Voraussetzungen zum Laden in die CPU

- Es besteht eine Verbindung zwischen Ihrem PG und der CPU über die MPI-Schnittstelle.
- Der Zugriff auf die CPU ist möglich.
- Das zu ladende Programm ist fehlerfrei übersetzt.
- Die CPU muss in einem Betriebszustand sein, bei dem das Laden zulässig ist (STOP oder RUN-P).
Beachten Sie jedoch beim Laden im RUN-P-Zustand, dass das Programm bausteinweise übertragen wird. Überschreiben Sie dabei ein altes CPU-Programm, können Konflikte auftreten, z. B. wenn sich Bausteinparameter geändert haben. Die CPU geht dann während der Bearbeitung des Zyklus in den Betriebszustand STOP. Es wird daher empfohlen, die CPU vor dem Laden in den Betriebszustand "STOP" zu schalten.
- Vor dem Laden des Anwenderprogramms sollten Sie die CPU urlöschen, um sicherzustellen, dass sich keine "alten" Bausteine auf der CPU befinden.

Betriebszustand STOP

Setzen Sie den Betriebszustand von RUN auf STOP, bevor Sie

- das gesamte oder Teile des Anwenderprogramm auf die CPU laden,
- ein Urlöschen der CPU veranlassen,
- den Anwenderspeicher komprimieren.

Neustart (Warmstart) (Übergang in den Betriebszustand RUN)

Wenn Sie vom Zustand "STOP" aus einen Neustart (Warmstart) veranlassen, wird das Programm neu gestartet und zunächst im Betriebszustand "ANLAUF" das Anlauf-Programm (OB 100) abgearbeitet. Ist der Anlauf erfolgreich, wechselt die CPU in den Betriebszustand RUN. Ein Neustart (Warmstart) ist erforderlich nach:

- Urlöschen der CPU,
- Laden des Anwenderprogramms im STOP-Zustand.

9.1.2 Was wird wann geladen?

Im Menü "Datei" finden Sie den Menübefehl "Laden in CPU". Im folgenden sind die Möglichkeiten aufgelistet, welche Objekte Sie unter welchen Voraussetzungen laden können.

Was wollen Sie laden?	Voraussetzung	Bemerkungen
Hardware-Konfiguration	Die Hardware-Konfiguration ist gerade geöffnet oder Das Objekt 'Hardware' ist markiert	Markieren können Sie das Objekt 'Hardware' <ul style="list-style-type: none"> • im Projektfenster • in der Sicht 'Projektübersicht' (nach Doppelklick auf das Symbol 'Projekt' im Projektfenster)
Bausteine	Der Baustein ist gerade im Bausteineditor geöffnet oder Ein oder mehrere Bausteine sind markiert	Markieren könne Sie Bausteine <ul style="list-style-type: none"> • im Projektfenster • in der Sicht 'Projektübersicht' (nach Doppelklick auf das Symbol 'Projekt' bzw. nach Doppelklick auf das Symbol 'Programm' im Projektfenster)
Ein oder mehrere beliebige Objekte (unabhängig von der angezeigten Sicht)	Das Objekt ist im Projektfenster markiert	Wenn das Projekt markiert ist, werden alle Bausteine und die Hardware-Konfiguration geladen. Wenn das Programm markiert ist, werden alle Bausteine des Programms geladen

Wenn sich bereits Bausteine mit identischen Bausteinnummern in der CPU befinden, können Sie im Folgedialog entscheiden, OB diese Bausteine überschrieben werden sollen oder nicht.

Wenn das zu ladende Objekt (z. B. ein Baustein oder die Hardware-Konfiguration) in STEP 7 Lite geöffnet ist, wird dieser aktuelle sichtbare Stand geladen (nicht die gespeicherten Bausteine oder die gespeicherte Hardware-Konfiguration)! Achten Sie aus Konsistenzgründen darauf, dass Sie den Stand, den Sie laden wollen, zuvor speichern!

9.1.3 Unterschied zwischen Speichern und Laden von Bausteinen

Hinweis zu Bausteinänderungen – erst speichern dann laden

Um neu erstellte Bausteine bzw. Änderungen im Anweisungsteil von Codebausteinen, in Deklarationstabellen oder bei Datenwerten in Datenbausteinen zu sichern, müssen Sie den entsprechenden Baustein abspeichern (Menübefehl **Datei > Speichern**). Damit wird das ganze Projekt gespeichert.

Änderungen, die Sie im Editor durchführen und mit dem Menübefehl **Datei > Laden in CPU** in die CPU transferieren, z. B. zum Testen kleiner Änderungen, sollten auf jeden Fall vor dem Beenden des Editors auch auf der Festplatte des PG gespeichert werden. Andernfalls haben Sie anschließend unterschiedliche Stände Ihres Anwenderprogramms in der CPU und im PG.

Allgemein ist zu empfehlen, immer zuerst Änderungen zu speichern und anschließend zu laden.

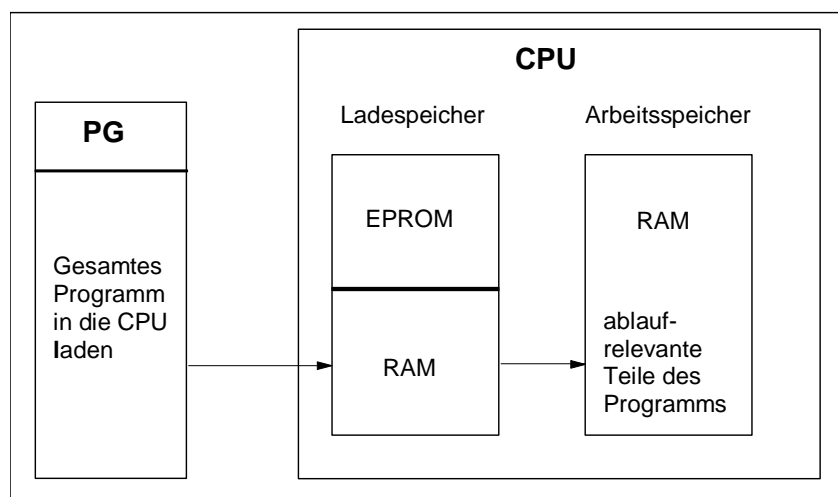
9.1.4 Lade- und Arbeitsspeicher in der CPU

Nach Abschluss der Konfiguration, Parametrierung und Programmerstellung können Sie komplette Anwenderprogramme oder einzelne Bausteine auf die angeschlossene CPU übertragen. Für den Test einzelner Bausteine müssen Sie zumindest einen OB sowie die darin aufgerufenen FBs und FCs und die benutzten DBs laden.

Die Hardware-Konfiguration müssen Sie ebenfalls auf das Zielsystem übertragen.

Zusammenspiel von Lade- und Arbeitsspeicher der CPU

Das gesamte Anwenderprogramm wird in den Ladespeicher geladen, die ablaufrelevanten Teile des Programms auch in den Arbeitsspeicher.



Ladespeicher der CPU

- Der Ladespeicher dient zur Aufnahme des Anwenderprogramms ohne Symboltabelle und Kommentare (diese bleiben im Speicherbereich des PG).
- Bausteine, die als nicht ablaufrelevant gekennzeichnet sind, werden ausschließlich in den Ladespeicher aufgenommen.
- Beim Ladespeicher kann es sich, abhängig von der CPU, um RAM-, ROM- oder EPROM-Speicher handeln.
- Bei S7-300 kann der Ladespeicher außer einem integrierten RAM- auch einen integrierten EEPROM-Anteil haben (z. B. CPU312 IFM und CPU314 IFM).

Arbeitsspeicher der CPU

Der Arbeitsspeicher (integriertes RAM) dient zur Aufnahme der für den Programmablauf relevanten Teile des Anwenderprogramms.

Mögliche Vorgehensweisen beim Laden

Über die Ladefunktion können Sie das Anwenderprogramm oder ladbare Objekte (z. B. Bausteine) in das Zielsystem laden. Ist ein Baustein schon im RAM der CPU vorhanden, so werden Sie beim Laden gefragt, OB der Baustein überschrieben werden soll oder nicht.

- Die ladbaren Objekte können Sie im Projektfenster markieren und dann den Menübefehl **Datei > Laden in CPU** ausführen.

Umgekehrt können Sie sich über die Ladefunktion aktuelle Inhalte von Bausteinen aus dem RAM-Ladespeicher der CPU in Ihr PG laden.

9.1.5 Lademöglichkeiten abhängig vom Ladespeicher

Aus der Aufteilung des Ladespeichers in einen RAM- und einen EEPROM-Bereich ergeben sich Konsequenzen für die Möglichkeiten beim Laden Ihres Anwenderprogramms bzw. von Bausteinen Ihres Anwenderprogramms. Um die Daten in die CPU zu laden, bestehen folgende Möglichkeiten:

Ladespeicher	Lademöglichkeiten
RAM	Laden und Löschen von einzelnen Bausteinen
	Laden und Löschen eines gesamten Anwenderprogrammes
	Nachladen von einzelnen Bausteinen
EPROM integriert oder steckbar	Laden von gesamten Anwenderprogrammen
EPROM steckbar	Laden von gesamten Anwenderprogrammen

Laden in das RAM über Online-Verbindung

In der CPU haben Sie bei Spannungsausfall keine Datensicherheit, falls das RAM ungepuffert ist. Die Daten im RAM gehen in diesem Fall verloren.

Speichern auf EPROM-Memory Card

Bausteine bzw. das Anwenderprogramm werden auf eine EPROM-Memory Card gespeichert, die sich im Schacht der CPU befinden muss.

Die gespeicherten Daten bleiben bei Spannungsausfall und beim Utlöschen der CPU erhalten. Der Inhalt des EPROMs wird nach Utlöschen der CPU und nach Netzausfall mit ungepuffertem RAM-Speicher bei Spannungswiederkehr wieder in den RAM-Bereich des Speichers der CPU kopiert.

Speichern im integrierten EPROM

Für die CPU 312 gibt es noch die Möglichkeit, den Inhalt des RAMs in das integrierte EPROM zu speichern. Die Daten im integrierten EPROM bleiben bei Spannungsausfall remanent. Der Inhalt des integrierten EPROMs wird nach Netzausfall mit ungepuffertem RAM-Speicher bei Spannungswiederkehr und nach Utlöschen der CPU wieder in den RAM-Bereich des Speichers der CPU kopiert.

9.1.6 Bausteine und Konfiguration in CPU laden und auf Memory Card speichern

9.1.6.1 Nachladen von Bausteinen in die CPU

Sie können Bausteine, die im Ladespeicher (RAM) bzw. Arbeitsspeicher der CPU vorhanden sind, mit einer neuen Version überschreiben (nachladen). Dabei wird der bis dahin vorhandene Stand überschrieben.

Die Vorgehensweise beim Nachladen entspricht der des Ladens von Bausteinen. Es erfolgt lediglich eine Abfrage, OB der vorhandene Baustein überschrieben werden soll.

Ein im EPROM gespeicherter Baustein kann nicht gelöscht werden, wird aber bei der Nachladefunktion für ungültig erklärt. Der ihn ersetzende Baustein wird in das RAM geladen. Dabei entstehen im Lade- oder Arbeitsspeicher Lücken. Falls diese Lücken dazu führen, dass keine neuen Bausteine mehr ladbar sind, sollten Sie die Speicher komprimieren.

Achtung

Im Fall eines Netzausfalls mit ungepuffertem RAM-Speicher und Spannungswiederkehr oder beim Utlöschen der CPU werden wieder die "alten" Bausteine aus dem EPROM gültig und geladen!

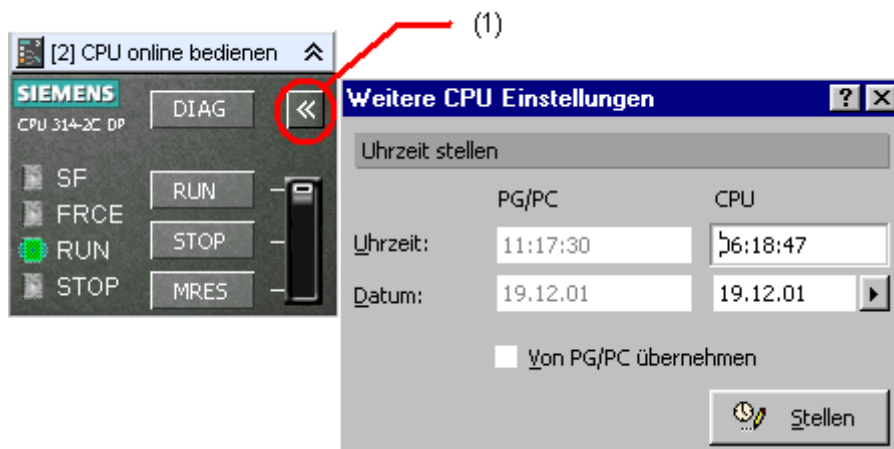
9.1.6.2 Speichern geladener Bausteine auf integriertem EPROM bzw. auf S7-Memory Card in der CPU

Bei CPU-Baugruppen, die ein integriertes EPROM besitzen (z. B. CPU 312), können Sie Bausteine vom Ladespeicher (RAM) in das integrierte EPROM speichern, um die Daten bei Spannungsausfall oder Umlöschen nicht zu verlieren.

Ebenso können Sie bei einer CPU mit gesteckter S7-Memory Card (5V-FEPROM) die Bausteine vom Ladespeicher (RAM) auf die S7-Memory Card übertragen.

Gehen Sie folgendermaßen vor:

1. Laden Sie die Bausteine in die CPU.
2. Erweitern Sie das CPU-Bedienpanel in der STEP 7 Lite-Benutzeroberfläche



(1): Klicken, um CPU-Bedienpanel zu erweitern/zu reduzieren

3. Klicken Sie im erweiterten CPU-Bedienpanel auf die Schaltfläche "RAM > ROM" (unter "weitere Funktionen" im CPU-Bedienpanel).

9.1.6.3 Laden einer Konfiguration in ein Zielsystem

Tipp

Vor dem Laden sollten Sie bei geöffneter HW-Konfiguration mit dem Menübefehl **Extras > Konsistenz prüfen** die Fehlerfreiheit Ihrer Stationskonfiguration prüfen. STEP 7 Lite prüft dann, OB aus der aktuellen Konfiguration ladbare Systemdaten erzeugt werden können. Bei der Konsistenzprüfung werden Ihnen vorliegende Fehler in einer Ausgabeleiste angezeigt.

Voraussetzungen zum Laden

- Das Programmiergerät ist über ein MPI-Kabel an die MPI-Schnittstelle der CPU angeschlossen.
- Bei vernetzter Anlage (Programmiergerät ist an ein Subnetz angeschlossen): Sämtliche Baugruppen eines Subnetzes müssen unterschiedliche Teilnehmeradressen besitzen.
- Die erstellte Konfiguration entspricht dem tatsächlichen Stationsaufbau. Eine Konfiguration kann nur dann in die Station geladen werden, wenn sie konsistent und fehlerfrei ist. Nur dann können Systemdatenbausteine (SDBs) erzeugt werden, die wiederum in die Baugruppen geladen werden.

Vorgehensweise

1. Markieren Sie im Projektfenster den Eintrag "Hardware"
2. Wählen Sie den Menübefehl **Datei > Laden in CPU** bzw. drücken Sie die rechte Maustaste und wählen im Kontextmenü den Menübefehl **Laden in CPU**

Die Konfiguration des gesamten Automatisierungssystems wird in die CPU geladen. CPU-Parameter werden sofort wirksam, die Parameter für die übrigen Baugruppen werden im Anlauf an die Baugruppen übertragen.

Achtung

Teilkonfigurationen, z. B. die Konfiguration einzelner Baugruppenträger, können nicht in eine Station geladen werden. Aus Konsistenzgründen lädt STEP 7 Lite immer die komplette Konfiguration in die CPU.

Betriebszustand der CPU beim Laden ändern

Wenn Sie die Funktion **Datei > Laden in CPU** anstoßen, können Sie dialoggeführt folgende Aktionen per PG ausführen:

- CPU in STOP versetzen
(falls Betriebsartenschalter auf RUN-P steht oder die Verbindung zur CPU durch Passwort legitimiert wurde)
- Speicher komprimieren
(falls nicht genügend zusammenhängender freier Speicher zur Verfügung steht)
- CPU wieder in RUN versetzen

9.2 Laden aus der CPU in das PG

Diese Funktion unterstützt Sie bei folgenden Tätigkeiten:

- Sichern von Informationen aus der CPU (z. B. für Service-Zwecke).
- Schnelles Konfigurieren und Bearbeiten einer Station, wenn die Hardware-Komponenten zum Beginn der Konfigurierungstätigkeit vorhanden sind.

Sichern von Informationen aus dem Zielsystem

Diese Maßnahme kann erforderlich sein, wenn z. B. die Offline-Projektdateien des in der CPU laufenden Standes nicht oder nur teilweise vorhanden sind. Sie können sich dann zumindest den online verfügbaren Teil der Projektdateien auf Ihr PG holen.

Schnelles Konfigurieren

Sie können sich die Eingabe des Stationsaufbaus erleichtern, indem Sie nach Aufbau der Hardware und Neustart (Warmstart) der Station die Konfigurationsdaten aus dem Zielsystem auf Ihr PG laden. Sie erhalten damit den Stationsaufbau mit den Typangaben der einzelnen Baugruppen. Danach müssen Sie nur noch die einzelnen Baugruppen durch Baugruppen aus dem Katalog ersetzen und parametrieren.

In das PG wird die Konfiguration für den zentralen Baugruppenträger ("Rack 0") und eventuell vorhandene Erweiterungsbaugruppenträger geladen.

Achtung

Beim Laden in das PG (ohne dass eine Offline-Konfiguration vorhanden ist) kann STEP 7 Lite nicht alle Bestellnummern der Komponenten vollständig ermitteln.

Die Baugruppen mit "unvollständig" erscheinenden Bestellnummern können Sie beim Konfigurieren der Hardware durch die entsprechende Baugruppen aus dem Katalog ersetzen.

9.2.1 Was wird wann zurückgeladen?

Im Menü "Datei" finden Sie den Menübefehle "Laden in PG" zum Zurückladen von Daten aus der CPU in das Offline-Projekt auf dem PG/PC.

Wenn Sie den Menübefehl ausführen, werden die Objekte in das PG/PC zurückgeladen, die Sie im Projektfenster bzw. in der Sicht "CPU online" markiert haben. Wenn sich bereits Bausteine mit identischen Bausteinnummern im Offline-Projekt befinden, können Sie im Folgedialog entscheiden, OB diese Bausteine überschrieben werden sollen oder nicht.

Was grundsätzlich nicht zurückgeladen werden kann

Beim Laden eines Programms vom PG in die CPU wird nicht alles, was in der Projektdatenhaltung gespeichert ist, in die CPU geladen. Diese Daten fehlen folglich, wenn Sie z. B. Bausteine zurückladen.

Für Daten, die von der CPU in das PG geladen werden, gilt:

- Bausteine enthalten keine symbolischen Namen für Formalparameter, temporäre Variablen und Marken. Als Ersatzsymbole erzeugt STEP 7 Lite Namen wie z. B. IN0, STAT1, M001. Die Kommentare aus der Variablendeklarationstabelle fehlen ebenfalls.
- Statt anwenderdefinierter Datentypen (UDT) werden Strukturen angezeigt und auch weiterhin verwendet. Die Änderung eines bereits verwendeten UDT wird nicht berücksichtigt.
- Bausteine enthalten keine Kommentare. Sämtliche Übersetzungen im Projekt fehlen.
- FUP- oder KOP-Bausteine enthalten keine Netzwerk-Kommentare und Titel und auch keine Bausteinkommentare und -titel.
- AWL-Bausteine enthalten keine Zeilenkommentare.
- Die Informationen, die zum "Schnittstelle aktualisieren bei Bausteinaufrufen" notwendig sind, fehlen. Sollte also der aufgerufene Baustein offline fehlen oder mit geänderter Schnittstelle vorliegen, wird der Programmcode des Netzwerks mit dem Konflikt zum aufgerufenen Baustein in expandierter AWL - Darstellung (Disassembly=MC7) angezeigt. Das ist unabhängig von der Erstsprache des Bausteins.
- Die Informationen, die für den "Operandenvorrang symbolisch" notwendig sind, fehlen. Zurückgeladenen Bausteine verhalten sich stets "absolut führend", unabhängig von den Einstellungen am Projekt.
- Querverweis-Daten fehlen. Abhilfe: Baustein im Programmeditor neu speichern
- Falls die CPU mit dem STEP 7 -Basispaket geladen wurde, können Daten zu den Themen "Globaldaten-Kommunikation (GD)", "Projektierung von symbolbezogenen Meldungen" und "Netzprojektierung" sowie "Dezentrale Peripherie" nicht weiterbearbeitet werden.
- Bei zurückgeladenen Forceaufträgen fehlen alle Kommentare und Formate.
- Kommentare in den Dialogen der Baugruppen werden nicht zurückgeladen

Falls Sie in ein "leeres" Projekt zurückladen, fehlen zusätzlich:

- die Symboltabelle mit den symbolischen Namen für die Operanden sowie Kommentare
- anwenderdefinierte Datentypen

Selektieren der zurückzuladenden Objekte

Es sind nur Elemente in das PG ladbar, die sich tatsächlich in der CPU befinden; d. h. wenn Elemente markiert sind, die als "nur offline" gekennzeichnet sind, ist der Menübefehl "Laden in PG" nicht aktivierbar.

Was ist markiert in der Projektübersicht (Online-CPU)?	Was wird in das PG geladen?	Bemerkungen
Das Projekt	Alles, was ladbar ist	Hardware-Konfiguration, alle Bausteine
Die Hardware-Konfiguration	Die Hardware-Konfiguration	Funktioniert in allen Sichten der Hardware-Konfiguration (HW-Vergleich, HW-Diagnose)
Das Programm	Alle Bausteine des Anwenderprogramms	-
Ein oder mehrere Bausteine	Die markierten Bausteine	-

9.2.2 Schritte zum Laden von Objekten aus der CPU in das PG/PC

1. Öffnen Sie das Projekt, in das die gewünschten Objekte (z. B. Bausteine) geladen werden sollen.
2. Markieren Sie das Objekt oder eine Auswahl der Objekte, die Sie in das PG/PC zurückladen wollen im Projektfenster, Register "Online-CPU".
Mögliche Objekte sind die Hardware, das Programm (mit allen Bausteinen) oder einzelne Bausteine.
3. Wählen Sie den Menübefehl **Datei > Laden in PG**.

Die markierten Objekte werden in die Datenhaltung des Projekts auf dem PG/PC übertragen.

9.2.3 Bearbeiten von geladenen Bausteinen im PG/PC

Sie können sich über die Ladefunktion (Menübefehl **Datei > Laden in PG**) aktuelle Inhalte von Bausteinen aus dem RAM-Ladespeicher der CPU in Ihr PG laden.

Achtung

Zeitstempelkonflikt bei Arbeiten online und offline

Die anschließend beschriebenen Vorgehensweisen führen zu Zeitstempelkonflikten und sollten deshalb vermieden werden.

Zeitstempelkonflikte entstehen beim Öffnen eines Bausteins aus der "Online-CPU-Sicht", wenn offline durchgeführte Änderungen nicht in die CPU geladen wurden.

Zeitstempelkonflikte entstehen außerdem beim Öffnen eines Bausteins aus der Projektsicht (d. h. offline), wenn ein Online-Baustein mit Zeitstempelkonflikt in das Anwenderprogramm offline kopiert und anschließend der Baustein offline geöffnet wird.

9.2.4 Bearbeiten einer geladenen Hardware-Konfiguration im PG/PC

Laden einer Hardware-Konfiguration aus der CPU in ein neues Projekt

Wenn Sie ohne Projektdaten im PG/PC eine Hardware-Konfiguration aus der CPU in das PG bzw. in den PC laden, dann kann STEP 7 Lite die genaue Bestellnummer und die damit verbundenen Eigenschaften der Baugruppe nicht vollständig ermitteln.

Diese Baugruppen werden in der Konfigurationstabelle mit Fragezeichen dargestellt.

Sie können die Baugruppen folgendermaßen spezifizieren:

1. Markieren Sie die Baugruppe in der Konfigurationstabelle
2. Wählen Sie das Register "Kompatibel" im Hardware-Katalog.
Dort werden alle Baugruppen aufgelistet, die mit der markierten Baugruppe austauschbar (kompatibel) sind.
3. Ersetzen Sie die Baugruppe in der Konfigurationstabelle durch die tatsächlich verwendete, kompatible Baugruppe per Drag&Drop.

9.3 Löschen in der CPU

9.3.1 Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU

Vor dem Laden Ihres Anwenderprogramms in die CPU sollten Sie die CPU urlöschen, um sicherzustellen, dass sich keine "alten" Bausteine mehr auf der CPU befinden.

Voraussetzung für das Urlöschen

Zum Urlöschen muss sich die CPU im Betriebszustand STOP befinden (Betriebsartenschalter auf STOP oder Betriebsartenschalter auf RUN-P und Ändern des Betriebszustands nach STOP über das CPU-Bedienpanel).

Urlöschen von CPUs

Beim Urlöschen von CPUs wird folgendes durchgeführt:

- Die CPU wird zurückgesetzt.
- Alle Anwenderdaten werden gelöscht (Bausteine und SDBs mit Ausnahme der MPI-Parametrierung).
- Die CPU bricht alle bestehenden Verbindungen ab.
- Wenn Daten auf einem EPROM vorhanden sind (Memory Card oder integriertes EPROM), kopiert die CPU nach dem Urlöschen den EPROM-Inhalt wieder in den RAM-Bereich des Speichers.

Der Inhalt des Diagnosepuffers, des Betriebsstundenzählers und die Parameter der MPI-Schnittstelle bleiben erhalten.

Urlöschen mit STEP 7 Lite

1. Schalten Sie die CPU folgendermaßen in den Betriebszustand STOP:
 - Stellen Sie den Betriebsartenschalter auf STOP.
 - Steht der Betriebsartenschalter auf RUN-P (RUN bei CPU 31xC), können Sie alternativ über das CPU-Bedienpanel den Betriebszustand auf STOP stellen.
 - Klicken Sie im CPU-Bedienpanel auf die Schaltfläche MRES.
Alternativ können Sie den Menübefehl **Extras > Urlöschen** verwenden.
2. Bestätigen Sie das Urlöschen im daraufhin erscheinenden Dialogfeld.

9.3.2 Löschen einzelner Bausteine in der CPU

Das Löschen einzelner Bausteinen in der CPU kann während der Testphase des CPU-Programms notwendig werden. Bausteine sind im Anwenderspeicher der CPU entweder im EPROM oder RAM gespeichert (abhängig von CPU und Ladevorgang).

- Bausteine, die im RAM geladen sind, können Sie direkt löschen. Der belegte Speicherplatz wird im Ladespeicher und im Arbeitsspeicher freigegeben.
- Bausteine, die im integrierten EPROM gespeichert sind, werden immer nach dem Urlöschen der CPU in den RAM-Bereich kopiert. Die Kopien im RAM-Bereich können direkt gelöscht werden. Die gelöschten Bausteine werden dabei im EPROM bis zum nächsten Urlöschen oder Netzausfall mit ungepuffertem RAM-Speicher für ungültig markiert. Nach Urlöschen oder Netzausfall mit ungepuffertem RAM-Speicher werden die "gelöschten" Bausteine wieder vom EPROM in das RAM kopiert und aktiv. Bausteine im integrierten EPROM (z. B. bei CPU 312) werden durch Überschreiben mit dem neuen RAM-Inhalt gelöscht.

Löschen im RAM der CPU

Sie können einzelne oder mehrere Bausteine in den Betriebszuständen STOP und RUN-P löschen. Wenn Sie in RUN-P einen Baustein löschen, der noch aufgerufen wird, geht entweder die CPU in STOP oder es wird ein Fehler-OB aufgerufen.

Gehen Sie folgendermaßen vor:

1. Markieren Sie die zu löschenden Bausteine im Fenster "Online-CPU".
2. Wählen Sie den Menübefehl **Bearbeiten > Löschen** oder drücken Sie die DEL-Taste.

Um das gesamte Anwenderprogramm der CPU zu löschen, können Sie die CPU auch urlöschen.

Löschen im integrierten EPROM

Das integrierte EPROM der CPU 312 wird gelöscht, indem Sie das EPROM wieder mit dem aktuellen RAM-Inhalt, in dem alle Anwenderbausteine gelöscht wurden, überschreiben.

Gehen Sie folgendermaßen vor:

1. Löschen Sie das Anwenderprogramm im RAM der CPU wie oben beschrieben.
2. Öffnen Sie das erweiterte CPU-Panel.
3. Wählen Sie im CPU-Panel die Schaltfläche "RAM > ROM".

9.3.3 Löschen der Memory Card in der CPU

Der Funktionsumfang - OB das gesamte Anwenderprogramm oder auch einzelne Bausteine in der Memory Card gelöscht werden können - hängt von der verwendeten CPU ab.

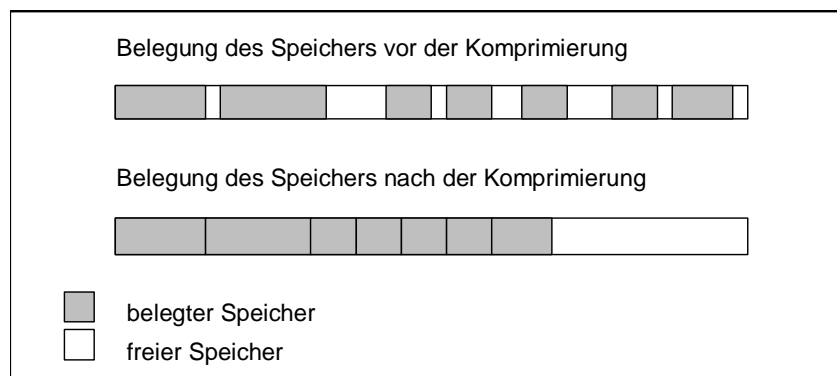
1. Öffnen Sie die Sicht "Online-CPU" des Projektfensters.
2. Doppelklicken Sie auf das Symbol S7-Memory Card.
3. Markieren Sie das Anwenderprogramm bzw. Teile davon und wählen Sie den Menübefehl **Bearbeiten > Löschen** oder drücken Sie die DEL-Taste.

9.4 Komprimieren des Anwenderspeichers (RAM)

9.4.1 Entstehung von Lücken im Anwenderspeicher (RAM)

Durch mehrfaches Löschen und Nachladen von Bausteinen entstehen im Anwenderspeicher (Lade- und Arbeitsspeicher) Lücken, die den nutzbaren Speicherbereich verringern. Durch Komprimieren werden die vorhandenen Bausteine lückenlos im Anwenderspeicher angeordnet und es entsteht ein zusammenhängender freier Speicherbereich.

Nachfolgendes Bild zeigt schematisch, wie belegte Speicherblöcke durch die Funktion "Speicher komprimieren" verschoben werden.



Möglichst im Betriebszustand STOP komprimieren

Nur das Komprimieren im Betriebszustand STOP schließt alle Speicherlücken. Wenn Sie im Betriebszustand RUN-P (Stellung des Betriebsartenschalters) komprimieren, so werden die gerade bearbeiteten Bausteine nicht verschoben, da sie geöffnet sind. Im Betriebszustand RUN (Stellung des Betriebsartenschalters) kann die Funktion "Komprimieren" nicht durchgeführt werden (Schreibschutz!).

9.4.2 Komprimieren des Speicherinhalts einer CPU

Möglichkeiten zum Komprimieren

Sie haben zwei Möglichkeiten zum Komprimieren des Anwenderspeichers:

- Tritt beim Laden ein Speicherplatzmangel auf dem Zielsystem auf, so wird ein Dialogfeld aufgeblendet, das Sie auf den Fehler hinweist. Durch Betätigen der entsprechenden Schaltfläche im Dialogfeld können Sie den Speicher komprimieren.
- Als vorbeugende Maßnahme können Sie sich die Speicherauslastung anzeigen lassen:
 - Öffnen Sie die Hardware in der Sicht "Online-CPU".
 - Wählen Sie in der Hardwaresicht das Register "HW-Diagnose" und selektieren Sie dort die CPU.
 - Wählen Sie den Menübefehl **Extras > Baugruppenzustand**.
 - Wählen Sie im folgenden Dialogfeld das Register "Speicher". In diesem Register gibt es eine Schaltfläche für das Komprimieren des Speichers, falls die CPU diese Funktion unterstützt.

10 Testen

10.1 Übersicht der verschiedenen Testarten

STEP 7 Lite bietet die Möglichkeit, den Ablauf Ihres Programms auf dem Zielsystem zu testen. Das ablauffähige Programm muss dazu auf dem Zielsystem geladen sein. Sie können dann Signalzustände und Werte von Variablen beobachten und Variablen mit Werten vorbelegen, um damit bestimmte Situationen für den Programmablauf zu simulieren.

Folgende Testmöglichkeiten stehen Ihnen dafür zur Verfügung:

- Testen mit der Variablen-tabelle
Anwendung: zum Durchspielen verschiedener Ablaufsituationen
- Testen mit dem Programmstatus
Anwendung: zur schrittweisen Verfolgung des Programmablaufs
- Testen mit Simulationsprogramm
Anwendung: zum Testen, wenn kein Zielsystem vorhanden ist

Zum Testen mit dem Simulationsprogramm benötigen Sie das Optionspaket S7-PLCSIM.

10.2 Testen mit Variablen-tabellen und Forceta-bellen

10.2.1 Einführung zum Testen mit Variablen-tabellen und Forceta-bellen

Variablen-tabellen und Forceta-bellen bieten den Vorteil, unterschiedliche Testumgebungen abspeichern zu können. So können Tests während einer Inbetriebsetzung oder zu Service- und Wartungszwecken reproduziert werden.

Beim Testen mit Variablen-tabellen stehen Ihnen folgende Funktionen zur Verfügung:

- **Beobachten von Variablen**
Damit lassen sich die aktuellen Werte einzelner Variablen eines Anwenderprogramms bzw. einer CPU am PG/PC anzeigen.
- **Steuern von Variablen**
Mit dieser Funktion können Sie einzelnen Variablen eines Anwenderprogramms bzw. einer CPU feste Werte zuweisen. Das Steuern ist auch beim Testen mit Programmstatus möglich.
- **PA freischalten und Steuern Sofort**
Diese beiden Funktionen geben Ihnen die Möglichkeit, einzelnen Peripherieausgängen einer CPU im Betriebszustand STOP feste Werte zuzuweisen.

- **Forcen von Variablen**

Verwenden Sie diese Funktion, um einzelnen Variablen eines Anwenderprogramms bzw. einer CPU mit einem Wert fest zu belegen, der vom Anwenderprogramm nicht überschrieben werden kann.

Sie können folgende Variablen beobachten bzw. steuern:

- Eingänge, Ausgänge, Merker, Zeiten und Zähler
- Inhalte von Datenbausteinen
- Peripherie

10.2.2 **Prinzipielle Vorgehensweise beim Beobachten und Steuern mit Variablen tabellen**

Um die Funktionen **Beobachten/Steuern** zu benutzen, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie das Symbol "Beobachten/Steuern" im Projektfenster. Im Arbeitsbereich wird die Sicht einer Variablen tabelle eingeblendet, in der Operanden beobachtet und gesteuert werden können.
2. Tragen Sie die zu beobachtenden bzw. zu steuernden Operanden in die Variablen tabelle ein oder wählen Sie unter "Variablen tabelle" eine zuvor erstellte Variablen tabelle aus.
3. Starten Sie Ihre Tests mit "Beobachten" oder "Steuern".
4. Speichern Sie die Änderungen in der Variablen tabelle mit dem Menübefehl **Datei > Speichern**.

10.2.3 **Prinzipielle Vorgehensweise beim Beobachten und Forcen mit Forceta bellen**

Um die Funktionen Beobachten/Forcen zu benutzen, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie das Symbol "Beobachten/Steuern" in der Projektübersicht. Im Arbeitsbereich wird die Sicht einer Forceta belle eingeblendet, in der Operanden beobachtet und geforct werden können.
2. Tragen Sie die zu beobachtenden bzw. zu forcenden Operanden in die Forceta belle ein oder wählen Sie unter "Forceta belle" eine zuvor erstellte Forceta belle aus.
3. Starten Sie Ihre Tests mit "Beobachten" oder "Werte Forcen".
4. Speichern Sie die Änderungen in der Forceta belle mit dem Menübefehl **Datei > Speichern**.

10.2.4 Bearbeiten und Speichern von Variablen Tabellen und Forcetabellen

10.2.4.1 Erstellen und Öffnen einer Variablen Tabelle

Variablen zum Beobachten oder Steuern werden in Variablen Tabellen abgespeichert. Eine einmal erstellte Variablen Tabelle können Sie speichern, duplizieren, ausdrucken und immer wieder zum Beobachten und Steuern verwenden.

Vorgehensweise zum Erstellen und Öffnen einer neuen Variablen Tabelle

1. In der Sicht "Beobachten/Steuern" klicken Sie auf die Schaltfläche "Tabellen verwalten".
2. Klicken Sie in dem Dialogfeld zur Verwaltung von Variablen Tabellen auf die Schaltfläche "Neu" und geben einen Name für die neue Variablen Tabelle ein.
3. Klicken Sie in dem Dialogfeld zur Verwaltung von Variablen Tabellen auf die Schaltfläche "Anzeigen" damit die neu erstellte Variablen Tabelle angezeigt wird.

Vorgehensweise zum Öffnen einer Variablen Tabelle

Schon vorhandene Variablen Tabelle öffnen Sie, indem Sie die Variablen Tabelle in der Sicht "Beobachten/Steuern" aus dem Feld "Variablen Tabelle" auswählen.

10.2.4.2 Erstellen und Öffnen einer Forcetabelle

Variablen zum Beobachten oder Forcen werden in Forcetabellen abgespeichert. Eine einmal erstellte Forcetabelle können Sie speichern, duplizieren, ausdrucken und immer wieder zum Beobachten und Forcen verwenden.

Vorgehensweise zum Erstellen und Öffnen einer neuen Forcetabelle

In der Sicht "Beobachten/Steuern" klicken Sie auf die Schaltfläche "Tabellen verwalten".

Klicken Sie in dem Dialogfeld zur Verwaltung von Forcetabellen auf die Schaltfläche "Neu" und geben einen Name für die neue Forcetabelle ein.

Klicken Sie in dem Dialogfeld zur Verwaltung von Forcetabellen auf die Schaltfläche "Anzeigen" damit die neu erstellte Forcetabelle angezeigt wird.

Vorgehensweise zum Öffnen einer Forcetabelle

Schon vorhandene Forcetabelle öffnen Sie, indem Sie die Forcetabelle in der Sicht "Beobachten/Steuern" aus dem Feld "Forcetabelle" auswählen.

10.2.4.3 Kopieren / Duplizieren von Variablentabellen

Wenn Sie eine einmal erstellte Variablentabelle als Muster für eine neu zu erstellende Variablentabelle verwenden wollen, dann duplizieren Sie das Muster folgendermaßen:

1. Klicken Sie in der Sicht "Beobachten/Steuern" auf die Schaltfläche "Tabellen verwalten".
2. Wählen Sie aus dem Feld der verfügbaren Variablentabellen die aus, die Sie als Muster duplizieren wollen.
3. Klicken Sie auf die Schaltfläche "Duplizieren".
4. Wählen Sie einen anderen Namen für die duplizierte Tabelle.
5. Klicken Sie auf "Übernehmen" und beenden Sie das Dialogfeld mit "Schließen". Die neue Variablentabelle hat denselben Aufbau wie die ursprüngliche und kann bearbeitet werden.

Variablentabellen aus einem bestehenden Projekt übernehmen

Wenn Sie für ein neues Projekt eine bereits bestehende Variablentabelle verwenden wollen, dann gehen Sie folgendermaßen vor:

1. Wählen Sie im Zielprojekt die Sicht "Beobachten/Steuern".
2. Wählen Sie im Zielprojekt die Variablentabelle, in die der Quellinhalt eingefügt werden soll.
3. Starten Sie STEP 7 Lite ein zweites mal und öffnen Sie das Quellprojekt.
4. Wählen Sie im Quellprojekt die Sicht "Beobachten/Steuern".
5. Wählen Sie im Quellprojekt die Variablentabelle, die als Quelle eingefügt werden soll.
6. Markieren Sie in dieser Variablentabelle den Bereich, den Sie übernehmen (kopieren) wollen.
7. Wählen Sie den Menübefehl **Bearbeiten > Kopieren**.
8. Wechseln Sie in das Zielprojekt und stellen Sie den Mauszeiger in die Variablentabelle des neuen Projekts.
9. Wählen Sie den Menübefehl **Bearbeiten > Einfügen**.

10.2.4.4 Kopieren / Duplizieren von Forcetabellen

Wenn Sie eine einmal erstellte Forcetabelle als Muster für eine neu zu erstellende Forcetabelle verwenden wollen, dann duplizieren Sie das Muster folgendermaßen:

1. Klicken Sie in der Sicht "Beobachten/Steuern" auf die Schaltfläche "Tabellen verwalten".
2. Wählen Sie aus dem Feld der verfügbaren Forcetabellen die aus, die Sie als Muster duplizieren wollen.
3. Klicken Sie auf die Schaltfläche "Duplizieren".
4. Wählen Sie einen anderen Namen für die duplizierte Tabelle.
5. Klicken Sie in dem Dialogfeld zur Verwaltung von Forcetabellen auf die Schaltfläche "Anzeigen" damit die duplizierte Forcetabelle angezeigt wird.

Forcetabellen aus einem bestehenden Projekt übernehmen

Wenn Sie für ein neues Projekt eine bereits bestehende **Forcetabelle** verwenden wollen, dann gehen Sie folgendermaßen vor:

1. Wählen Sie im Zielprojekt die Sicht "Beobachten/Steuern".
2. Wählen Sie im Zielprojekt die **Forcetabelle**, in die der Quellinhalt eingefügt werden soll.
3. Starten Sie STEP 7 Lite ein zweites mal und öffnen Sie das Quellprojekt.
4. Wählen Sie im Quellprojekt die Sicht "Beobachten/Steuern".
5. Wählen Sie im Quellprojekt die **Forcetabelle**, die als Quelle eingefügt werden soll.
6. Markieren Sie in dieser **Forcetabelle** den Bereich, den Sie übernehmen (kopieren) wollen.
7. Wählen Sie den Menübefehl **Bearbeiten > Kopieren**.
8. Wechseln Sie in das Zielprojekt und stellen Sie den Mauszeiger in die **Forcetabelle** des neuen Projekts.
9. Wählen Sie den Menübefehl **Bearbeiten > Einfügen**.

10.2.4.5 Speichern einer Variablentabelle

Gespeicherte Variablentabellen können Sie beim erneuten Test Ihres Programms wieder zum Beobachten und Steuern verwenden.

1. Speichern Sie die Variablentabelle mit dem Menübefehl **Datei > Speichern**.

Beim Speichern der Variablentabelle werden auch Spaltenbreiten, Beobachtmodus und Steuermodus abgespeichert.

10.2.4.6 Speichern einer Forcetabelle

Gespeicherte Forcetabellen können Sie beim erneuten Test Ihres Programms wieder zum Beobachten und Forcen verwenden.

1. Speichern Sie die Forcetabelle mit dem Menübefehl **Datei > Speichern**.

Beim Speichern der Forcetabelle werden auch Spaltenbreiten, und Beobachtmodus abgespeichert.

10.2.5 Eingeben von Variablen in Variablen Tabellen und Forcetabellen

10.2.5.1 Eingabe von Operanden oder Symbolen in eine Variablen Tabelle

Bestimmen Sie die Variablen, deren Werte Sie beobachten oder steuern wollen und tragen Sie diese in die Variablen Tabelle ein.

Wenn Sie z. B. das Eingangsbit 1.0, das Merkerwort 5 und das Ausgangsbyte 0 beobachten möchten, geben Sie in der Operandenspalte folgendes ein:

Beispiel:

E 1.0
MW 5
AB 0

Beispiel für eine ausgefüllte Variablen Tabelle

Nachfolgendes Bild zeigt eine Variablen Tabelle mit den eingeblendeten Spalten: Operand, Symbol, Anzeigeformat, Statuswert und Steuerwert.

Status	Operand	Symbol	Statuswert	Anzeigeformat	Steuerwert	Kommentar
						OB1 Netzwerk 1
	E0.1	"Taster 1"	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	
	E0.2	"Taster 2"	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	
	A4.0	"Lampe Grün"	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	
						OB1 Netzwerk 3
	E0.5	"Automatik Ein"	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	
	E0.6	"Hand Ein"	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	
	A4.2	"Automatikbetrieb"	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	
						Aufruf FB1 für Pos

Hinweise zur Variableneingabe über Symbole

- Die zu steuernde Variable können Sie als Operand oder als Symbol angeben. Operanden und Symbole können Sie sowohl in die Spalte "Operand" als auch in die Spalte "Symbol" eintragen. Der Eintrag wird automatisch in die passende Spalte geschrieben.
Falls in der Symboltabelle ein entsprechendes Symbol definiert ist, können Sie während der Eingabe den gewünschten Operanden bzw. das gewünschte Symbol aus einer Liste auswählen.
- Sie können nur solche Symbole eintragen, die bereits in der Symboltabelle definiert sind.
- Ein Symbol muss genau so eingegeben werden, wie es in der Symboltabelle definiert ist.
- Symbolnamen, die Sonderzeichen enthalten, müssen in Hochkommata eingeschlossen werden (z. B. "Motor.Aus", "Motor+Aus", "Motor-Aus").
- Um neue Symbole in der Symboltabelle zu definieren, wählen Sie den Menübefehl **Ansicht > Symbole**.

Syntaxprüfung

Wenn Sie Variablen in die Variablen-tabelle eintragen, so wird vor Verlassen der Zeile eine Syntaxprüfung durchgeführt. Fehlerhafte Einträge werden rot gekennzeichnet.

Maximale Größe

Eine Variablen-tabelle kann maximal 1024 Zeilen umfassen.

Einzelne Operanden vom Steuern und Beobachten ausschließen

Um einzelne Operanden vom Steuern und Beobachten auszuschließen, schalten Sie die entsprechende Zeile aus. Positionieren Sie die Schreibmarke in die gewünschte Zeile und wählen Sie im Kontextmenü (Klick mit der rechten Maustaste) den Kontextmenübefehl **Zeile ausschalten**.

10.2.5.2 Eingabe von Operanden oder Symbolen in eine Forcetablelle

Bestimmen Sie die Variablen, deren Werte Sie beobachten oder forcen wollen und tragen Sie diese in die Forcetablelle ein.

Wenn Sie z. B. das Eingangsbit 1.0 und das Ausgangsbyte 0 beobachten möchten, geben Sie in der Operandenspalte folgendes ein:

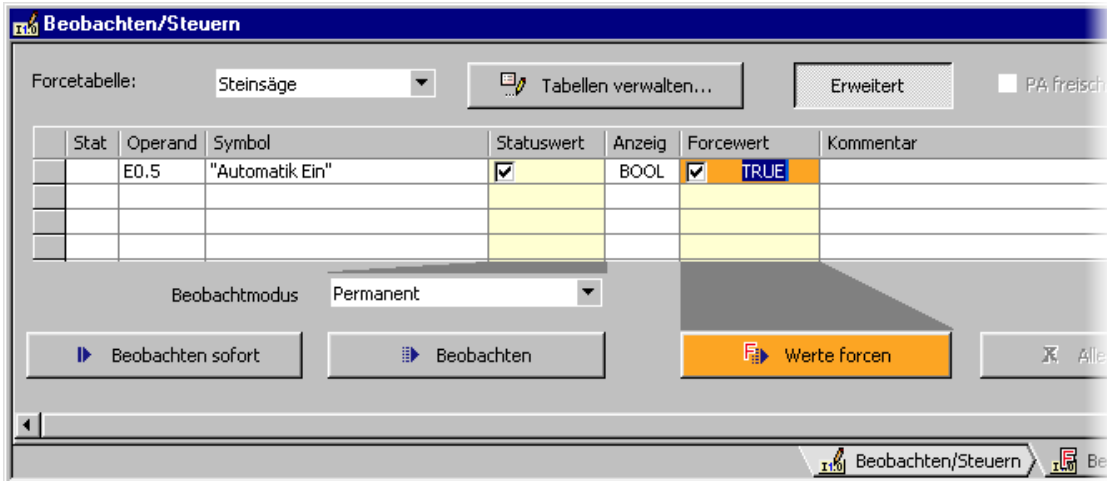
Beispiel:

E 1.0

AB 0

Beispiel für eine ausgefüllte Forcetabelle

Nachfolgendes Bild zeigt eine Forcetabelle mit den eingeblendeten Spalten: Operand, Symbol, Anzeigeformat, Statuswert und Forcewart.



Hinweise zur Variableneingabe über Symbole

- Die zu forcenden Variable können Sie als Operand oder als Symbol angeben. Operanden und Symbole können Sie sowohl in die Spalte "Operand" als auch in die Spalte "Symbol" eintragen. Der Eintrag wird automatisch in die passende Spalte geschrieben.
Falls in der Symboltabelle ein entsprechendes Symbol definiert ist, können Sie während der Eingabe den gewünschten Operanden bzw. das gewünschte Symbol aus einer Liste auswählen.
- Sie können nur solche Symbole eintragen, die bereits in der Symboltabelle definiert sind.
- Ein Symbol muss genau so eingegeben werden, wie es in der Symboltabelle definiert ist.
- Symbolnamen, die Sonderzeichen enthalten, müssen in Hochkommata eingeschlossen werden (z. B. "Motor.Aus", "Motor+Aus", "Motor-Aus").
- Um neue Symbole in der Symboltabelle zu definieren, wählen Sie den Menübefehl **Ansicht > Symbole**.

Syntaxprüfung

Wenn Sie Variablen in die Forcetabelle eintragen, so wird vor Verlassen der Zeile eine Syntaxprüfung durchgeführt. Fehlerhafte Einträge werden rot gekennzeichnet.

Maximale Größe

Eine Forcetabelle kann maximal 1024 Zeilen umfassen.

Einzelne Operanden vom Steuern und Beobachten ausschließen

Um einzelne Operanden vom Forcen und Beobachten auszuschließen, schalten Sie die entsprechende Zeile aus. Positionieren Sie die Schreibmarke in die gewünschte Zeile und wählen Sie im Kontextmenü (Klick mit der rechten Maustaste) den Kontextmenübefehl **Zeile ausschalten**.

10.2.5.3 Einfügen eines zusammenhängenden Operandenbereichs in eine Variablen-tabelle

1. Öffnen Sie eine Variablen-tabelle.
2. Positionieren Sie die Einfügemarke in die Zeile, nach der Sie den zusammenhängenden Operandenbereich einfügen möchten.
3. Wählen Sie den Menübefehl **Einfügen > Operandenbereich**. Das Dialogfeld "Operandenbereich einfügen" wird angezeigt.
4. Tragen Sie in das Feld "Ab Operand" z. B. EW1 ein.
5. Tragen Sie in das Feld "Anzahl" die Anzahl der einzufügenden Operanden ein.
6. Wählen Sie aus der angezeigten Liste das gewünschte Anzeigeformat aus.
7. Klicken Sie auf "OK".

Der Operandenbereich wird in die Variablen-tabelle eingefügt.

10.2.5.4 Einfügen eines zusammenhängenden Operandenbereichs in eine Forcetabelle

1. Öffnen Sie eine Forcetabelle.
2. Positionieren Sie die Einfügemarke in die Zeile, nach der Sie den zusammenhängenden Operandenbereich einfügen möchten.
3. Wählen Sie den Menübefehl **Einfügen > Operandenbereich**. Das Dialogfeld "Operandenbereich einfügen" wird angezeigt.
4. Tragen Sie in das Feld "Ab Operand" z.B. EW1 ein.
5. Tragen Sie in das Feld "Anzahl" die Anzahl der einzufügenden Operanden ein.
6. Wählen Sie aus der angezeigten Liste das gewünschte Anzeigeformat aus.
7. Klicken Sie auf "OK".

Der Operandenbereich wird in die Forcetabelle eingefügt.

10.2.5.5 Obergrenzen für die Eingabe von Zeiten

Beachten Sie folgende Obergrenzen für die Eingabe von Zeiten:

W#16#3999 (Maximalwert im BCD-Format)

Beispiele

Operand	Anzeigeformat	Eingabe	Anzeige Steuerwert	Erläuterung
T 1	SIMATIC_ZEIT	137	S5TIME#130MS	Konvertierung in Millisekunden
MW 4	SIMATIC_ZEIT	137	S5TIME#890MS	Darstellung im BCD-Format möglich
MW 4	HEX	137	W#16#0089	Darstellung im BCD-Format möglich
MW 6	HEX	157	W#16#009D	Darstellung im BCD-Format nicht möglich, deshalb ist das Anzeigeformat SIMATIC_ZEIT nicht wählbar

Hinweis

- Sie können Zeiten in Millisekunden-Schritten eingeben, aber der eingegebene Wert wird an ein Zeitraster angepasst. Die Größe des Zeitrasters ist abhängig von der Größe des eingegebenen Zeitwertes (137 ergibt 130 ms, die 7 ms wurden abgerundet).
- Die Steuerwerte von Operanden vom Datentyp WORD, z. B. EW 1, werden ins BCD-Format konvertiert. Nicht jedes Bitmuster ist jedoch eine gültige BCD-Zahl! Wenn bei einem Operanden vom Datentyp WORD die Eingabe nicht als SIMATIC_ZEIT dargestellt werden kann, wird automatisch auf das voreingestellte Format umgeschaltet, so dass der eingegebene Wert angezeigt werden kann.

BCD-Format für Variablen im Format SIMATIC_ZEIT

Werte von Variablen im Format SIMATIC_ZEIT werden im BCD-Format eingegeben.

Die 16 Bit haben folgende Bedeutung:

| 0 0 x x | h h h h | z z z z | e e e e |

Bit 15 und 14 sind immer Null.

Bit 13 und 12 (mit xx markiert) legen den Multiplikator für die Bits 0 bis 11 fest:

- 00 => Multiplikator 10 Millisekunden
- 01 => Multiplikator 100 Millisekunden
- 10 => Multiplikator 1 Sekunde
- 11 => Multiplikator 10 Sekunden

Bit 11 bis 8 Hunderter (hhhh)

Bit 7 bis 4 Zehner (zzzz)

Bit 3 bis 0 Einer (eeee)

10.2.5.6 Obergrenzen für die Eingabe von Zählern

Bitte beachten Sie folgende Obergrenzen für die Eingabe von Zählern:

Obergrenzen für Zähler: C#999
W#16#0999 (Maximalwert im BCD-Format)

Beispiele

Operand	Anzeigeformat	Eingabe	Anzeige Steuerwert	Erläuterung
Z 1	ZAEHLER	137	C#137	Konvertierung
MW 4	ZAEHLER	137	C#89	Darstellung im BCD-Format möglich
MW 4	HEX	137	W#16#0089	Darstellung im BCD-Format möglich
MW 6	HEX	157	W#16#009D	Darstellung im BCD-Format nicht möglich, deshalb ist das Anzeigeformat ZAEHLER nicht wählbar

Hinweis

- Wenn Sie bei einem Zähler eine Dezimalzahl eingeben und den Wert nicht durch C# kennzeichnen, so wird dieser Wert automatisch in das BCD-Format umgerechnet (137 ergibt C#137).
- Die Steuerwerte von Operanden vom Datentyp WORD, z. B. EW 1, werden ins BCD-Format konvertiert. Nicht jedes Bitmuster ist jedoch eine gültige BCD-Zahl! Wenn bei einem Operanden vom Datentyp WORD die Eingabe nicht als ZÄHLER dargestellt werden kann, wird automatisch auf das voreingestellte Format umgeschaltet, so dass der eingegebene Wert angezeigt werden kann.

10.2.5.7 Beispiele

Beispiel für die Eingabe von Operanden in Variablentabellen

Erlaubter Operand:	Datentyp:	Beispiel (Mnemonic deutsch):
Eingang Ausgang Merker	BOOL	E 1.0 A 1.7 M 10.1
Eingang Ausgang Merker	BYTE	EB 1 AB 10 MB 100
Eingang Ausgang Merker	WORD	EW 1 AW 10 MW 100
Eingang Ausgang Merker	DWORD	ED 1 AD 10 MD 100
Peripherie (Eingang Ausgang)	BYTE	PEB 0 PAB 1
Peripherie (Eingang Ausgang)	WORD	PEW 0 PAW 1
Peripherie (Eingang Ausgang)	DWORD	PED 0 PAD 1
Zeiten	TIMER	T 1
Zähler	COUNTER	Z 1
Datenbaustein	BOOL	DB1.DBX 1.0
Datenbaustein	BYTE	DB1.DBB 1
Datenbaustein	WORD	DB1.DBW 1
Datenbaustein	DWORD	DB1.DBD 1

Beispiel für die Eingabe von Operanden in Forcetabellen

Erlaubter Operand:	Datentyp:	Beispiel (Mnemonic deutsch):
Eingang Ausgang Merker	BOOL	E 1.0 A 1.7 M 10.1
Eingang Ausgang Merker	BYTE	EB 1 AB 10 MB 100
Eingang Ausgang Merker	WORD	EW 1 AW 10 MW 100
Eingang Ausgang Merker	DWORD	ED 1 AD 10 MD 100
Peripherie (Eingang Ausgang)	BYTE	PEB 0 PAB 1
Peripherie (Eingang Ausgang)	WORD	PEW 0 PAW 1
Peripherie (Eingang Ausgang)	DWORD	PED 0 PAD 1

Hinweis:

Beim Forcen von S7-300-Baugruppen sind nur Eingänge und Ausgänge erlaubt.

Beispiel für die Eingabe eines zusammenhängenden Operandenbereichs

Öffnen Sie eine Variablen- oder Forcetable und lassen Sie sich das Dialogfeld "Bereich einfügen" mit dem Menübefehl **Einfügen > Operandenbereich** anzeigen.

Bei den Angaben im Dialogfeld werden folgende Zeilen für Merker in die Variablen- oder Forcetable eingefügt:

Ab Operand: M 3.0

Anzahl: 10

Anzeigeformat: BIN

Operand	Anzeigeformat
M 3.0	BIN
M 3.1	BIN
M 3.2	BIN
M 3.3	BIN
M 3.4	BIN
M 3.5	BIN
M 3.6	BIN
M 3.7	BIN
M 4.0	BIN
M 4.1	BIN

Beachten Sie in diesem Beispiel, wie sich die Bezeichnung in der Spalte "Operand" nach dem achten Eintrag ändert.

Beispiele für die Eingabe von Steuer-/Forcewerten

Bitoperanden

Mögliche Bitoperanden	erlaubte Steuer-/Forcewerte
E1.0	true
M1.7	false
A10.7	0
DB1.DBX1.1	1
E1.1	2#0
M1.6	2#1

Byteoperanden

mögliche Byteoperanden	erlaubte Steuer-/Forcewerte
EB 1	2#00110011
MB 12	b#16#1F
MB 14	1F
AB 10	'a'
DB1.DBB 1	10
PAB 2	12 (kein Forcen möglich)

Wortoperanden

mögliche Wortoperanden	erlaubte Steuer-/Forcewerte
EW 1	2#0011001100110011
MW 12	w#16#ABCD
MW 14	ABCD
AW 10	b#(12,34)
DB1.DBW 1	'ab'
PAW 2	12345 (kein Forcen möglich)
MW 3	12345
MW 5	S5t#12s340ms
MW 7	0.3s oder 0,3s
MW 9	C#123
MW 11	d#1990-12-31

Doppelwortoperanden

mögliche Doppelwortoperanden	erlaubte Steuer-/Forcewerte
ED 1	2#00110011001100110011001100110011
MD 0	1.23e4
MD 4	1.2
AD 10	dw#16#abcdef10
AD 12	ABCDEF10
DB1.DBD 1	b#(12,34,56,78)
PAD 2	'abcd' (kein Forcen möglich)
MD 8	L# -12
MD 12	L#12
MD 16	123456789
MD 20	123456789
MD 24	T#12s345ms
MD 28	Tod#1:2:34.567
MD 32	p#e0.0

Zeit

mögliche Operanden vom Typ Zeit	erlaubte Steuer-/Forcewerte	Erläuterung
T 1	0ms	Zeitwert in Millisekunden (ms)
T 12	20ms	Zeitwert in Millisekunden (ms)
T 14	12345ms	Zeitwert in Millisekunden (ms)
T 16	s5t#12s340ms	Zeitwert 12s 340ms
T 18	10.3ms	Zeitwert 10 ms
T 20	10.3s	Zeitwert 10s 300 ms

Das Steuern einer Zeit beeinflusst nur den Wert, nicht den Zustand. Das heißt, die Zeit T1 kann auf den Wert 0 gesteuert werden, das Verknüpfungsergebnis bei U T1 wird nicht verändert.

Die Zeichenfolgen "s5t", "s5time" können sowohl klein als auch groß geschrieben werden.

Zähler

Mögliche Operanden vom Typ Zähler	erlaubte Steuer-/Forcewerte
Z 1	0
Z 14	20
Z 16	c#123

Das Steuern eines Zählers beeinflusst nur den Wert, nicht den Zustand. Das heißt, der Zähler Z1 kann auf den Wert 0 gesteuert werden, das Verknüpfungsergebnis bei U Z1 wird nicht verändert.

10.2.6 Bearbeiten von Variablen in Variablen Tabellen und Forcetabellen

10.2.6.1 Festlegen des Anzeigeformats

Mit Eingabe eines Operanden wird das Anzeigeformat für diesen Operanden automatisch vorbesetzt.

Sie können das Anzeigeformat für den Operanden ändern:

1. Klicken Sie in der Spalte "Anzeigeformat" auf die entsprechende Zelle der Zeile.
2. Wählen Sie in der nun erscheinenden Klappliste unter den zulässigen Anzeigeformaten aus.

10.2.6.2 Ausschneiden von markierten Bereichen in die Zwischenablage

1. Markieren Sie eine oder mehrere komplette Zeilen, indem Sie diese mit dem Mauszeiger bei gedrückter linker Maustaste von oben nach unten überstreichen;
einen Operanden, ein Symbol oder einen Steuerwert, indem Sie mit dem Mauszeiger von links nach rechts bei gedrückter linker Maustaste darüber streichen.
2. Schneiden Sie den Bereich mit dem Menübefehl **Bearbeiten > Ausschneiden** aus. Der Bereich ist noch solange in der Zwischenablage vorhanden, bis er überschrieben wird.

10.2.6.3 Einfügen von Bereichen aus der Zwischenablage in die Variablen Tabelle bzw. Forcetable

1. Platzieren Sie den Mauszeiger an der Stelle der Variablen Tabelle bzw. Forcetable, an der Sie den Bereich aus der Zwischenablage einfügen wollen.
2. Fügen Sie den Bereich aus der Zwischenablage mit dem Menübefehl **Bearbeiten > Einfügen** ein.

10.2.6.4 Kopieren von markierten Bereichen in die Zwischenablage

1. Markieren Sie
eine oder mehrere komplette Zeilen, indem Sie diese mit dem Mauszeiger bei gedrückter linker Maustaste von oben nach unten überstreichen;
einen Operanden, ein Symbol oder einen Steuerwert, indem Sie mit dem Mauszeiger von links nach rechts bei gedrückter linker Maustaste darüber streichen.
2. Kopieren Sie den Bereich mit dem Menübefehl **Bearbeiten > Kopieren** in die Zwischenablage.

10.2.7 Beobachten von Variablen

10.2.7.1 Einführung zum Beobachten von Variablen

Zum Beobachten von Variablen haben Sie folgende Möglichkeiten:

- Schalten Sie die Funktion "Beobachten" durch Klicken auf die Schaltfläche "Beobachten" ein. Die Werte der ausgewählten Variablen werden in der Variablentabelle bzw. Forcetable angezeigt.
- Ist die Schaltfläche „Erweitert“ gedrückt:
- Schalten Sie die Funktion "Beobachten" durch Klicken auf die Schaltfläche "Beobachten Start" ein. Die Werte der ausgewählten Variablen werden, abhängig vom eingestellten Beobachtmodus, in der Variablentabelle bzw. Forcetable angezeigt. Wenn Sie einen Beobachtmodus "permanent" eingestellt haben, schalten Sie die Funktion "Beobachten Start" durch erneutes Klicken auf die Schaltfläche "Beobachten Start" wieder aus.
- Aktualisieren Sie die Werte der ausgewählten Variablen einmalig und sofort durch Klicken auf die Schaltfläche "Beobachten sofort". Die aktuellen Werte der ausgewählten Variablen werden in der Variablentabelle bzw. Forcetable angezeigt.

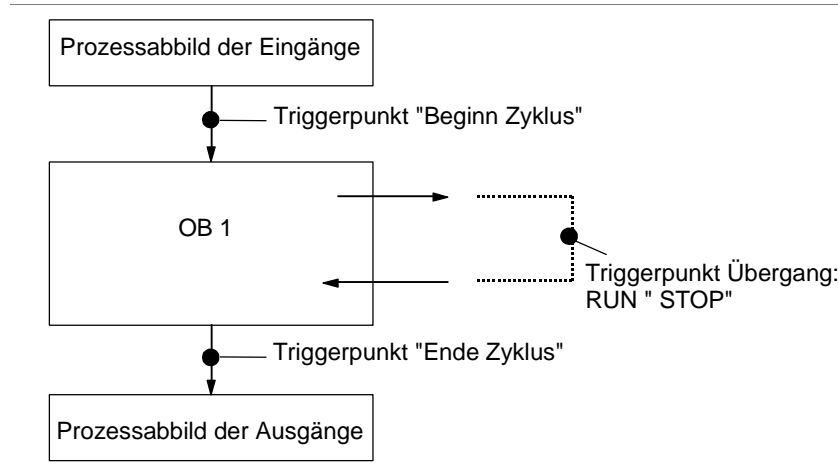
10.2.7.2 Festlegen des Beobachtmodus

Mit Festlegung des Beobachtmodus können Sie den Triggerpunkt und die Dauer des Beobachtens von Variablen festlegen. Folgende Beobachtmodi stehen zur Auswahl:

- Permanent
- Zyklusbeginn einmalig
- Zyklusende einmalig
- Zyklusbeginn permanent
- Zyklusende permanent
- Übergang von RUN nach STOP einmalig
- Übergang von RUN nach STOP permanent

Triggerpunkt

Der Triggerpunkt "Zyklusbeginn", "Zyklusende" und "Übergang von RUN nach STOP" legt fest, zu welchem Zeitpunkt die Variablen aus der CPU gelesen bzw. aktualisiert werden sollen. Die Lage der Triggerpunkte veranschaulicht das folgende Bild.



Damit Ihnen der gesteuerte Wert in der Spalte "Statuswert" angezeigt wird, sollten Sie den Triggerpunkt für Beobachten auf "Permanent" einstellen!

Beobachten sofort

Sie können die Werte ausgewählter Variablen über die Schaltfläche "Beobachten sofort" bzw. "Steuern sofort" aktualisieren. Dieser Auftrag wird einmalig und schnellstmöglich durchgeführt, ohne Bezug zu einer bestimmten Stelle im Anwenderprogramm. Diese Funktionen werden hauptsächlich im STOP-Zustand zum Steuern und Beobachten angewendet.

10.2.7.3 Beobachten von Variablen

1. Öffnen Sie die Sicht "Beobachten/Steuern" und wählen Sie in der Klappliste "Variablentabelle" bzw. "Forcetabelle" die Tabelle aus, die die zu beobachtenden Variablen enthält.
2. Stellen Sie sicher, dass eine Online-Verbindung zur CPU besteht.
3. Starten Sie das Beobachten mit Klick auf die Schaltfläche "Beobachten".
4. Sie können Sie das Beobachten durch erneutes Klicken auf die gedrückte Schaltfläche "Beobachten" wieder beenden.

10.2.7.4 Beobachten von Variablen einmalig und sofort

Gehen Sie folgendermaßen vor:

1. Öffnen Sie die Sicht "Beobachten/Steuern" und wählen Sie in der Klappliste "Variablen-tabelle" bzw. "Forcetable" die Tabelle aus, die die zu beobachtenden Variablen enthält.
2. Stellen Sie sicher, dass eine Online-Verbindung zur CPU besteht.
3. Durch Klicken auf die Schaltfläche „Erweitern“ erhalten Sie weitere Möglichkeiten zum Beobachten von Variablen.
4. Durch Klicken auf die Schaltfläche "Beobachten sofort" können Sie sich die Werte der Variablen einmalig und sofort anzeigen lassen.

10.2.8 Steuern von Variablen

10.2.8.1 Einführung zum Steuern von Variablen

Zum Steuern von Variablen haben Sie folgende Möglichkeiten:

- Schalten Sie die Funktion "Steuern" durch Klicken auf die Schaltfläche "Steuern" ein. Schalten Sie die Funktion "Steuern" durch erneutes Klicken auf die Schaltfläche "Steuern" wieder aus.
- Ist die Schaltfläche "Erweitert" gedrückt:
- Schalten Sie die Funktion "Steuern" durch Klicken auf die Schaltfläche "Steuern Start" ein. Das Anwenderprogramm übernimmt die Steuerwerte für die ausgewählten Variablen aus der Variablentabelle, abhängig vom eingestellten Steuermodus. Wenn Sie einen permanenten Steuermodus eingestellt haben, schalten Sie die Funktion "Steuern" durch erneutes Klicken auf die Schaltfläche "Steuern Start" wieder aus.
- Aktualisieren Sie die Werte der ausgewählten Variablen einmalig und sofort durch Klicken auf die Schaltfläche "Steuern sofort".
- Weitere Möglichkeiten bieten die Funktionen "Forcen" und "Peripherieausgänge (PA) freischalten".

Beachten Sie beim Steuern

- Steuern kann nicht rückgängig gemacht werden (z. B. mit dem Menübefehl **Bearbeiten > Rückgängig**).



Gefahr

Ein Verändern der Variablenwerte bei laufendem Anlagenbetrieb kann bei Funktionsstörungen oder Programmfehlern schwere Sach- und Personenschäden verursachen!

Vergewissern Sie sich, dass keine gefährlichen Zustände eintreten können, bevor Sie die Funktion "Steuern" ausführen!

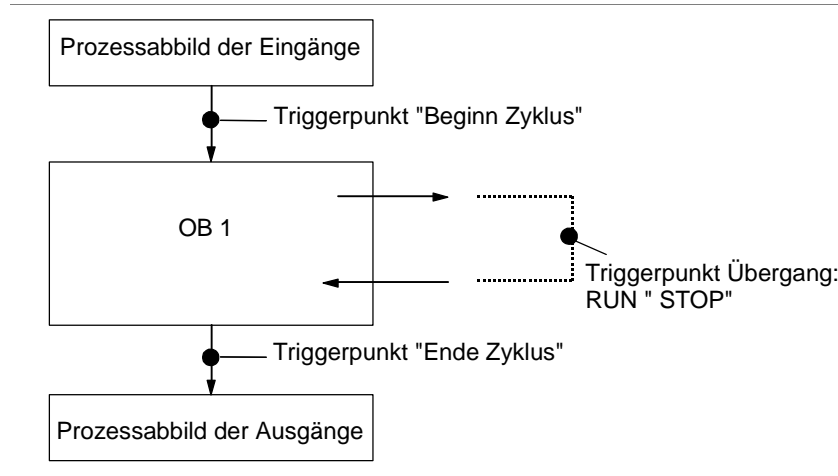
10.2.8.2 Festlegen des Steuermodus

Mit Festlegung des Steuermodus können Sie den Triggerpunkt und die Dauer des Steuern von Variablen festlegen. Folgende Steuermodi stehen zur Auswahl:

- Permanent
- Zyklusbeginn einmalig
- Zyklusende einmalig
- Zyklusbeginn permanent
- Zyklusende permanent
- Übergang von RUN nach STOP einmalig
- Übergang von RUN nach STOP permanent

Triggerpunkt

Der Triggerpunkt "Zyklusbeginn", "Zyklusende" und "Übergang von RUN nach STOP" legt fest, zu welchem Zeitpunkt die Variablen aus der CPU gelesen bzw. aktualisiert werden sollen. Die Lage der Triggerpunkte veranschaulicht das folgende Bild.



Aus der Lage der Triggerpunkte geht hervor:

- Das Steuern von Eingängen ist nur sinnvoll am Zyklusbeginn (entspricht Beginn des Anwenderprogramms OB 1), da sonst das Prozessabbild der Eingänge nach dem Steuern wieder aktualisiert und damit überschrieben wird.
- Das Steuern von Ausgängen ist nur sinnvoll am Zyklusende (entspricht Ende des Anwenderprogramms OB 1), da sonst das Prozessabbild der Ausgänge vom Anwenderprogramm überschrieben werden kann.

Damit Ihnen der gesteuerte Wert in der Spalte "Statuswert" angezeigt wird, sollten Sie den Triggerpunkt für Beobachten auf "Permanent" einstellen!

Beim Steuern von Variablen gilt bezüglich der Triggerpunkte:

- Wenn Sie als Steuermodus "einmalig" eingestellt haben, so erhalten Sie eine Meldung, falls die ausgewählten Variablen nicht zu steuern sind.
- Beim Steuermodus "permanent" erhalten Sie keine Rückmeldung.

Steuern sofort

Sie können die Werte ausgewählter Variablen durch Klicken auf die Schaltfläche "Steuern sofort" steuern. Dieser Auftrag wird einmalig und schnellstmöglich durchgeführt, ohne Bezug zu einer bestimmten Stelle im Anwenderprogramm. Diese Funktion wird hauptsächlich im STOP-Zustand zum Steuern angewendet.

10.2.8.3 Steuern von Variablen

1. Öffnen Sie die Sicht "Beobachten/Steuern" und wählen Sie in der Klappliste "Variablentabelle" die Tabelle, die die zu steuernden Variablen enthält, bzw. aktivieren Sie das Fenster, das die entsprechende Variablentabelle enthält.
2. Stellen Sie eine Verbindung zur gewünschten CPU her, damit Sie die Variablen der aktiven Variablentabelle steuern können.
3. Stellen Sie für das Steuern der Variablen den geeigneten Steuermodus ein.

Achtung

Das Einstellen des Steuermodus **geht nicht** bei laufendem Steuern! Stoppen Sie ggf. das Steuern. Steuern ist nicht aktiv, wenn die Schaltfläche "Steuern Start " nicht gedrückt ist.

4. Tragen Sie für die zu steuernden Variablen in der Spalte "Steuerwert" die vorzugebenden Werte ein und aktivieren Sie das Optionskästchen neben dem Steuerwert.
5. Starten Sie das Steuern durch Klicken auf die Schaltfläche "Steuern Start".
Wurde ein permanenter Steuermodus angewählt, bleibt die Schaltfläche gedrückt.
Wurde ein einmaliger Steuermodus angewählt, so wird die Steuerfunktion einmalig ausgeführt und die Schaltfläche springt anschließend wieder heraus.
6. Möchten Sie neue Werte vorgeben, einen anderen Trigger einstellen oder das Steuern beenden, dann klicken Sie bei gedrückter Schaltfläche "Steuern Start" nochmals auf diese Schaltfläche, so dass sie wieder herausspringt.
Für die Vereinbarung eines neuen Modus beginnen Sie wieder mit Schritt 3.
Für die Vorgabe von neuen Werten beginnen Sie wieder mit Schritt 4.

10.2.8.4 Sofortiges Steuern von Variablen

Gehen Sie folgendermaßen vor:

1. Öffnen Sie die Variablentabelle, die die zu steuernden Variablen enthält.
2. Tragen Sie für die zu steuernden Variablen in der Spalte "Steuerwert" die vorzugebenden Werte ein und aktivieren Sie die Optionskästchen neben den Werten.
3. Klicken Sie auf die Schaltfläche "Steuern sofort", um den Variablen Werte einmalig und sofort zuzuweisen.

10.2.8.5 Steuern: CPU im STOP mit eigenen Werten initialisieren

Gehen Sie folgendermaßen vor:

1. Öffnen Sie das CPU-Panel und schalten Sie die CPU in den Betriebszustand STOP.
2. Tragen Sie in die Variablen-tabelle bei den jeweiligen Variablen die gewünschten Steuerwerte ein und aktivieren Sie die Optionskästchen neben den Steuerwerten.
3. Aktivieren Sie die Steuerwerte durch Klicken auf die Schaltfläche "Steuern Start".
4. Schalten Sie die CPU über das CPU-Panel wieder in den Betriebszustand RUN.

10.2.8.6 Steuern der Peripherieausgänge im STOP-Zustand der CPU

Die Funktion "PA freischalten" schaltet die Ausgabesperre der Peripherieausgänge (PA) ab. Dies ermöglicht das Steuern der Peripherieausgänge im STOP-Zustand der CPU.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie die Sicht "Beobachten/Steuern" und wählen Sie in der Klappliste "Variablen-tabelle" die Tabelle aus, die die zu steuernden Peripherieausgänge enthält.
2. Öffnen Sie das CPU-Panel und schalten Sie die CPU in den Betriebszustand STOP.
3. Tragen Sie für die zu steuernden Peripherieausgänge in der Spalte "Steuerwert" die entsprechenden Werte ein und aktivieren Sie die Optionskästchen neben den Werten.
4. Schalten Sie den Modus "PA freischalten" ein, indem Sie das entsprechende Kontrollkästchen in der Sicht "Beobachten/Steuern" aktivieren.
5. Steuern Sie die Peripherieausgänge, indem Sie auf die Schaltfläche "Steuern sofort" klicken.
6. "PA freischalten" bleibt so lange aktiviert, bis Sie das Optionskästchen wieder deaktivieren.
7. Für die Vorgabe von neuen Werten beginnen Sie wieder mit Schritt 3.

Hinweis

- "PA freischalten" ist nur im Betriebszustand STOP möglich.
 - Der Modus "PA freischalten" wird durch folgende Ereignisse beendet:
 - durch Betriebszustandswechsel der CPU (Anzeige einer Meldung)
 - durch Deaktivieren des Optionskästchens "PA freischalten"
-

10.2.9 Forcen von Variablen

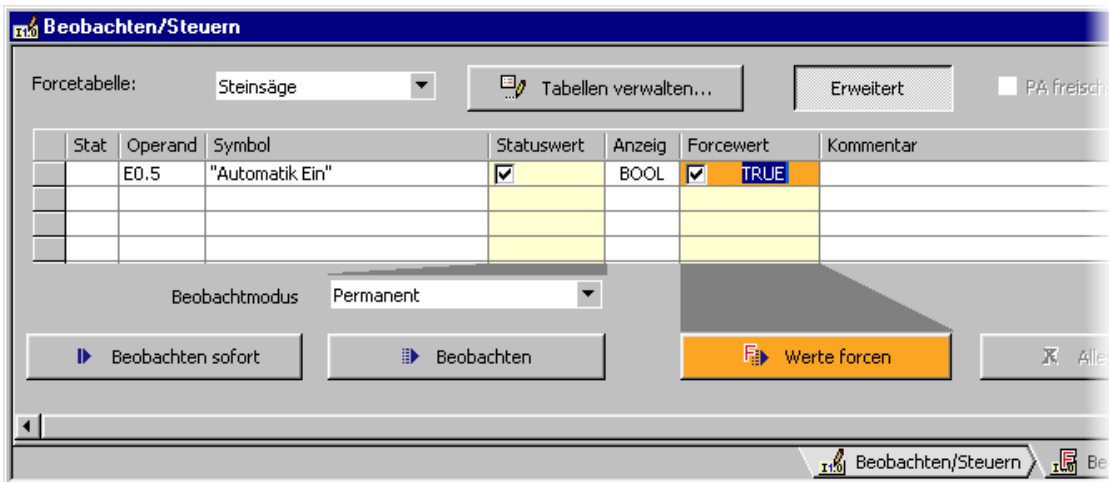
10.2.9.1 Einführung zum Forcen von Variablen

Die Funktionen zum Forcen sind nur in der Sicht "Beobachten/Forcen" anwählbar.

Zur Anzeige dieser Sicht doppelklicken Sie im Projektfenster auf "Beobachten/Steuern" und wählen anschließend das Register "Beobachten/Forcen".

Sie können einzelne Variablen eines Anwenderprogramms mit festen Werten vorbelegen, so dass sie auch vom Anwenderprogramm, das in der CPU abläuft, nicht verändert oder überschrieben werden können. Voraussetzung ist, dass die CPU diese Möglichkeit unterstützt.

Beispiel



10.2.9.2 Sicherheitsvorkehrungen zum Forcen von Variablen



Personen- und Sachschaden verhüten!

Beachten Sie, dass beim Ausführen der Funktion "Forcen" eine falsche Handlung

- das Leben oder die Gesundheit von Personen gefährden oder
- Schäden an der Maschine oder an der gesamten Anlage verursachen kann.



Vorsicht

- Bevor Sie die Funktion "Forcen" starten, sollten Sie sicherstellen, dass niemand gleichzeitig auf derselben CPU diese Funktion ausführt.
- Ein Forceauftrag kann nur durch Klicken der Schaltfläche "Alles entforcen" gelöscht bzw. beendet werden. Das Schließen der Sicht "Forcen" oder Beenden der Sicht "Beobachten/Steuern" löscht den Forceauftrag nicht.
- "Forcen" kann mit dem Menübefehl **Bearbeiten > Rückgängig** nicht rückgängig gemacht werden.
- Informieren Sie sich über die Unterschiede zwischen Forcen und Steuern von Variablen.
- Wenn eine CPU die Funktion Forcen nicht unterstützt, sind alle das Forcen betreffende Schaltflächen nicht anwählbar.

Wenn die Ausgabesperre über das Optionskästchen "PA freischalten" aufgehoben ist, geben alle geforcten Ausgabebaugruppen ihren Forcewert aus.

10.2.9.3 Anzeigen der Werte, die von der CPU geforct werden

1. Stellen Sie sicher, dass eine Online-Verbindung zur CPU besteht.
2. Öffnen Sie die Sicht "Beobachten/Steuern" und wählen Sie im Register "Beobachten / Forcen" in der Klappliste „Forcetabelle“ die Tabelle „Standard“ aus.

10.2.9.4 Werte forcen

1. Öffnen Sie die Sicht "Beobachten/Steuern" und wählen Sie das Register "Beobachten / Forcen" aus.
2. Tragen Sie in der Spalte "Operand" die Variablen ein, die Sie forcen wollen.

3. In der Spalte "Forcewert" geben Sie die Werte ein, mit denen die Variablen fest belegt werden sollen und aktivieren Sie die Optionskästchen neben den Werten.
4. Starten Sie das Forcen, indem Sie auf die Schaltfläche "Werte forcen" klicken.
Ergebnis:
 - Wenn kein Forceauftrag aktiv ist, werden die Variablen mit den Forcewerten fest belegt.
 - Wenn bereits ein Forceauftrag aktiv ist, müssen Sie sich entscheiden, OB Sie den bereits aktiven Forceauftrag ersetzen wollen. Stammt der bereits vorhandene Forceauftrag nicht von Ihnen, sprechen Sie sich bitte mit dem Erzeuger dieses Forceauftrags ab, bevor Sie den Forceauftrag ersetzen.

10.2.9.5 Werte entforcen

Den Forceauftrag beenden Sie, indem Sie auf die Schaltfläche "Alles entforcen" klicken. Stammt der Forceauftrag nicht von Ihnen, sprechen Sie sich bitte mit dem Erzeuger dieses Forceauftrags ab, bevor Sie den Forceauftrag löschen. Das Schließen des Fensters Forcewerte oder das Wechseln zu einer anderen Sicht löscht die Forcewerte in der CPU **nicht!**

10.2.9.6 Unterschiede zwischen Forcen und Steuern von Variablen

In der folgenden Übersicht sind die Unterschiede zwischen Forcen und Steuern zusammengefasst:

Merkmal / Funktion	Forcen mit CPU 318-2DP	Forcen mit S7-300 (ohne CPU 318-2DP)	Steuern
Merker (M)	ja	-	ja
Zeiten und Zähler (T, Z)	-	-	ja
Datenbausteine (DB)	-	-	ja
Peripherieeingänge (PEB, PEW, PED)	ja	-	-
Peripherieausgänge (PAB, PAW, PAD)	ja	-	ja
Eingänge und Ausgänge (E, A)	ja	ja	ja
Anwenderprogramm kann die Steuer-/ Forcewerte überschreiben	-	ja	ja
Ersetzen des Forcewertes unterbrechungsfrei wirksam	ja	ja	-
Nach Beenden der Applikation behalten die Variablen die Werte	ja	ja	-
Nach dem Auftrennen der Verbindung zur CPU behalten die Variablen die Werte	ja	ja	-
Trigger einstellen	immer Trigger sofort	immer Trigger sofort	einmalig oder permanent

Achtung

- Beim "PA freischalten" werden die Forcewerte für geforcete Peripherieausgänge an den entsprechenden Ausgabebaugruppen wirksam, die Steuerwerte für permanent gesteuerte Peripherieausgänge jedoch nicht.

10.3 Testen mit Programmstatus

10.3.1 Testen mit Programmstatus

Sie können Ihr Programm testen, indem Sie sich für jede Anweisung den Zustand des Programmstatus (VKE, Statusbit) oder den Inhalt der entsprechenden Register anzeigen lassen. Den Umfang der angezeigten Information legen Sie im Dialogfeld "Einstellungen" fest. Das Dialogfeld öffnen Sie mit dem Menübefehl

Extras > Einstellungen... in der Bausteineditor-Sicht.



Warnung

Ein Test bei laufendem Anlagenbetrieb kann bei Funktionsstörungen oder Programmfehlern schwere Sach- und Personenschäden verursachen!

Vergewissern Sie sich, dass keine gefährlichen Zustände eintreten können, bevor Sie diese Funktion ausführen!

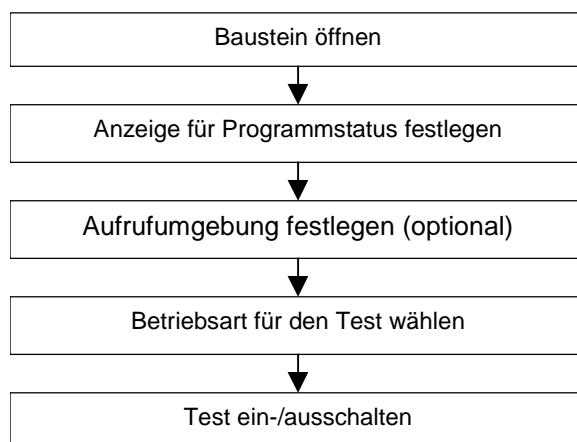
Voraussetzungen

Um den Programmstatus anzeigen lassen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Sie müssen den Baustein fehlerfrei gespeichert und danach auf die CPU geladen haben.
- Die CPU ist in Betrieb, das Anwenderprogramm läuft.
- Sie haben den Baustein geöffnet.

Prinzipielle Vorgehensweise für die Beobachtung des Programmstatus

Es ist dringend zu empfehlen, nicht sofort das gesamte Programm aufzurufen und zu testen, sondern die Bausteine nach und nach aufzurufen und zu testen. Dabei beginnen Sie mit den unterlagerten Codebausteinen, d. h. Sie testen die Bausteine in der letzten Schachtelungstiefe der Aufrufhierarchie, z. B. indem Sie diese im OB 1 aufrufen und über den Menübefehl **Beobachten und Steuern von Variablen** dem Baustein die zu testende Umgebung schaffen.



10.3.2 Anzeigen im Programmstatus

Die Anzeige des **Programmstatus** wird zyklisch aktualisiert. Sie beginnt mit dem markierten Netzwerk.

Voreingestellte Kennzeichnungen in KOP und FUP

- Status erfüllt: grüne durchgezogene Linien
- Status nicht erfüllt: blaue gestrichelte Linien
- Status unbekannt: schwarze durchgezogene Linien

Diese Voreinstellung von Linienart und Farbe ist änderbar über den Menübefehl **Extras > Einstellungen** / Register KOP/FUP.

Status von Elementen

- Der Status eines Kontakts ist
 - erfüllt, wenn der Operand den Wert "1" hat,
 - nicht erfüllt, wenn der Operand den Wert "0" hat,
 - unbekannt, wenn der Wert des Operanden unbekannt ist.
- Der Status von Elementen mit Freigabeausgang (ENO) entspricht dem Status eines Kontakts mit dem Wert des ENO-Ausgangs als Operanden.
- Der Status von Elementen mit Q-Ausgang entspricht dem Status eines Kontakts mit dem Wert des Operanden.
- Der Status bei CALLs ist erfüllt, wenn nach dem Aufruf das BIE-Bit gesetzt ist.
- Der Status einer Sprungoperation ist erfüllt, wenn der Sprung ausgeführt wird, d. h. wenn die Sprungbedingung erfüllt ist.
- Elemente mit Freigabeausgang (ENO) werden schwarz dargestellt, wenn der Freigabeausgang nicht beschaltet ist.

Status von Linien

- Linien sind schwarz, wenn sie nicht durchlaufen wurden oder ihr Status unbekannt ist.
- Der Status von Linien, die an der Stromschiene beginnen, ist immer erfüllt ("1").
- Der Status von Linien am Beginn von Parallelzweigen ist immer erfüllt ("1").
- Der Status der Linie nach einem Element ist erfüllt, wenn sowohl der Status der Linie vor dem Element als auch der Status des Elements erfüllt ist.
- Der Status der Linie nach NOT ist erfüllt, wenn der Status der Linie vor NOT nicht erfüllt ist (und umgekehrt).
- Der Status der Linie **nach** der Zusammenführung mehrerer Linien ist erfüllt, wenn
 - sowohl der Status von mindestens einer Linie **vor** der Zusammenführung erfüllt ist,
 - als auch der Status der Linie vor der Verzweigung erfüllt ist.

Status von Parametern

- Die Werte von Parametern **in fetter Schrift** sind aktuell.
- Die Werte von Parametern in magerer Schrift stammen aus einem vorhergehenden Zyklus; die Programmstelle wurde im aktuellen Zyklus nicht durchlaufen.

10.3.3 Programmstatus von Datenbausteinen

Einen Datenbaustein in der Datensicht können Sie online beobachten. Der Datenbaustein darf vor dem Start des Programmstatus nicht geändert werden. Besteht ein struktureller Unterschied (Deklaration) zwischen dem Online- und dem Offline-Datenbaustein, so kann direkt per Rückfrage der Offline-Datenbaustein in die CPU geladen werden.

Der Datenbaustein muss sich in der Ansicht "Datensicht" befinden, damit die Onlinewerte in der Spalte "Aktualwert" dargestellt werden können. Aktualisiert wird dabei nur der am Bildschirm sichtbare Teil des Datenbausteins. Während der Status aktiv ist, kann nicht in die Deklarationssicht umgeschaltet werden.

Während der Aktualisierung ist der grüne Laufbalken in der Statuszeile des Datenbausteins sichtbar und der Betriebszustand wird eingeblendet.

Die Werte werden im Format des jeweiligen Datentyps ausgegeben, eine Änderung des Formats ist nicht möglich.

Nach der Beendigung des Programmstatus wird in der Aktualwertspalte wieder der vor dem Programmstatus gültige Inhalt dargestellt. Eine Übernahme der aktualisierten Online-Werte in den Offline-Datenbaustein ist nicht möglich.

Aktualisierung von Datentypen

Alle elementaren Datentypen werden sowohl in einem globalen DB als auch in allen Deklarationen (in/out/inout/stat) eines Instanzdatenbausteins aktualisiert.

Einige Datentypen können nicht aktualisiert werden. Bei aktiviertem Programmstatus werden diese Felder mit nicht aktualisierten Daten in der Spalte "Aktualwert" mit grauem Hintergrund dargestellt.

- Die zusammengesetzten Datentypen DATE_AND_TIME und STRING werden nicht aktualisiert.
- In den zusammengesetzten Datentypen ARRAY, STRUCT, UDT, FB, SFB werden nur diejenigen Elemente aktualisiert, die elementare Datentypen sind.
- In der Inout-Deklaration eines Instanzdatenbausteins wird nur der Pointer auf den zusammengesetzten Datentyp dargestellt aber nicht dessen Elemente. Der Pointer wird nicht aktualisiert.
- Parametertypen werden nicht aktualisiert.

10.3.4 Schritte zum Testen im Programmstatus

10.3.4.1 Festlegen der Anzeigen für den Programmstatus

Sie haben die Möglichkeit, die Anzeige des Programmstatus in einem AWL-, FUP- oder KOP-Baustein selbst festzulegen.

Gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl Extras > Einstellungen....
2. Wählen Sie im Dialogfeld in der Klappliste im obersten Bereich des Dialogfelds "Einstellungen" den Eintrag "Bausteineditor".
3. Wählen Sie die gewünschten Optionen für den Test aus. Folgende Statusfelder können Sie sich anzeigen lassen

Bei Ankreuzen von ...	wird angezeigt:
Statusbit	Statusbit, also Bit 2 des Statusworts
Verknüpfungsergebnis	Bit 1 des Statusworts; gibt das Ergebnis einer Verknüpfung oder eines arithmetischen Vergleichs an.
Akku 1	Inhalt von Akku 1
Adressregister 1/2	Inhalt des entsprechenden Adressregisters bei registerindirekter Adressierung (bereichsintern oder -übergreifend)
Akku 2	Inhalt von Akku 2
DB-Register 1/2	Inhalt des Datenbausteinregisters, des ersten bzw. zweiten geöffneten Datenbausteins
Extras	Indirekte Speicherreferenz; Angabe des Pointers (Adresse), nicht des Adressinhalts; nur bei speicherindirekter Adressierung, nicht bei registerindirekter Adressierung möglich. Inhalt eines Zeitworts oder Zählerworts, wenn entsprechende Operationen in der Anweisung vorkommen
Statuswort	alle Statusbits des Statuswortes

10.3.4.2 Einstellen der Aufrufumgebung des Bausteins

Für die Aufzeichnung des Programmstatus können Sie durch das Festlegen der Aufrufumgebung genaue Bedingungen angeben. Der Programmstatus wird dann nur aufgezeichnet, wenn die angegebene Triggerbedingung erfüllt ist.

Gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Test > Aufrufumgebung**.
2. Legen Sie im angezeigten Dialogfeld die Triggerbedingungen fest und bestätigen Sie diese mit "OK".

Auswahlmöglichkeit	Bedeutung
Aufrufpfad	Hier können Sie den Aufrufpfad angeben, über den der zu testende Baustein aufgerufen werden muss, um eine Statusaufzeichnung auszulösen. Sie können die letzten drei Aufrufebenen vor Erreichen des Testbausteins eingeben.
Mit Adresse	Deaktivieren Sie dieses Optionskästchen, wenn die Aufrufpfadbedingung aufgehoben werden soll.
Offene Datenbausteine	Hier wird die Aufrufumgebung durch Angabe von einem oder zwei Datenbausteinen festgelegt. Die Statusaufzeichnung erfolgt, wenn der zu testende Baustein mit den jeweils angegebenen Datenbausteinen aufgerufen wurde.

Aufrufumgebung für Baustein-Instanzen festlegen

Um den Programmstatus eines Bausteins in einer bestimmten Instanz anzeigen zu lassen, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Test > Betrieb** und stellen Sie den Betriebsmodus "Testbetrieb" ein.
2. Öffnen Sie den aufrufenden Baustein und positionieren Sie den Mauszeiger auf die gewünschte Aufrufanweisung (CALL-Zeile in AWL oder Box des Bausteins in KOP/FUP).
3. Wählen Sie über die rechte Maustaste den Menübefehl **Aufgerufenen Baustein > Beobachten mit Aufrufpfad**.

Ergebnis: Der aufgerufene Baustein wird geöffnet, in die Triggerbedingungen des Bausteins wird der Aufruf als Kriterium eingetragen und der Status für diese Instanz des Bausteins wird aktiviert.

Bereits bestehende Triggerbedingungen für Datenbausteine bleiben unverändert bestehen.

10.3.4.3 Festlegen der Betriebsart für den Test

Testfunktionen verlängern die Zykluszeit des Anwenderprogramms.

Mit der Umschaltung zwischen Testbetrieb und Prozessbetrieb haben Sie Einfluss auf die Zykluszeitbelastung durch Testfunktionen.

Sie können z. B. in der Inbetriebnahmephase den Testbetrieb einstellen, bei dem eine höhere Zykluszeit in Kauf genommen wird.

Im Betrieb können Sie dann eine kürzere Zykluszeit wählen und den Prozessbetrieb einstellen, der dann Einfluss auf die Funktionsweise des Programmstatus hat (siehe unten).

Einstellen der Betriebsart für den Test

Grundsätzlich gibt es zwei Möglichkeiten, die Betriebsart einzustellen, wobei eine CPU immer nur eine Möglichkeit unterstützt:

- Im Rahmen der CPU-Parametrierung im Abschnitt "Betrieb" der CPU-Parameter (z. B. bei S7-300-CPU's). Die Parametrierung muss in die CPU geladen werden, damit die eingestellte Betriebsart aktiv wird.
- Online im Rahmen des Programmtests bei geöffnetem Codebaustein über den Menübefehl **Test > Betrieb**.
Hinweis: Wenn die Umschaltung der Betriebsarten im Rahmen der CPU-Parametrierung erfolgt, wird hier nur die eingestellte Betriebsart angezeigt, sie ist nicht umschaltbar.

Auswirkungen der eingestellten Betriebsart

Betrieb	Erläuterung
Testbetrieb	Alle Testfunktionen sind ohne Einschränkung nutzbar. Größere Verlängerungen der CPU Zykluszeit können auftreten, da z. B. der Status von Anweisungen in programmierten Schleifen bei jedem Durchlauf ermittelt wird.
Prozessbetrieb	Die Testfunktion Programmstatus wird eingeschränkt, um eine möglichst geringe Belastung der Zykluszeit zu gewährleisten. <ul style="list-style-type: none"> • So sind z. B. keine Aufrufbedingungen erlaubt. • Die Statusanzeige einer programmierten Schleife wird an der Rücksprungstelle abgebrochen. • Die Testfunktionen "HALT" und schrittweise Programmausführung können nicht ausgeführt werden.

10.3.4.4 Steuern von Variablen im Programmstatus

Voraussetzung: Der Online-Baustein ist geöffnet.

Die im Anschluss beschriebenen Aktionen bewirken ein einmaliges und sofortiges Steuern der ausgewählten Variablen.

Steuern von Variablen des Datentyps BOOL

1. Markieren Sie den Operanden, den Sie steuern möchten.
2. Wählen Sie den Menübefehl **Test > Steuern auf 1** oder **Test > Steuern auf 0**.

Steuern von nicht-boolschen Variablen

1. Markieren Sie den Operanden, den Sie steuern möchten.
2. Wählen Sie den Menübefehl **Test > Steuern**.
3. Tragen Sie in das angezeigte Dialogfeld den Wert ein, den die Variable annehmen soll (Steuerwert).
4. Schließen Sie das Dialogfeld.

Alternative Vorgehensweise

1. Positionieren Sie den Mauszeiger über den Operand, den Sie steuern möchten.
2. Betätigen Sie die rechte Maustaste und wählen Sie den entsprechenden Befehl zum Steuern aus dem angezeigten Popup-Menü.

10.3.4.5 Ein- und Ausschalten des Tests mit Programmstatus

1. Starten Sie die Aufzeichnung des Programmstatus mit dem Menübefehl **Test > Beobachten** (Häkchen vor dem Menübefehl) oder über einen der beiden Kontextmenübefehle **Aufgerufenen Baustein > Beobachten** bzw. **Aufgerufenen Baustein > Beobachten mit Aufrufpfad**.
2. Werten Sie den AWL-Programmstatus des Bausteins aus, der in Form einer Tabelle angezeigt wird.
3. Die Anzeige können Sie durch erneute Anwahl des Menübefehls **Test > Beobachten** wieder ausschalten (kein Häkchen vor dem Menübefehl).

11 Diagnose

11.1 Diagnosefunktionen

Die Systemdiagnose ist die Erkennung, Auswertung und Meldung von Fehlern, die innerhalb eines Automatisierungssystems auftreten. Dazu gibt es in jeder CPU einen Diagnosepuffer, in den nähere Informationen zu allen Diagnoseereignissen in der Reihenfolge ihres Auftretens eingetragen werden.

Diagnoseereignisse

Als Diagnoseereignisse werden Ihnen folgende Einträge angezeigt, z. B.

- interne und externe Fehler auf einer Baugruppe
- Systemfehler in der CPU
- Betriebszustandsübergänge (z. B. von RUN nach STOP)
- Fehler im Anwenderprogramm
- Ziehen/Stecken von Baugruppen
- mit der Systemfunktion SFC 52 eingetragene Anwendermeldungen

Beim Löschen von CPUs bleibt der Inhalt des Diagnosepuffers erhalten. Fehler im System können durch den Diagnosepuffer auch nach längerer Zeit noch ausgewertet werden, um die Ursache für einen STOP festzustellen oder um das Auftreten einzelner Diagnoseereignisse zurückzuverfolgen und zuordnen zu können.

Aufzeichnung der Diagnosedaten

Die Erfassung von Diagnosedaten durch die Systemdiagnose muss nicht programmiert werden, sie ist standardmäßig vorhanden und läuft automatisch ab. SIMATIC S7 bietet verschiedene Diagnosefunktionen. Einige sind in die CPU integriert, andere werden von den Baugruppen (SMs) zur Verfügung gestellt.

Anzeige von Fehlern

Baugruppeninterne und -externe Fehler werden über LED-Anzeigen auf der Frontplatten der jeweiligen Baugruppe angezeigt. Die LED-Anzeigen und ihre Auswertung ist beschrieben in den Handbüchern zur S7-Hardware. Bei S7-300 werden interne und externe Fehler zu einem Sammelfehler zusammengefasst.

Die CPU erkennt Systemfehler sowie Fehler im Anwenderprogramm und trägt Diagnoseereignisse in die Systemzustandsliste und den Diagnosepuffer ein. Diese Diagnosemeldungen können am PG ausgelesen werden.

Diagnosefähige Baugruppen erkennen interne und externe Baugruppenfehler und erzeugen einen Diagnosealarm, auf den Sie mit Hilfe eines Alarm-OBs reagieren können.

11.2 Diagnose der Hardware und Fehlersuche

Grundsätzliche Vorgehensweise

- Prüfen Sie zunächst, OB die projektierte Konfiguration (das ist die Konfiguration in der Sicht "HW-Konfiguration") mit der geladenen Konfiguration übereinstimmt. Die Gleichheit bzw. Ungleichheit prüfen Sie in der Sicht "HW-Vergleich".
- Prüfen Sie dann, OB zu einzelnen Baugruppen Fehler vorliegen.
- Lassen Sie sich die Fehler zu den Baugruppen ausgeben.

Weitere Diagnosemöglichkeiten in der Diagnosesicht

Durch Doppelklick auf eine Baugruppe können Sie sich den Baugruppenzustand zu dieser Baugruppe anzeigen lassen.

11.3 Vergleich der Konfiguration 'Online/Offline/Physik'

Einführung


"Online-Konfiguration" wird die in die CPU geladene Konfiguration genannt; "Offline-Konfiguration" die projektierte Konfiguration.


Die Konfiguration, die eine CPU von sich aus erkennt, ohne dass überhaupt eine Konfiguration geladen wurde, wird "Physik" genannt.


Konzept des Hardware-Vergleichs

In der Sicht "HW-Vergleich" werden die projektierte (Offline-) Konfiguration, die geladenen (Online-) Konfiguration und die Physik miteinander verglichen. Unterschiede werden in der Konfigurationstabelle durch Symbole gekennzeichnet. In der Konfigurationstabelle bekommen Sie eine Übersicht, welche Baugruppen bzw. Baugruppenparametrierungen nicht übereinstimmen.

Wenn z. B. in einer Zeile eine Digitaleingabebaugruppe projektiert wurde, tatsächlich aber eine Digitalausgabebaugruppe steckt, dann wird das in der betreffenden Zeile (=Steckplatz) durch das Symbol "ungleich" gekennzeichnet:


Symbol für "ungleich": 

Wenn die Baugruppe zwar projektiert, aber online nicht vorhanden ist, dann erscheint folgendes Symbol: 

Wenn die tatsächlich gesteckte Baugruppe der projektierten Baugruppe entspricht, aber andere Baugruppenparameter hat, dann erscheint folgendes Symbol: 

Hinweise

Bei einem Vergleich der Physik mit der Offline-Konfiguration bzw. mit der Online-Konfiguration kann STEP 7 Lite die Identität der Physik nicht 100%ig ermitteln. Der Baugruppentyp kann zwar ermittelt und verglichen werden, aber die genaue Bestellnummer nicht.

In diesem Fall erscheint das Symbol für "vermutlich identische Baugruppen": 

Unterschiede des Hardware-Vergleichs im Detail

Unterhalb der Konfigurationstabelle finden Sie in der "Delta-Liste" eine detaillierte Auflistung der Unterschiede zwischen den Baugruppen, nach Steckplätzen geordnet.

Wenn Sie die Parameter von Baugruppen vergleichen wollen, dann doppelklicken Sie in der Sicht "HW-Vergleich" auf die entsprechende Baugruppe. Der Parametrierdialog, der in diesem Fall schreibgeschützt geöffnet wird, kennzeichnet unterschiedliche Parameter farblich. Die betroffenen Parameterwerte sind gelb unterlegt.

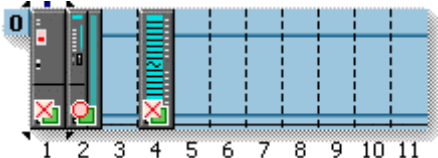
Über die Schaltfläche "Nächster Unterschied" und "Vorheriger Unterschied" können Sie schnell zu den betroffenen Parametern navigieren.

11.4 Aufbau der Sicht 'HW-Vergleich'

Die Sicht "HW-Vergleich" zeigt zwei Ansichten der aktuellen Stationskonfiguration

- Die grafische Sicht mit realitätsnaher Anordnung der Baugruppen auf ihren Steckplätzen sowie Symbolen, die Aufschluss über Gleichheit bzw. Ungleichheit der Konfiguration (Online/Offline/Physik) geben.
- Die tabellarische Sicht mit Detailinformationen zu den gesteckten Baugruppen (z. B. Adressen und Bestellnummern); mit jeweils unterschiedlichen Spalten für die verglichenen Konfigurationen (z. B. Offline - Online).

Außerdem sind Schaltflächen sichtbar, die eine Umschaltung der Vergleichstabellen ermöglichen.

Bereich in HW-Vergleich-Sicht		Bedeutung																																				
		<p>Grafische Sicht der HW-Konfiguration mit markiertem Steckplatz.</p> <p>Vom markierten Steckplatz aus geht ein Pfeil auf den gleichwertigen Steckplatz in der tabellarischen Sicht der HW-Konfiguration. Wenn eine Online-Verbindung zur CPU besteht, dann wird der Synchronisationszustand symbolisch angezeigt (gleich/ungleich).</p> <p>Über die Schaltflächen "Vergleich: ..." können Sie bestimmen, was verglichen werden soll.</p>																																				
<table border="1"> <thead> <tr> <th colspan="6">Offline</th> </tr> <tr> <th>Stpl.</th> <th>Baugruppe</th> <th>Bestellnummer</th> <th></th> <th>Baugruppe</th> <th>Bestellnummer</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>PS307 10A</td> <td>6ES7 307-1KA0...</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>CPU312 C</td> <td>6ES7 312-5BD0...</td> <td></td> <td>CPU314 C-...</td> <td>6ES7 314-6CF0...</td> </tr> <tr> <td>3</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>4</td> <td>SM 331 AI2...</td> <td>6ES7 331-7KB8...</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Offline						Stpl.	Baugruppe	Bestellnummer		Baugruppe	Bestellnummer	1	PS307 10A	6ES7 307-1KA0...				2	CPU312 C	6ES7 312-5BD0...		CPU314 C-...	6ES7 314-6CF0...	3						4	SM 331 AI2...	6ES7 331-7KB8...				<p>Markierter Steckplatz in der tabellarischen Sicht auf die Hardware-Konfiguration.</p> <p>Die je nach Vergleichsart sind die Tabellenspalten unterschiedlich benannt. Zwischen den zu vergleichenden Konfigurationen werden Symbole eingeblendet, die den Synchronisationszustand anzeigen (gleich/ungleich).</p>
Offline																																						
Stpl.	Baugruppe	Bestellnummer		Baugruppe	Bestellnummer																																	
1	PS307 10A	6ES7 307-1KA0...																																				
2	CPU312 C	6ES7 312-5BD0...		CPU314 C-...	6ES7 314-6CF0...																																	
3																																						
4	SM 331 AI2...	6ES7 331-7KB8...																																				
<p>Delta-Liste:</p> <ul style="list-style-type: none"> Rack 0, Steckplatz 1 Baugruppe nur Offline vorhanden Rack 0, Steckplatz 2 Unterschiedliche Baugruppen Offline: 6ES7 312-5BD00-0AB0 Online: 6ES7 314-6CF00-0AB0 Rack 0, Steckplatz 4 Baugruppe nur Offline vorhanden 		<p>Delta-Liste zur Anzeige von Unterschieden.</p> <p>Angezeigt werden die Parameter, deren Werte sich beim Vergleich der Konfigurationen als unterschiedlich herausgestellt haben. Die unterschiedlichen Werte werden in den entsprechenden Spalten angezeigt.</p>																																				

11.5 Gestörte Baugruppen ermitteln

Voraussetzung für die Ermittlung von gestörten Baugruppen ist eine Online-Verbindung zwischen PG/PC und der CPU.

Vorgehensweise

1. Doppelklicken Sie im Projektfenster auf "Hardware".
2. Wählen Sie die Sicht "HW-Diagnose".

Die Sicht "HW-Diagnose" zeigt die Stationskonfiguration, wie sie aus der CPU ermittelt wurde. Das Vorliegen von Diagnoseinformationen für eine Baugruppe können Sie anhand von Diagnosesymbolen erkennen. Diagnosesymbole zeigen den Status der zugehörigen Baugruppe und bei CPUs auch den Betriebszustand an.

Ausführliche Diagnoseinformationen zu jeder Baugruppe werden im Dialog "Baugruppenzustand" angezeigt, den Sie durch Mausklick auf die Schaltfläche "Erweiterte Diagnoseinformationen" aufrufen können.

Aktualisieren der Anzeige

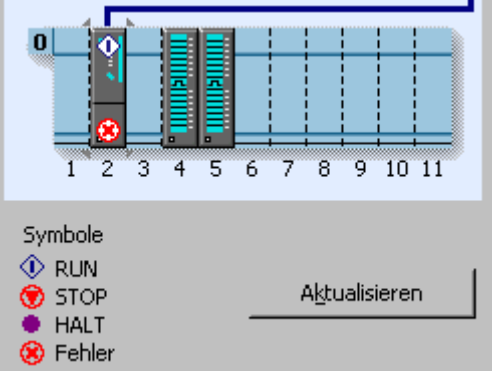
Zum Aktualisieren der Zustandssymbole haben Sie folgende Möglichkeiten:

- Drücken Sie die Funktionstaste F5.
- wählen Sie im Fenster den Menübefehl **Ansicht > Alles Aktualisieren**.
- Klicken Sie auf die Schaltfläche "Ansicht aktualisieren".

11.6 Aufbau der Sicht 'HW-Diagnose'

Die Sicht "HW-Diagnose" zeigt zwei Ansichten der aktuellen Stationskonfiguration

- Die grafische Sicht mit realitätsnaher Anordnung der Baugruppen auf ihren Steckplätzen
- Die tabellarische Sicht mit Detailinformationen zu den gesteckten Baugruppen (z. B. Adressen und Bestellnummern)

Bereich in HW-Diagnose-Sicht	Bedeutung																														
	<p>Grafische Sicht der HW-Konfiguration mit markiertem Steckplatz/markierter Baugruppe.</p> <p>Von der markierten Baugruppe aus geht ein Pfeil auf den gleichwertigen Steckplatz in der tabellarischen Sicht der HW-Konfiguration. Der Zustand der Baugruppen im Zielsystem wird durch Symbole angezeigt.</p> <p>Über die Schaltfläche "Aktualisieren" können Sie den Zustand der Symbole aktualisieren lassen.</p>																														
<table border="1" data-bbox="247 987 917 1216"> <thead> <tr> <th>Stpl.</th> <th>Baugr.-Status</th> <th>Baugruppe</th> <th>Bestellnummer</th> <th>Basisadresse</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>✖ Fehler</td> <td>CPU 314</td> <td>6ES7 314-1AE8...</td> <td></td> </tr> <tr> <td>3</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>4</td> <td>O.K.</td> <td>SM 32* D...</td> <td>6ES7 32* Stan...</td> <td>E 0</td> </tr> <tr> <td>5</td> <td>O.K.</td> <td>SM 32* DO</td> <td>6ES7 32* Stan...</td> <td>A 4</td> </tr> </tbody> </table>	Stpl.	Baugr.-Status	Baugruppe	Bestellnummer	Basisadresse	1					2	✖ Fehler	CPU 314	6ES7 314-1AE8...		3					4	O.K.	SM 32* D...	6ES7 32* Stan...	E 0	5	O.K.	SM 32* DO	6ES7 32* Stan...	A 4	<p>Markierter Steckplatz/markierte Baugruppe in der tabellarischen Sicht auf die Hardware-Konfiguration.</p> <p>Der Status der Baugruppen wird in der Spalte "Baugruppen-Status" angezeigt.</p> <p>Über die Register am oberen Rand kann auf verschiedene Baugruppenträger zugegriffen werden.</p>
Stpl.	Baugr.-Status	Baugruppe	Bestellnummer	Basisadresse																											
1																															
2	✖ Fehler	CPU 314	6ES7 314-1AE8...																												
3																															
4	O.K.	SM 32* D...	6ES7 32* Stan...	E 0																											
5	O.K.	SM 32* DO	6ES7 32* Stan...	A 4																											
<p>Baugruppe: CPU 314 Systemkennung: SIMATIC</p> <table border="1" data-bbox="247 1413 965 1507"> <thead> <tr> <th>Version:</th> <th>Bestell-Nr. / Bezeichn.</th> <th>Komponente</th> <th>Ausgabestand</th> </tr> </thead> <tbody> <tr> <td></td> <td>6ES7 314-1AE84-0AB0</td> <td>Hardware</td> <td>1</td> </tr> <tr> <td></td> <td>---</td> <td>Firmware</td> <td>V1.0.0</td> </tr> </tbody> </table> <p>Rack: 0 Adresse:</p> <p>Steckplatz: 2 Baugruppenbreite:</p> <p>Status: Baugruppe gestört</p> <p style="text-align: center;">Erweiterte Diagnoseinformationen...</p>	Version:	Bestell-Nr. / Bezeichn.	Komponente	Ausgabestand		6ES7 314-1AE84-0AB0	Hardware	1		---	Firmware	V1.0.0	<p>Detailinformation zur markierten Baugruppe in der HW-Diagnose-Sicht.</p> <p>Über die Schaltfläche "Erweiterte Diagnoseinformationen" können Sie den Baugruppenzustands-Dialog öffnen, um z. B. den Diagnosepuffer auszulesen.</p>																		
Version:	Bestell-Nr. / Bezeichn.	Komponente	Ausgabestand																												
	6ES7 314-1AE84-0AB0	Hardware	1																												
	---	Firmware	V1.0.0																												

11.7 Baugruppenzustand

11.7.1 Aufruf des Baugruppenzustands

Baugruppenzustand aus dem CPU-Bedienpanel aufrufen

Unabhängig von der eingestellten Sicht ist der Baugruppenzustand am CPU-Bedienpanel sichtbar.

Baugruppenzustand aus der Sicht "HW-Diagnose" aufrufen

1. Doppelklicken Sie im Projektfenster auf das Symbol "Hardware".
2. Wählen Sie das Register "HW-Diagnose".
3. Markieren Sie eine als gestört angezeigte Baugruppe.
4. Wählen Sie den Menübefehl **Extras > Baugruppenzustand** oder klicken Sie auf die Schaltfläche "Erweiterte Diagnoseinformationen".

Ergebnis

Der Registerdialog "Baugruppenzustand" wird für CPUs und für diagnosefähige Baugruppen angezeigt. Je nach Diagnosefähigkeit der Baugruppe wird eine unterschiedliche Anzahl von Registerseiten im Dialogfeld "Baugruppenzustand" angezeigt. Bei nicht-diagnosefähigen Baugruppen werden Sie die Zustandsinformationen aus der Sicht 'HW-Diagnose' aus.

Beispiel: Baugruppenzustand einer CPU

Folgende Informationen werden z. B. im oberen Teil des Baugruppenzustandsdialogs einer CPU angezeigt:

- Betriebszustand der CPU (z. B. RUN),
- Baugruppenzustand (z. B. OB ein Fehler vorliegt)
- Forceauftrag (aktiv oder nicht aktiv)

Darunter befinden sich die Register des Baugruppenzustands; hier mit Register "Diagnosepuffer" im Vordergrund. Je nach Typ der markierten Baugruppe, zu dem der Baugruppenzustand aufgerufen wurde, ist eine unterschiedliche Anzahl von Registern sichtbar.

Im Register "Diagnosepuffer" finden Sie die Einträge (=Ereignisse) in zeitlicher Reihenfolge. Der jüngste Eintrag steht immer oben (Eintrag Nr. 1). Im Feld darunter stehen die Details zum markierten Ereignis.

Über Optionskästchen können Sie die Ansichtseinstellungen für die Ansicht des Dialogs beeinflussen (Filter und hexacodierte Anzeige der Ereignisse).

Die Schaltflächen erlauben das **Speichern** im Textformat (*.txt), das Ändern von **Einstellungen** (z. B. für den Filter) sowie die Anzeige der **Hilfe** zum markierten Ereignis, das durch die hexadezimal codierte Ereignis-ID gekennzeichnet ist. Nur bei Anzeigen, die auf einen Baustein referenzieren, kann der Baustein direkt zur weiteren Bearbeitung über die Schaltfläche **Baustein öffnen** geöffnet werden.

Baugruppenzustand - CPU 314 ONLINE

Betriebszustand der CPU: **RUN** Baugruppenzustand: **Fehler**

Forceauftrag: **aktiv**

Diagnosepuffer | Stacks | Speicher | Zykluszeit | Zeitsystem | Leistungsdaten | Kommunikation | Allgemein

Ereignisse: Filter-Einstellungen aktiv Ereignisse hexcodiert anzeigen

Nr.	Uhrzeit	Datum	Ereignis
1	12:37:33:505	13.11.00	Betriebszustandsübergang von ANLAUF nach RUN
2	12:37:33:505	13.11.00	Manuelle Neustart (Warmstart)-Anforderung
3	12:37:33:472	13.11.00	Betriebszustandsübergang von STOP nach ANLAUF
4	12:37:29:143	13.11.00	STOP durch PG-Stop-Bedienung oder wegen SFB 20 "STOP"
5	10:14:29:772	13.11.00	Betriebszustandsübergang von ANLAUF nach RUN
6	10:14:29:772	13.11.00	Manuelle Neustart (Warmstart)-Anforderung
7	10:14:29:742	13.11.00	Betriebszustandsübergang von STOP nach ANLAUF
8	10:14:24:669	13.11.00	Urlöschen durchgeführt

Details zum Ereignis: 1 von 10 Ereignis-ID: 16# 4302

Betriebszustandsübergang von **ANLAUF** nach **RUN**

Anlaufinformation:

- Uhr für Zeitstempel bei letztem NETZ-EIN nicht gepuffert
- Einprozessorbetrieb

Aktuelle/letzte durchgeführte Anlaufart:

- Neustart (Warmstart) durch MPI-Bedienung: letzter NETZ-EIN gepuffert

Speichern unter... Einstellungen... Baustein öffnen Hilfe zum Ereignis

Hilfe Aktualisieren Drucken... Schließen

11.7.2 AuskunftsFunktionen des Baugruppenzustands

Die AuskunftsFunktionen stehen auf der gleichnamigen Registerseite des Dialogfelds "Baugruppenzustand" zur Verfügung. Im konkreten Anwendungsfall werden nur die Registerseiten angezeigt, die für die ausgewählte Baugruppe sinnvoll sind.

AuskunftsFunktion	Information	Einsatz
Allgemein	Identifikationsdaten der ausgewählten Baugruppe, z. B. Typ, Bestellnummer, Ausgabestand, Status, Steckplatz im Baugruppenträger.	Die Online-Auskunft aus der gesteckten Baugruppe kann mit den Daten der projizierten Baugruppe verglichen werden.
Diagnosepuffer	Übersicht über Ereignisse im Diagnosepuffer sowie detaillierte Information zum ausgewählten Ereignis.	Zum Auswerten der STOP-Ursache einer CPU und zum Auswerten der vorangegangenen Ereignisse auf der ausgewählten Baugruppe. Fehler im System können durch den Diagnosepuffer auch nach längerer Zeit noch ausgewertet werden, um die Ursache für einen STOP festzustellen oder um das Auftreten einzelner Diagnoseereignisse zurückzuverfolgen und zuordnen zu können.
Diagnosealarm	Diagnosedaten der ausgewählten Baugruppe.	Zur Ermittlung der Ursache einer Baugruppenstörung.
Speicher	Speicherausbau, Aktuelle Auslastung von Arbeitsspeicher und Ladespeicher der gewählten CPU.	Vor dem Übertragen neuer oder erweiterter Bausteine auf eine CPU um zu prüfen, OB genügend Ladespeicher in dieser CPU/FM zur Verfügung steht sowie zum Komprimieren des Speicherinhalts.
Zykluszeit	Dauer des längsten, kürzesten und letzten Zyklus der gewählten CPU.	Zur Kontrolle der parametrisierten Mindestzykluszeit sowie der maximalen und aktuellen Zykluszeit.
Zeitsystem	Aktuelle Uhrzeit, Betriebsstunden und Informationen zur Uhrensynchronisation (Synchronisationsintervalle).	Zur Anzeige von Uhrzeit und Datum einer Baugruppe und zur Kontrolle der Uhrzeitsynchronisation.
Leistungsdaten	Operandenbereiche und die verfügbaren Bausteine der angewählten (CPU-/FM-) Baugruppe.	Vor und während der Erstellung eines Anwenderprogramms und zur Überprüfung, OB die CPU geeignete Voraussetzungen zum Ablauf eines Anwenderprogramms besitzt, z. B. bezüglich Größe des Prozessabbilds.
Leistungsdaten (Fortsetzung)	Anzeige aller Bausteinarten, die im Funktionsumfang der angewählten Baugruppe verfügbar sind. Liste der OBs, SFBs und SFCs, die in dieser Baugruppe eingesetzt werden können.	Zur Prüfung, welche Standard-Bausteine Ihr Anwenderprogramm enthalten bzw. aufrufen darf, um auf der angewählten CPU ablaufen zu können.

Auskunftsfunktion	Information	Einsatz
Kommunikation	Baudraten, die Verbindungsübersicht, die Kommunikationsbelastung sowie die maximale Größe der Telegramme auf dem K-Bus der angewählten Baugruppe.	Um festzustellen, wieviel und welche Verbindungen der CPU möglich bzw. belegt sind.
Stacks	Register Stacks : kann nur im Betriebszustand STOP oder HALT aufgeblättert werden. Der B-Stack der angewählten Baugruppe wird angezeigt. Sie können außerdem U-Stack und L-Stack auslesen und an die Fehlerstelle des unterbrechenden Bausteins springen.	Zur Feststellung der Ursache eines Übergangs in STOP und zur Korrektur eines Bausteins.

Angezeigte Zusatzinformation

Folgende Informationen werden immer dargestellt:

- OB ein Forceauftrag aktiv ist (nur bei CPUs, die 'Forcen' unterstützen)
- Betriebszustand der zugehörigen CPU (z. B. RUN, STOP)
- Status der ausgewählten Baugruppe (z. B. Fehler, o.k.)

Gleichzeitig mehrere Baugruppen anzeigen

Sie können den Baugruppenzustand verschiedener Baugruppen abrufen und gleichzeitig darstellen. Dazu müssen Sie in den entsprechenden Baugruppenkontext wechseln, eine andere Baugruppe auswählen und den Baugruppenzustand dazu aufrufen. Es wird Ihnen dann ein weiterer Registerdialog angezeigt. Zu einer Baugruppe kann jedoch nur ein Registerdialog geöffnet werden.

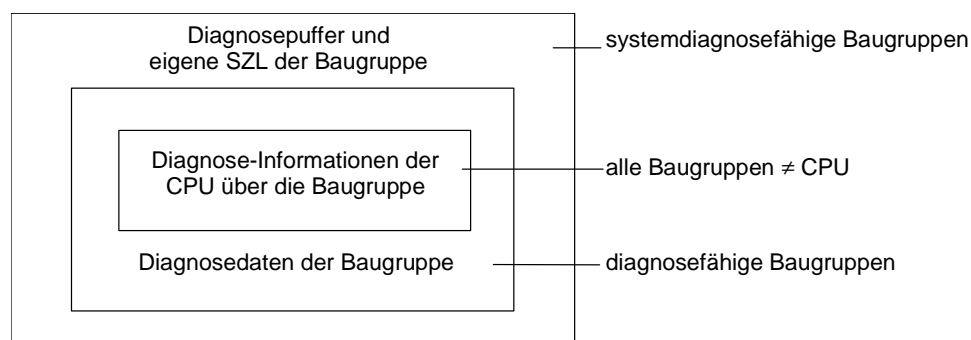
Aktualisieren der Anzeigen im Baugruppenzustand

Bei jedem Wechsel auf eine andere Registerseite des Dialogfelds "Baugruppenzustand" werden die Daten neu von der Baugruppe gelesen. Während der Anzeige einer Seite wird der Inhalt jedoch nicht automatisch aktualisiert. Durch Klicken auf die Schaltfläche "Aktualisieren" werden die Daten erneut von der Baugruppe gelesen, ohne die Registerseite zu wechseln.

11.7.3 Baugruppentyp-abhängiger Umfang der Auskunftsfunktionen im Baugruppenzustand

Der Umfang der Informationen, die ausgewertet und angezeigt werden können, ist abhängig von der angewählten Baugruppe.

Je nach Umfang der Informationen werden die Baugruppen eingeteilt in systemdiagnosefähige, diagnosefähige oder nicht diagnosefähige Baugruppen. Diesen Zusammenhang veranschaulicht die folgende Grafik:



- Systemdiagnosefähig sind komplexe Baugruppen wie z. B. die Baugruppen FM351 und FM354. Diese Baugruppen haben einen Diagnosepuffer und verwalten intern eine Systemzustandsliste (SZL).
- Diagnosefähig sind Baugruppen, die einen Diagnosealarm auslösen können; das sind die meisten Analogbaugruppen.
- Nicht diagnosefähig sind Baugruppen, die keinen Diagnosealarm auslösen können; das sind die meisten Digitalbaugruppen.

Angezeigte Register

Die Tabelle zeigt, welche Register bei den einzelnen Baugruppentypen im Dialogfeld "Baugruppenzustand" vorhanden sind.

Register	CPU	Systemdiagnosefähige Baugruppe	Diagnosefähige Baugruppe	Nicht diagnosefähige Baugruppe
Allgemein	ja	ja	ja	–
Diagnosepuffer	ja	ja	–	–
Diagnosealarm	–	ja	ja	–
Speicher	ja	–	–	–
Zykluszeit	ja	–	–	–
Zeitsystem	ja	–	–	–
Leistungsdaten	ja	–	–	–
Stacks	ja	–	–	–
Kommunikation	ja	–	–	–

Neben der Information auf den Registern wird der Betriebszustand bei Baugruppen mit Betriebszustand angezeigt. Außerdem wird der Status angegeben, den diese Baugruppe aus Sicht der CPU besitzt (z. B. OK, Fehler, Baugruppe nicht vorhanden).

11.8 Diagnose im Betriebszustand STOP

11.8.1 Prinzipielles Vorgehen zum Ermitteln einer STOP-Ursache

Um zu ermitteln, warum die CPU in den Betriebszustand "STOP" gewechselt hat, gehen Sie folgendermaßen vor:

1. Markieren Sie in der Sicht "HW-Diagnose" die CPU, die in den Betriebszustand "STOP" gegangen ist.
2. Klicken Sie auf die Schaltfläche "Erweiterte Diagnoseinformationen".
3. Wählen Sie das Register "Diagnosepuffer".
Anhand der letzten Einträge können Sie die STOP-Ursache ermitteln.

Beispiel: Programmierfehler

Der Eintrag "STOP durch nicht geladenen Programmierfehler-OB" bedeutet beispielsweise, dass die CPU einen Programmierfehler erkannt und danach versucht hat, den (nicht vorhandenen) OB zur Behandlung des Programmierfehlers zu starten. Auf den eigentlichen Programmierfehler weist der vorhergehende Eintrag hin.

1. Markieren Sie die Meldung zum Programmierfehler.
2. Klicken Sie auf die Schaltfläche "Baustein öffnen".
3. Wählen Sie die Registerseite "Stacks".

11.8.2 Stack-Inhalte im Betriebszustand STOP

Durch Auswertung des Diagnosepuffers und der Stack-Inhalte können Sie die Ursache für die Störung im Ablauf eines Anwenderprogrammes ermitteln.

Wenn die CPU z. B. durch einen Programmierfehler oder durch den Stoppbefehl in den Betriebszustand "STOP" gewechselt hat, wird auf der Registerseite "Stacks" des Baugruppenzustands der Baustein-Stack angezeigt. Weitere Stack-Inhalte können Sie über die Schaltflächen "U-Stack" und "L-Stack" anzeigen. Die Stack-Inhalte geben Ihnen Informationen darüber, welcher Befehl in welchem Baustein zu dem STOP bei der CPU geführt hat.

B-Stack-Inhalt

Im B-Stack werden alle Bausteine aufgeführt, die bis zum Übergang in den STOP-Zustand aufgerufen wurden und noch nicht zu Ende bearbeitet worden sind.

U-Stack-Inhalt

Durch Klicken auf die Schaltfläche "U-Stack" werden Ihnen Daten zur Unterbrechungsstelle angezeigt. Der Unterbrechungsstack (U-Stack) enthält Daten bzw. Zustände, die zum Zeitpunkt der Unterbrechung gültig waren, z. B.

- Akku-Inhalte und Registerinhalte
- Offene DBs und deren Größe
- Inhalt des Statusworts
- Prioritätsklasse
- Unterbrochenen Baustein
- Baustein, in dem der Programmablauf nach der Unterbrechung fortgesetzt würde.

L-Stack-Inhalt

Für jeden im B-Stack aufgelisteten Baustein können Sie sich durch Selektieren des Bausteins und Klicken auf die Schaltfläche "L-Stack" die zugehörigen Lokaldaten anzeigen lassen.

Der Lokaldaten-Stack (L-Stack) enthält die Lokaldatenwerte der Bausteine, mit denen das Anwenderprogramm bis zum Zeitpunkt der Unterbrechung gearbeitet hat.

Für die Interpretation und Auswertung der angezeigten Lokaldaten sind sehr gute Systemkenntnisse erforderlich. Der vordere Teil der angezeigten Daten entspricht den temporären Variablen des Bausteins.

11.8.3 Öffnen des Bausteins zu einem Diagnosepuffer- oder Stack-Eintrag

11.8.3.1 Öffnen des Bausteins zu einem Diagnosepuffereintrag

Bei Diagnosepuffereinträgen, die auf eine Fehlerstelle (Bausteintyp, -nummer, Relativadresse) verweisen, können Sie den zugehörigen, ereignisverursachenden Baustein öffnen, um die Fehlerursache zu beheben.

1. Wählen Sie das Diagnoseereignis im oberen Listenfeld aus.
2. Klicken Sie auf die Schaltfläche "Baustein öffnen". Der Baustein wird im Editor (z. B. in AWL) geöffnet und der Mauszeiger steht an der ereignisverursachenden Programmstelle.
3. Korrigieren Sie den Fehler im Baustein.

Achtung

Der Diagnosepuffer speichert bis zur maximalen Anzahl alle Diagnoseeinträge. Diese bleiben auch erhalten, wenn zwischenzeitlich ein anderes Anwenderprogramm geladen wurde.

Daher kann es vorkommen, dass ältere Diagnoseeinträge sich auf Bausteine beziehen, die nicht mehr in der CPU vorhanden sind. Im ungünstigen Fall liegt unter gleichem Bausteinnamen ein Baustein auf der CPU, der nicht zur Diagnosemeldung gehört.

In seltenen Fällen können folgende Konstellationen auftreten:

- Das Diagnoseereignis ist älter als das Datum der letzten Bausteinänderung:
 - Sie erhalten das Dialogfeld "Baustein öffnen" mit dem Hinweis, dass der Baustein zwischenzeitlich geändert wurde. Es kann auch sein, dass es lediglich ein namensgleicher Baustein eines anderen Programms ist.
 - Sie können den Baustein offline im entsprechenden Programm auswählen und offline ändern.
- Der ereignisverursachende Baustein ist nicht mehr auf der CPU:
 - Sie erhalten das Dialogfeld "Baustein öffnen" mit dem Hinweis, dass der referenzierte Baustein in der CPU nicht vorhanden ist. Nach dem Zeitpunkt des Diagnoseeintrags wurde der Baustein gelöscht.

Sie können den Baustein offline im entsprechenden Programm auswählen und offline ändern.

11.8.3.2 Öffnen des Bausteins aus der B-Stack-Liste

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf die Schaltfläche "Baustein öffnen". Der Baustein wird online geöffnet. Die Einfügemarke steht an der Stelle, an der die Bearbeitung nach dem Rücksprung aus dem aufgerufenen Baustein fortgesetzt werden würde.
2. Öffnen Sie den Baustein offline (aus dem Projektfenster) und führen Sie im Programmeditor Ihre Änderungen durch.

11.8.3.3 Öffnen des Bausteins aus der U-Stack-Liste

Gehen Sie folgendermaßen vor:

1. Klicken Sie auf die Schaltfläche "Baustein öffnen". Der Baustein wird online geöffnet. Die Einfügemarke steht an der fehlerverursachenden Programmstelle.
2. Öffnen Sie den Baustein offline (aus dem Projektfenster) und führen Sie im Programmeditor Ihre Änderungen durch.

11.9 Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern

11.9.1 Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern

Die Registerseite "Zykluszeit" des Baugruppenzustands gibt Ihnen Auskunft über die Zykluszeiten des Anwenderprogramms.

Wenn die Dauer des längsten Zyklus dicht an der Überwachungszeit liegt, besteht die Gefahr, dass Schwankungen in der Zykluszeit zu einem Zeitfehler führen. Dies können Sie vermeiden, indem Sie die maximale Zykluszeit des Anwenderprogramms vergrößern.

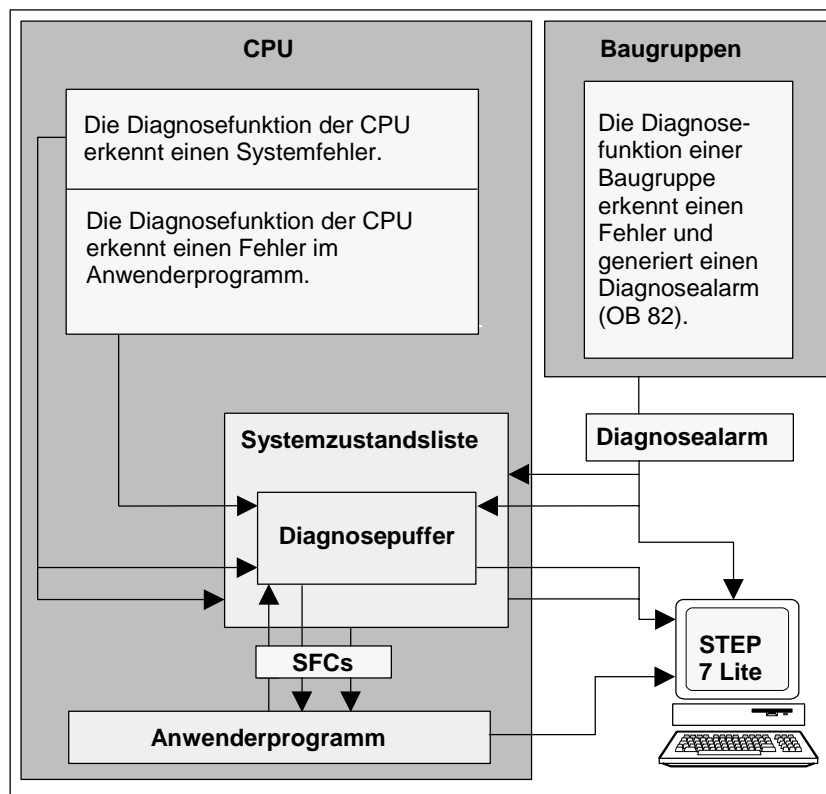
Einstellen der Zykluszeit

Die maximale und minimale Zykluszeit können Sie bei der Konfigurierung der Hardware festlegen. Dazu markieren Sie in der Sicht "HW-Konfiguration" die CPU aus und wählen über die rechte Maustaste den Menübefehl **Baugruppenparameter**. Im Abschnitt "Zyklus" können Sie die Werte einstellen.

11.10 Übermittlung von Diagnoseinformationen

11.10.1 Übermittlung von Diagnoseinformationen

Das folgende Bild zeigt die Übermittlung von Diagnoseinformationen bei SIMATIC S7.



Auslesen der Diagnoseinformation

Sie können die Diagnoseeinträge im Anwenderprogramm mit der SFC 51 RDSYSST auslesen oder sich die Diagnosemeldungen mit STEP 7 Lite im Klartext anzeigen lassen.

Sie geben Auskunft darüber:

- wo und wann der Fehler aufgetreten ist
- zu welcher Art von Diagnoseereignissen der Eintrag gehört (anwenderdefiniertes Diagnoseereignis, synchroner/asynchroner Fehler, Betriebszustandswechsel).

11.10.2 Systemzustandsliste SZL

Die Systemzustandsliste SZL beschreibt den aktuellen Zustand des Automatisierungssystems: sie gibt einen Überblick über den Ausbau, die aktuelle Parametrierung, die aktuellen Zustände und Abläufe in der CPU und den zugeordneten Baugruppen.

Die Daten der SZL können nur gelesen, nicht geändert werden. Sie ist eine virtuelle Liste, die nur auf Anforderung zusammengestellt wird.

Die Informationen, die Sie über die SZL ausgeben können, lassen sich in vier Bereiche aufteilen:



Systemzustandsliste lesen

Es gibt zwei Möglichkeiten, die Informationen der Systemzustandsliste SZL zu lesen:

- implizit über STEP 7 Lite-Menübefehle vom Programmiergerät aus (z. B. Speicherausbau, statische CPU-Daten, Diagnosepuffer, Zustandsanzeigen).
- explizit über die Systemfunktion SFC 51 RDSYSST vom Anwenderprogramm aus mittels Angabe der gewünschten Teillistennummer (siehe Baustein-Hilfen).

Systemdaten der SZL

Systemdaten sind feste oder parametrisierte Kenndaten einer CPU. Die folgende Tabelle zeigt, zu welchen Themenbereichen Informationen (Teillisten der SZL) ausgegeben werden können.

Bereich	Information
Baugruppen-Identifikation	Bestellnummer, Typkennung und Ausgabestand der Baugruppe
CPU-Merkmale	Zeitsystem und Sprachbeschreibung der CPU
Speicherbereiche	Speicherausbau der Baugruppe (Größe des Arbeitsspeichers)
Systembereiche	Systemspeicher der Baugruppe (z. B. Anzahl der Merker, Zeiten, Zähler, Speichertyp)
Bausteintypen	Welche Bausteintypen (OB, DB, SDB, FC, FB) in Baugruppe vorhanden, wie viele Bausteine eines Typs maximal vorhanden und maximale Größe eines Bausteintyps
Alarm-Fehlerzuordnung	Zuordnung Alarme/Fehler zu OBs
Alarmstatus	Aktuelle Alarmbearbeitung/-generierung
Status der Prioritätsklassen	Welcher OB in Bearbeitung, welche Prioritätsklasse durch Parametrierung gesperrt
Betriebszustand und Betriebszustandsübergang	Welche Betriebszustände möglich, letzter Übergang, aktueller Betriebszustand

Diagnosezustandsdaten in der CPU

Diagnosezustandsdaten beschreiben den aktuellen Zustand der Komponenten, die durch die Systemdiagnose überwacht werden. Die folgende Tabelle zeigt, zu welchen Themenbereichen Informationen (Teillisten der SZL) ausgegeben werden können:

Bereich	Information
Kommunikationszustandsdaten	Aktuell im System eingestellte Kommunikationsfunktionen
Diagnoseteilnehmer	An der CPU angemeldete diagnosefähige Baugruppen
Startinformationsliste des OB	Startinformationen zu den OBs der CPU
Startereignisliste	Startereignisse und Prioritätsklassen der OBs
Baugruppenzustandsinformation	Zustandsinformationen aller gesteckten, gestörten, prozessalarmerzeugenden, zugeordneten Baugruppen

Diagnosedaten auf Baugruppen

Außer der CPU gibt es noch andere diagnosefähige Baugruppen (SMs, CPs, FMs), deren Diagnosedaten in die Systemzustandsliste eingetragen werden. Die folgende Tabelle zeigt, zu welchen Themenbereichen Informationen (Teillisten der SZL) ausgegeben werden können.

Bereich	Information
Baugruppendiagnoseinfo	Baugruppenanfangsadresse, interne/externe Fehler, Kanalfehler, Parameterfehler (4 Byte)
Baugruppendiagnosedaten	Alle Diagnosedaten einer bestimmten Baugruppe

11.10.3 Eigene Diagnosemeldungen senden

Sie können die standardmäßige Systemdiagnose von SIMATIC S7 außerdem erweitern, indem Sie über die Systemfunktion SFC 52 WR_USMSG

- anwenderdefinierte Diagnoseereignisse (z. B. Informationen zum Ablauf des Anwenderprogramms) in den Diagnosepuffer eintragen
- anwenderdefinierte Diagnosemeldungen an angemeldete Teilnehmer (Beobachtungsgeräte wie PG, OP, TD) senden.

Anwenderdefinierte Diagnoseereignisse

Die Diagnoseereignisse sind in Ereignisklassen von 1 bis F eingeteilt. Die anwenderdefinierten Diagnoseereignisse gehören zu den Ereignisklassen 8 bis B. Sie lassen sich in zwei Gruppen unterteilen:

- Die Ereignisklassen 8 und 9 umfassen Ereignisse mit festgelegter Nummer und vorgefertigtem Text, den Sie anhand der Nummer abrufen können.
- Die Ereignisklassen A und B umfassen Ereignisse mit frei wählbarer Nummer (A000 bis A0FF, B000 bis B0FF) und frei wählbarem Text.

Diagnosemeldungen an Teilnehmer senden

Zusätzlich zum Eintrag eines anwenderdefinierten Diagnoseereignisses in den Diagnosepuffer können Sie mit der SFC 52 WR_USMSG Ihre anwenderdefinierten Diagnosemeldungen auch noch an angemeldete Anzeigegeräte senden. Bei Aufruf der SFC 52 mit SEND = 1 wird diese Diagnosemeldung in den Sendepuffer geschrieben und automatisch an den oder die an der CPU angemeldeten Teilnehmer geschickt.

Ist ein Senden nicht möglich (z. B. weil kein Teilnehmer angemeldet wurde oder der Sendepuffer voll ist), erfolgt dennoch der Eintrag des anwenderdefinierten Diagnoseereignisses in den Diagnosepuffer.

Meldung mit Quittieranzeige erzeugen

Falls Sie ein anwenderdefiniertes Diagnoseereignis quittieren und diese Quittierung programmtechnisch erfassen wollen, gehen Sie folgendermaßen vor:

- Beim kommenden Ereignis beschreiben Sie eine Variable vom Typ BOOL mit 1, beim gehenden Ereignis beschreiben Sie diese mit 0.
- Diese Variable überwachen Sie mit Hilfe des SFB 33 ALARM.

11.11 Maßnahmen im Programm zur Störungsbehandlung

Die CPU ruft bei Erkennung von Fehlern im Programmablauf (Synchronfehler) und von Fehlern im Automatisierungssystem (Asynchronfehler) den zu dem jeweiligen Fehler gehörenden Fehler-OB auf:

Aufgetretener Fehler	Fehler-OB
Zeitfehler	OB 80
Stromversorgungsfehler	OB 81
Diagnosealarm	OB 82
CPU-Hardwarefehler	OB 84
Programmablauffehler	OB 85
Baugruppenträgerausfall oder Ausfall einer Station bei dezentraler Peripherie	OB 86
Kommunikationsfehler	OB 87
Programmierfehler	OB 121
Peripheriezugriffsfehler	OB 122

Ist der entsprechende OB nicht vorhanden, geht die CPU in den Betriebszustand "STOP". Andernfalls besteht die Möglichkeit, in dem OB Anweisungen zu hinterlegen, wie auf diese Fehlersituation reagiert werden soll. Dadurch können evtl. die Auswirkungen des Fehlers gemindert oder aufgehoben werden.

Prinzipielle Vorgehensweise

Anlegen und Öffnen des OB

1. Rufen Sie den Baugruppenzustand Ihrer CPU auf.
2. Wählen Sie das Register "Leistungsdaten".
3. Entnehmen Sie der angezeigten Liste, OB der zu programmierende OB für diese CPU zulässig ist.
4. Fügen Sie den OB in den Ordner "Bausteine" Ihres Programms ein und öffnen Sie diesen.
5. Geben Sie das Programm zur Behandlung des Fehlers ein.
6. Laden Sie den OB auf das Zielsystem.

Programmieren der Maßnahmen zur Fehlerbehandlung

1. Werten Sie die Lokaldaten des OB aus, um genauere Fehlerursachen zu ermitteln.
Die Variablen OB8x_FLT_ID bzw. OB12x_SW_FLT der Lokaldaten enthalten den Fehlercode. Die Bedeutung ist im Referenzhandbuch System- und Standardfunktionen beschrieben.
2. Verzweigen Sie in den Programmteil, mit dem auf diesen Fehler reagiert wird.

Ein Beispiel zur Behandlung von Diagnosealarmen enthält die Referenzhilfe zu System- und Standardfunktionen unter dem Titel "Beispiel zur Baugruppendiagnose mit der SFC 51 (RDSYSST)".

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.1 Auswerten des Ausgangsparameter RET_VAL

Eine Systemfunktion zeigt über den Ausgangsparameter RET_VAL (Rückgabewert) an, OB die CPU die Funktion der SFC erfolgreich ausführen konnte oder nicht.

Fehlerinformationen im Rückgabewert

Der Rückgabewert ist vom Datentyp Ganzzahl (INT). Das Vorzeichen einer Ganzzahl gibt an, OB es sich um eine positive oder negative Ganzzahl handelt. Die Relation des Rückgabewerts zu dem Wert "0" zeigt an, OB während der Bearbeitung der Funktion ein Fehler aufgetreten ist (siehe auch nachfolgende Tabelle):

- Tritt während der Bearbeitung der Funktion ein Fehler auf, ist der Rückgabewert kleiner als Null. Das Vorzeichenbit der Ganzzahl ist "1".
- Wird die Funktion fehlerfrei bearbeitet, ist der Rückgabewert größer gleich Null. Das Vorzeichenbit der Ganzzahl ist "0".

Bearbeitung der SFC durch die CPU	Rückgabewert	Vorzeichen der Ganzzahl
fehlerhaft	kleiner als "0"	negativ (Vorzeichenbit ist "1")
fehlerfrei	größer oder gleich "0"	positiv (Vorzeichenbit ist "0")

Reagieren auf Fehlerinformationen

Tritt während der Bearbeitung einer SFC ein Fehler auf, dann stellt die SFC über den Rückgabewert RET_VAL einen Fehlercode zur Verfügung.

Man unterscheidet dabei zwischen:

- einem allgemeinen Fehlercode, den alle SFCs ausgeben können und
- einem spezifischen Fehlercode, den eine SFC abhängig von seinen spezifischen Funktionen ausgeben kann.

Funktionswertübergabe

Einige SFCs benutzen den Ausgangsparameter RET_VAL auch zur Übergabe des Funktionswertes, z. B. übergibt die SFC 64 TIME_TCK mit RET_VAL die gelesene Systemzeit.

Ausführliche Informationen zum Parameter RET_VAL finden Sie in der Hilfe zu SFBs/SFCs.

11.11.2 Fehler-OBs als Reaktion auf die Erkennung eines Fehlers

Feststellbare Fehler

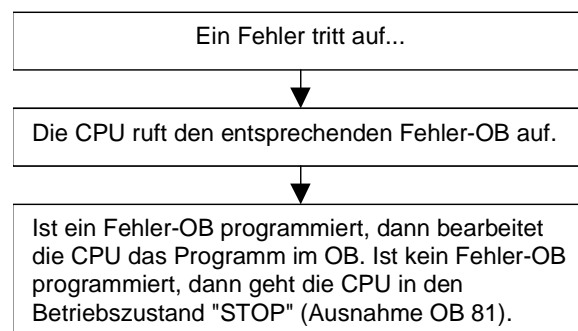
Das Systemprogramm kann folgende Fehler feststellen:

- fehlerhaftes Arbeiten der CPU
- Fehler in der Systemprogrammbearbeitung
- Fehler im Anwenderprogramm
- Fehler in der Peripherie

Je nach Fehlerart wird die CPU in den STOP-Zustand gesetzt oder ein Fehler-OB aufgerufen.

Programmieren von Reaktionen

Sie können Programme entwerfen, um auf die verschiedenen Arten von Fehlern zu reagieren und das weitere Verhalten der CPU zu bestimmen. Das Programm für einen bestimmten Fehler kann dann in einem Fehler-OB gespeichert werden. Wenn der Fehler-OB aufgerufen wird, wird das Programm bearbeitet.



Fehler-OBs

Es wird unterschieden zwischen synchronen und asynchronen Fehlern:

- Synchroner Fehler können einem MC7-Befehl zugeordnet werden (z. B. Ladebefehl auf gezogene Signalbaugruppe).
- Asynchrone Fehler können einer Prioritätsklasse oder dem gesamten Automatisierungssystem zugeordnet werden (z. B. Zykluszeitüberlauf).

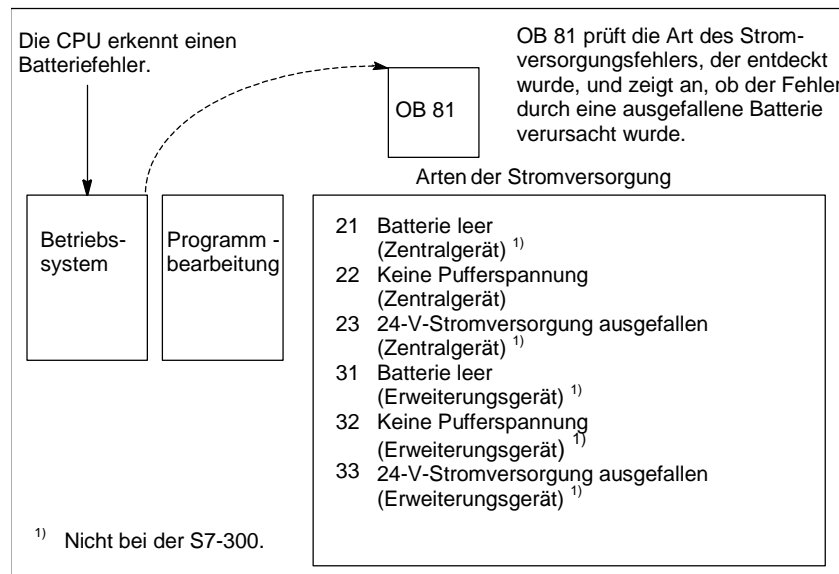
Nachfolgende Tabelle zeigt welche Fehlerarten prinzipiell auftreten können. OB Ihre CPU die angegebenen OBs anbietet, entnehmen Sie bitte dem Handbuch *Automatisierungssystem S7-300, Aufbauen, CPU-Daten*.

Fehlertyp	Fehlerart	OB	Priorität
Asynchron	Zeitfehler	OB 80	26
	Stromversorgungsfehler	OB 81	(bzw. 28, wenn der Fehler-OB im Anlaufprogramm vorkommt)
	Diagnosealarm	OB 82	
	CPU-Hardwarefehler	OB 84	
	Programmablauffehler	OB 85	
	Baugruppenträgerausfall	OB 86	
	Kommunikationsfehler	OB 87	
Synchron	Programmierfehler	OB 121	Priorität des fehlerverursachenden OBs
	Zugriffsfehler	OB 122	

Beispiel zur Verwendung des Fehler-OB 81

Sie können mit den Lokaldaten (Startinformation) des Fehler-OB die Art des aufgetretenen Fehlers auswerten.

Wenn die CPU beispielsweise einen Batteriefehler erkennt, dann ruft das Betriebssystem OB 81 auf (siehe Bild).



Sie können ein Programm schreiben, das den Ereigniscode, der den Aufruf von OB 81 ausgelöst hat, auswertet. Sie können auch ein Programm schreiben, das eine Reaktion hervorruft, wie beispielsweise das Einschalten eines Ausgangs, der an eine Lampe der Operator-Station angeschlossen ist.

Lokaldaten des Fehler-OB 81

Nachfolgende Tabelle beschreibt die temporären (TEMP) Variablen, die in der Variablendeklarationstabelle des OB 81 deklariert sind.

In der Symboltabelle muss auch das Symbol "Batteriefehler" (BOOL) als Ausgang gekennzeichnet werden (z. B. A 4.0), so dass andere Teile des Programms auf diese Daten zugreifen können.

Dekl.	Name	Typ	Beschreibung
TEMP	OB81_EV_CLASS	BYTE	Fehlerklasse/Fehlerkennung 39xx
TEMP	OB81_FLT_ID	BYTE	Fehlercode: b#16#22 = Pufferspannung im Zentralgerät fehlt
TEMP	OB81_PRIORITY	BYTE	Prioritätsklasse = 26/28
TEMP	OB81_OB_NUMBR	BYTE	81 = OB 81
TEMP	OB81_RESERVED_1	BYTE	Reserviert
TEMP	OB81_RESERVED_2	BYTE	Reserviert
TEMP	OB81_MDL_ADDR	INT	Reserviert
TEMP	OB81_RESERVED_3	BYTE	Nur relevant für die Fehlercodes B#16#31, B#16#32, B#16#33
TEMP	OB81_RESERVED_4	BYTE	
TEMP	OB81_RESERVED_5	BYTE	
TEMP	OB81_RESERVED_6	BYTE	
TEMP	OB81_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB gestartet wurde

Beispielprogramm für den Fehler-OB 81

Anhand eines AWL-Beispielprogramms wird gezeigt, wie Sie den Fehlercode in OB 81 lesen können.

Das Programm ist folgendermaßen aufgebaut:

- Der Fehlercode in OB 81 (OB81_FLT_ID) wird gelesen und mit dem Wert des Ereignisses "Batterie leer" (B#16#3921) verglichen.
- Entspricht der Fehlercode dem Code für "Batterie leer", dann springt das Programm zur Marke BFeh und schaltet den Ausgang "Batteriefehler" ein.
- Entspricht der Fehlercode nicht dem Code für "Batterie leer", dann vergleicht das Programm den Code mit dem Code für "Batterieausfall".
- Entspricht der Fehlercode dem Code für "Batterieausfall", dann springt das Programm zur Marke BFeh und schaltet den Ausgang "Batteriefehler" ein. Andernfalls wird der Baustein beendet.

AWL	Beschreibung
L	B#16#21 // Vergleiche Ereigniscode "Batterie leer" // (B#16#21) mit
L	#OB81_FLT_ID // dem Fehlercode für OB 81.
==I	// Wenn gleich (Batterie ist leer), // dann springe zu BF.
SPB BF	
L	B#16#22 // Vergleiche Ereigniscode "Batterieausfall" // (b#16#22) mit
==I	// dem Fehlercode für OB 81.
SPB BF	// Wenn gleich, dann springe zu BF.
BEA	// keine Meldung über Batteriefehler
BF: L	B#16#39 // Vergleiche Kennung für kommendes Ereignis
L	#OB81_EV_CLASS // mit dem Fehlercode für OB 81.
==I	// wenn Batterieausfall oder leere Batterie // entdeckt wird,
S	Batteriefehler // setze Batteriefehler // (Variable aus Symboltabelle)
L	B#16#38 // Vergleiche Kennung für gehendes Ereignis mit
==I	// dem Fehlercode für OB 81.
R	Batteriefehler // rücksetze Batteriefehler, wenn behoben.

Ausführliche Informationen zu OBs, SFBs und SFCs und Erläuterungen zu den Ereignis-IDs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.3 Einfügen von Ersatzwerten bei Fehlererkennung

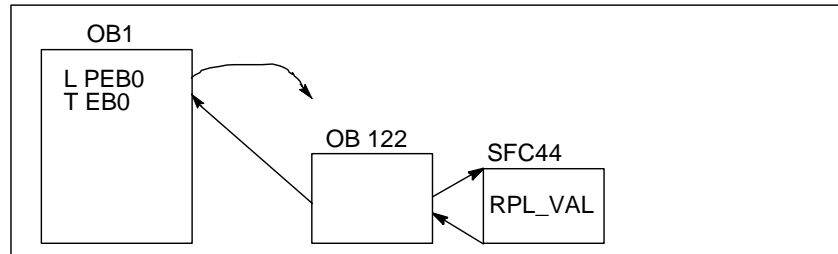
Für bestimmte Fehlerarten (z. B. Drahtbruch bei Eingangssignal) können Sie Ersatzwerte vorgeben für Werte, die aufgrund des Fehlers nicht verfügbar sind. Es gibt zwei Möglichkeiten Ersatzwerte vorzugeben:

- Sie können mit STEP 7 Lite Ersatzwerte für parametrierbare Ausgabebaugruppen parametrieren. Nicht parametrierbare Ausgabebaugruppen haben den voreingestellten Ersatzwert 0.
- Sie können mit Hilfe der SFC 44 RPL_VAL in Fehler-OBs Ersatzwerte programmieren (nur für Eingabebaugruppen).

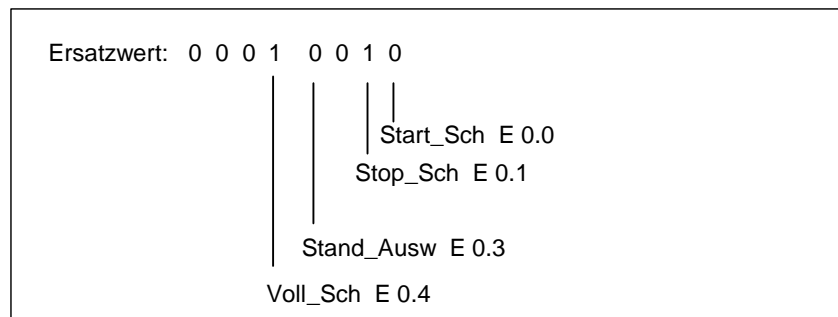
Bei allen Ladebefehlen, die zu synchronen Fehlern führen, können Sie im Fehler-OB einen Ersatzwert für den AKKU-Inhalt angeben.

Beispielprogramm zum Ersetzen eines Wertes

Im folgenden Beispielprogramm wird ein Ersatzwert in der SFC 44 RPL_VAL bereitgestellt. Nachfolgendes Bild zeigt, wie OB 122 aufgerufen werden kann, wenn die CPU erkennt, dass eine Eingabebaugruppe nicht reagiert.



In diesem Beispiel wird der Ersatzwert aus nachfolgendem Bild in das Programm eingegeben, damit das Programm mit sinnvollen Werten weiterlaufen kann.



Fällt eine Eingabebaugruppe aus, erzeugt die Bearbeitung der Anweisung L PEB0 einen synchronen Fehler und startet den OB 122. Standardmäßig liest der Ladebefehl den Wert 0 ein. Sie können jedoch mit der SFC 44 beliebige zum Prozess passende Ersatzwerte definieren. Die SFC ersetzt den AKKU-Inhalt durch den vorgegebenen Ersatzwert.

Das folgende Beispielprogramm könnte in OB 122 gespeichert sein. Die folgende Tabelle zeigt die temporären Variablen, die in diesem Fall in der Variablendeklarationstabelle des OB 122 deklariert werden müssen.

Dekl.	Name	Typ	Beschreibung
TEMP	OB122_EV_CLASS	BYTE	Fehlerklasse/Fehlerkennung 29xx
TEMP	OB122_SW_FLT	BYTE	Fehlercode: 16#42, 16#43
TEMP	OB122_PRIORITY	BYTE	Prioritätsklasse = Priorität des OB, in dem der Fehler aufgetreten ist
TEMP	OB122_OB_NUMBR	BYTE	122 = OB 122
TEMP	OB122_BLK_TYPE	BYTE	Bausteintyp, in dem der Fehler aufgetreten ist
TEMP	OB122_MEM_AREA	BYTE	Speicherbereich und Zugriffsart
TEMP	OB122_MEM_ADDR	WORD	Adresse im Speicher, an der der Fehler aufgetreten ist

Dekl.	Name	Typ	Beschreibung
TEMP	OB122_BLK_NUM	WORD	Nummer des Bausteins, in dem der Fehler aufgetreten ist
TEMP	OB122_PRG_ADDR	WORD	Relativadresse des fehlerverursachenden Befehls
TEMP	OB122_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB gestartet wurde
TEMP	Fehler	INT	Speichert den Fehlercode von SFC44

AWL	Beschreibung
<pre> L B#16#2942 L #OB122_SW_FLT ==I SPB QFeh L B#16#2943 <> I SPB Stop Qfeh: CALL "REPL_VAL" VAL := DW#16#2912 RET_VAL := #Fehler L #Fehler L 0 ==I BEB Stop: CALL "STP" </pre>	<p>Vergleiche den Ereigniscode von OB 122 mit dem Ereigniscode (B#16#2942) für die Quittierung eines Zeitfehlers beim Lesen von der Peripherie. Wenn gleich, springe zu "QFeh".</p> <p>Vergleiche den Ereigniscode von OB 122 mit dem Ereigniscode (B#16#2943) für einen Adressierungsfehler (Schreiben von einer Baugruppe, die nicht vorhanden ist). Wenn ungleich, springe zu "Stop".</p> <p>Marke "QFeh": Übergibt DW#16#2912 (binär 10010) an die SFC 44 (REPL_VAL). SFC 44 lädt diesen Wert in AKKU 1 (und ersetzt den Wert, der den Aufruf von OB 122 ausgelöst hat). Speichert den SFC-Fehlercode in #Fehler.</p> <p>Vergleicht #Fehler mit 0 (wenn gleich, dann ist bei der Bearbeitung von OB 122 kein Fehler aufgetreten). Beende den Baustein, wenn kein Fehler aufgetreten ist.</p> <p>Marke "Stop": Ruft SFC 46 "STP" auf und versetzt die CPU in den Betriebszustand STOP.</p>

11.11.4 Zeitfehler (OB 80)

Beschreibung

Das Betriebssystem der CPU ruft OB 80 auf, wenn ein Zeitfehler auftritt. Zeitfehler können z. B. sein:

- Überschreiten der Maximalzykluszeit
- Überspringen von Uhrzeitalarmen durch Vorstellen der Uhrzeit
- zu große Verzögerung beim Bearbeiten einer Prioritätsklasse

Programmieren des OB 80

Den OB 80 müssen Sie mit STEP 7 Lite als Objekt in Ihrem Programm erzeugen. Schreiben Sie das Programm, das im OB 80 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 80 beispielsweise dazu nutzen:

- die Startinformation des OB 80 auszuwerten und festzustellen, welche Uhrzeitalarme übersprungen wurden
- mit Hilfe der SFC 29 CAN_TINT den übersprungenen Uhrzeitalarm zu deaktivieren, damit er nicht ausgeführt wird und um mit der neu eingestellten Uhrzeit einen sauberen Aufsetzpunkt für die Bearbeitung der Uhrzeitalarme zu haben

Wenn Sie übersprungene Uhrzeitalarme im OB 80 nicht deaktivieren, wird der erste übersprungene Uhrzeitalarm bearbeitet, alle anderen werden ignoriert.

Haben Sie OB 80 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Zeitfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.5 Stromversorgungsfehler (OB 81)

Beschreibung

Das Betriebssystem der CPU ruft OB 81 auf, wenn im Zentralgerät oder einem Erweiterungsgerät

- die 24V-Spannungsversorgung
- eine Batterie
- die komplette Pufferung

ausgefallen ist bzw. wenn die Störung beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 81

Den OB 81 können Sie mit STEP 7 Lite als Baustein in Ihr Programm einfügen. Schreiben Sie das Programm, das im OB 81 bearbeitet werden soll, in den erzeugten Baustein.

Sie können den OB 81 beispielsweise dazu nutzen

- die Startinformation des OB 81 auszuwerten und festzustellen, welcher Stromversorgungsfehler vorliegt
- die Nummer des Baugruppenträgers mit der defekten Stromversorgung zu ermitteln
- eine Lampe an einer Operator Station anzusteuern, um dem Wartungspersonal anzuzeigen, daß eine Batterie auszutauschen ist.

Haben Sie OB 81 nicht programmiert, wechselt die CPU im Gegensatz zu allen anderen asynchronen Fehler-OBs nicht in den Betriebszustand "STOP", wenn ein Stromversorgungsfehler erkannt wird. Der Fehler wird aber in den Diagnosepuffer eingetragen und die entsprechende LED auf der Frontplatte zeigt den Fehler an.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.6 Diagnosealarm (OB 82)

Beschreibung

Das Betriebssystem der CPU ruft OB 82 auf, wenn eine diagnosefähige Baugruppe, bei der Sie den Diagnosealarm freigegeben haben, einen Fehler erkennt und wenn der Fehler beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 82

Den OB 82 müssen Sie mit STEP 7 Lite als Objekt in Ihrem Programm erzeugen. Schreiben Sie das Programm, das im OB 82 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 82 beispielsweise dazu nutzen

- die Startinformation des OB 82 auszuwerten
- eine genaue Diagnose des aufgetretenen Fehlers vorzunehmen.

Wenn ein Diagnosealarm ausgelöst wird, trägt die gestörte Baugruppe automatisch 4 Byte Diagnosedaten und ihre Anfangsadresse in die Startinformation des Diagnosealarm-OBs und in den Diagnosepuffer ein. Damit erhalten Sie die Information, auf welcher Baugruppe und wann ein Fehler aufgetreten ist.

Weitere Diagnosedaten der gestörten Baugruppe (auf welchem Kanal ist der Fehler aufgetreten, um welchen Fehler handelt es sich) können Sie mit einem entsprechenden Programm im OB 82 auswerten. Mit der SFC 51 RDSYSST können Sie Baugruppendiagnosedaten auslesen und mit der SFC 52 WR_USRMSG diese Information in den Diagnosepuffer eintragen. Außerdem können Sie zusätzlich die selbstdefinierte Diagnosemeldung an ein angemeldetes Beobachtungsgerät senden.

Haben Sie OB 82 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Diagnosealarm ausgelöst wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.7 CPU-Hardwarefehler (OB 84)

Beschreibung

Das Betriebssystem der CPU ruft OB 84 auf, wenn ein Fehler bei der Schnittstelle zum MPI-Netz, zum K-Bus oder zur Anschaltung für die dezentrale Peripherie, z. B. fehlerhafter Signalpegel auf der Leitung erkannt wird bzw. wenn der Fehler beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 84

Den OB 84 müssen Sie mit STEP 7 Lite als Objekt in Ihrem Programm erzeugen. Schreiben Sie das Programm, das im OB 84 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 84 beispielsweise dazu nutzen

- die Startinformation des OB 84 auszuwerten
- mit Hilfe der Systemfunktion SFC 52 WR_USMSG eine Meldung zum Diagnosepuffer zu senden.

Haben Sie OB 84 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein CPU-Hardwarefehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.8 Programmablauffehler (OB 85)

Beschreibung

Das Betriebssystem der CPU ruft OB 85 auf, wenn

- ein Startereignis für einen Alarm-OB vorliegt, aber der OB nicht ausgeführt werden kann, weil er nicht in die CPU geladen wurde,
- ein Fehler beim Zugriff auf den Instanz-Datenbaustein eines Systemfunktionsbausteins aufgetreten ist,
- ein Fehler bei der Prozessabbild-Aktualisierung (Baugruppe nicht vorhanden oder defekt) aufgetreten ist.

Programmieren des OB 85

Den OB 85 müssen Sie mit STEP 7 Lite als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 85 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 85 beispielsweise dazu nutzen

- die Startinformation des OB 85 auszuwerten und festzustellen, welche Baugruppe defekt ist oder fehlt (Angabe der Baugruppen-Anfangsadresse)
- mit Hilfe der SFC 49 LGC_GADR den Steckplatz der betreffenden Baugruppe zu ermitteln

Haben Sie OB 85 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Prioritätsklassenfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.9 Baugruppenträgerausfall (OB 86)

Beschreibung

Das Betriebssystem der CPU ruft OB 86 auf, wenn ein Baugruppenträgerausfall erkannt wird, z. B. bei

- Baugruppenträgerausfall (fehlende oder defekte IM bzw. unterbrochene Verbindungsleitung,
- dezentraler Spannungsausfall eines Baugruppenträgers,
- Ausfall eines DP-Slaves in einem Mastersystem des Bussystems PROFIBUS-DP,

bzw. wenn der Fehler beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 86

Den OB 86 müssen Sie mit STEP 7 Lite als Objekt in Ihrem Programm erzeugen. Schreiben Sie das Programm, das im OB 86 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 86 beispielsweise dazu nutzen

- die Startinformation des OB 86 auszuwerten und festzustellen, welcher Baugruppenträger defekt ist oder fehlt.
- mit Hilfe der Systemfunktion SFC 52 WR_USMSG eine Meldung in den Diagnosepuffer einzutragen und die Meldung zu einem Beobachtungsgerät zu senden.

Haben Sie OB 86 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Baugruppenträgerausfall erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.10 Kommunikationsfehler (OB 87)

Beschreibung

Das Betriebssystem der CPU ruft OB 87 auf, wenn ein Kommunikationsfehler beim Datenaustausch über Kommunikations-Funktionsbausteine oder Globaldatenkommunikation auftritt, z. B.

- beim Empfang von Globaldaten wurde eine falsche Telegrammkennung erkannt,
- der Datenbaustein für die Statusinformation der Globaldaten ist nicht vorhanden oder zu kurz.

Programmieren des OB 87

Den OB 87 müssen Sie mit STEP 7 Lite als Objekt in Ihrem Programm erzeugen. Schreiben Sie das Programm, das im OB 87 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 87 beispielsweise dazu nutzen

- die Startinformation des OB 87 auszuwerten und
- wenn der Datenbaustein für die Statusinformation der Globaldatenkommunikation fehlt, einen Datenbaustein anzulegen.

Haben Sie OB 87 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Kommunikationsfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.11 Programmierfehler (OB 121)

Beschreibung

Das Betriebssystem der CPU ruft OB 121 auf, wenn ein Programmierfehler auftritt, z. B.

- adressierte Zeiten nicht vorhanden,
- aufgerufener Baustein nicht geladen.

Programmieren des OB 121

Den OB 121 müssen Sie mit STEP 7 Lite als Objekt in Ihrem Programm erzeugen. Schreiben Sie das Programm, das im OB 121 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 121 beispielsweise dazu nutzen

- die Startinformation des OB 121 auszuwerten
- die Fehlerursache in einen Melde-Datenbaustein einzutragen

Haben Sie OB 121 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Programmierfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

11.11.12 Peripheriezugriffsfehler (OB 122)

Beschreibung

Das Betriebssystem der CPU ruft OB 122 auf, wenn mit einer STEP 7 Lite-Operation auf einen Ein- oder Ausgang einer Signalbaugruppe zugegriffen wird, dem zum Zeitpunkt des letzten Neustarts keine Baugruppe zugeordnet war, z. B.

- Fehler bei Peripherie-Direktzugriff (Baugruppe defekt oder nicht vorhanden),
- Zugriff auf eine Peripherie-Adresse, die der CPU nicht bekannt ist.

Programmieren des OB 122

Den OB 122 müssen Sie mit STEP 7 Lite als Objekt in Ihrem Programm erzeugen. Schreiben Sie das Programm, das im OB 122 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 122 beispielsweise dazu nutzen

- die Startinformation des OB 122 auszuwerten
- die Systemfunktion SFC 44 aufzurufen und einen Ersatzwert für eine Eingabebaugruppe vorzugeben, damit das Programm mit einem sinnvollen, prozessabhängigen Wert weiterbearbeitet werden kann

Haben Sie OB 122 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Peripheriezugriffsfehler erkannt wird.

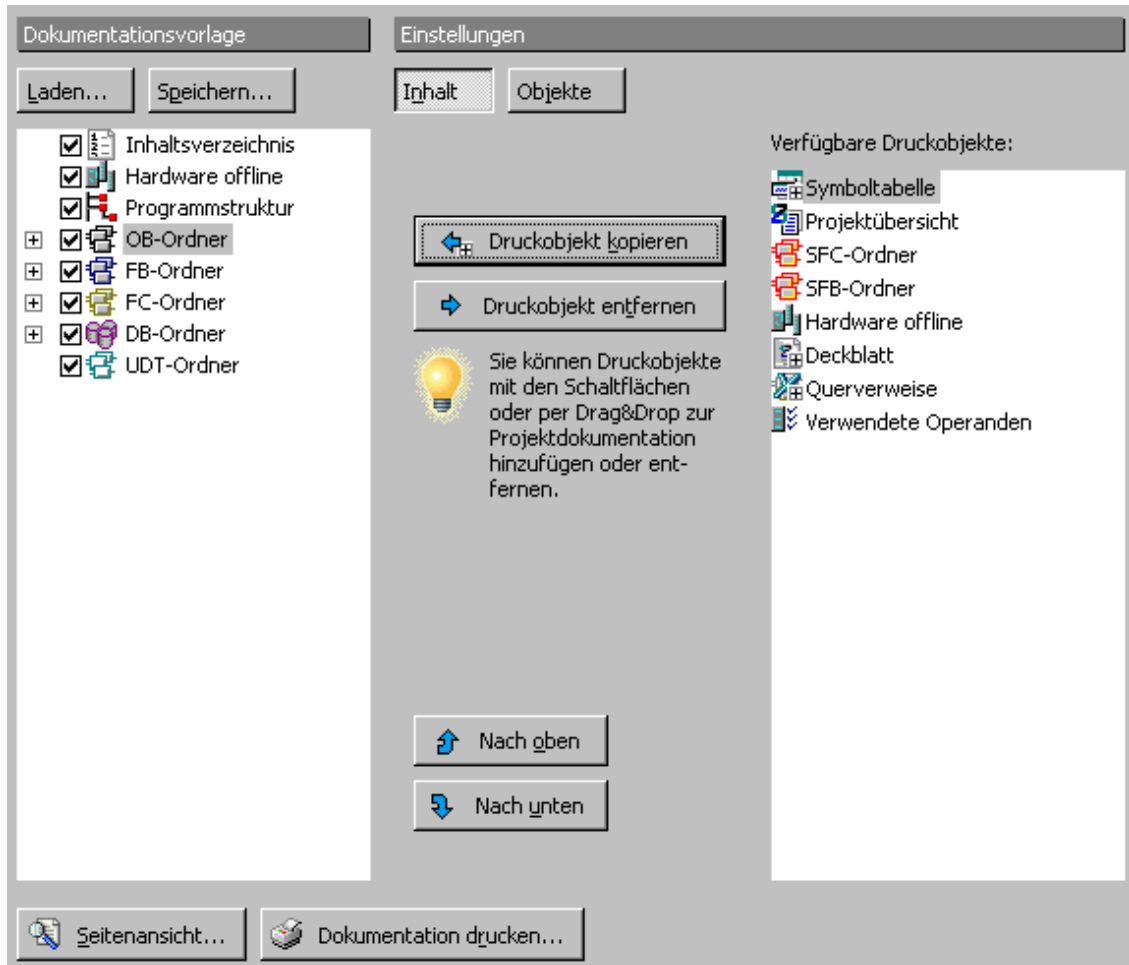
Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

12 Drucken einer Projektdokumentation

12.1 Übersicht zur Projektdokumentation

Mit STEP 7 Lite können Sie Ihre Projektdokumentation frei und individuell gestalten. Die Sicht zur Gestaltung der Projektdokumentation öffnen Sie mit Doppelklicken auf das Symbol "Projektdokumentation".

- Aus einer Druckobjektliste wählen Sie die von Ihnen gewünschten Druckobjekte, wie Deckblatt, Querverweisliste, Symboltabelle, Bausteinordner usw. für Ihre Projektdokumentation aus und passen ihre Reihenfolge in der Dokumentation nach Ihren Wünschen an.
- Die Art ihrer Darstellung (Anzeige mit Symbolen oder ohne usw.) lässt sich bei einzelnen Druckobjekten in den "Optionen" anpassen.
- Für jedes Druckobjekt lässt sich die Schriftart, Schriftgröße und Schriftstil individuell einstellen. Ist eine individuelle Unterscheidung der Druckobjekte nicht gewünscht, so kann die **Schriftart Formatvorlage** "Standard" oder eine zuvor abgespeicherte Vorlage verwendet werden.
- Das Seitenformat "Hochformat" oder "Querformat", als auch die Gestaltung der Kopf- und Fußzeilen lassen sich im Seitenlayout für jedes Druckobjekt individuell einstellen. Wünschen Sie eine einheitliche Gestaltung so können Sie dem Seitenlayout auch die **Seitenlayout Formatvorlage** "Standard" oder eine zuvor abgespeicherte Vorlage zuweisen.
- Um das zu erwartende Druckergebnis vorab zu überprüfen, können Sie jedes Druckobjekt in einer "Seitenansicht" kontrollieren.
- Haben Sie die gesamte Projektdokumentation nach Ihren Wünschen eingestellt, so können Sie alle Einstellungen für eine spätere Verwendung als **Dokumentationsvorlage** abspeichern. Sie können sich so für unterschiedliche Anwendungsfälle, z. B. für die "Übergabedokumentation", für die Dokumentation für Inbetriebsetzung oder für Service und Instandhaltung, eigene Dokumentationsvorlagen anlegen.
- Mit der Schaltfläche "Dokumentation drucken" können Sie vor dem Ausdruck der Projektdokumentation noch den Drucker und individuelle Einstellungen für den Drucker auswählen.



12.2 Projektdokumentation zusammenstellen

In STEP 7 Lite können Sie Ihre Projektdokumentation frei gestalten und die Elemente, die im Ausdruck erscheinen sollen, mittels Druckobjekten auswählen.

Druckobjekte auswählen

Um den Inhalt der Projektdokumentation zusammenzustellen, gehen Sie folgendermaßen vor:

1. Rufen Sie die Sicht zur Gestaltung der Projektdokumentation mit dem Menübefehl **Ansicht > Projektdokumentation** auf. Alternativ können Sie sich die Sicht mit Doppelklicken auf "Projektdokumentation" (im Projektfenster) anzeigen lassen.
Sie sehen links die Liste der Druckobjekte, die in Ihrem späteren Ausdruck erscheinen sollen. Rechts werden die zur Verfügung stehenden Druckobjekte aufgelistet (Schaltfläche "Inhalt" unter der Überschrift "Einstellungen" ist gedrückt).
2. Ziehen Sie nun die von Ihnen gewünschten Druckobjekte per Drag&Drop von rechts nach links, an die Stelle der Liste, in der sie später gedruckt werden sollen. Alternativ können Sie das Druckobjekt rechts selektieren und mit der Schaltfläche "Druckobjekt verschieben" in die Liste Ihrer Druckobjekte aufnehmen. Einige der Druckobjekte (wie Deckblatt) können Sie mehrmals in ihrer Projektdokumentation einfügen.

Werden einzelne Druckobjekte in Ihrer Projektdokumentation nicht benötigt, so können Sie diese temporär in der Druckliste durch Deaktivieren des Optionskästchens unterdrücken. Wollen Sie das Druckobjekt aus der Druckliste entfernen, so ziehen Sie es per Drag&Drop auf die rechte Seite zurück. Alternativ können Sie das Druckobjekt links selektieren und mit der Schaltfläche "Druckobjekt entfernen" aus der Druckliste entfernen.

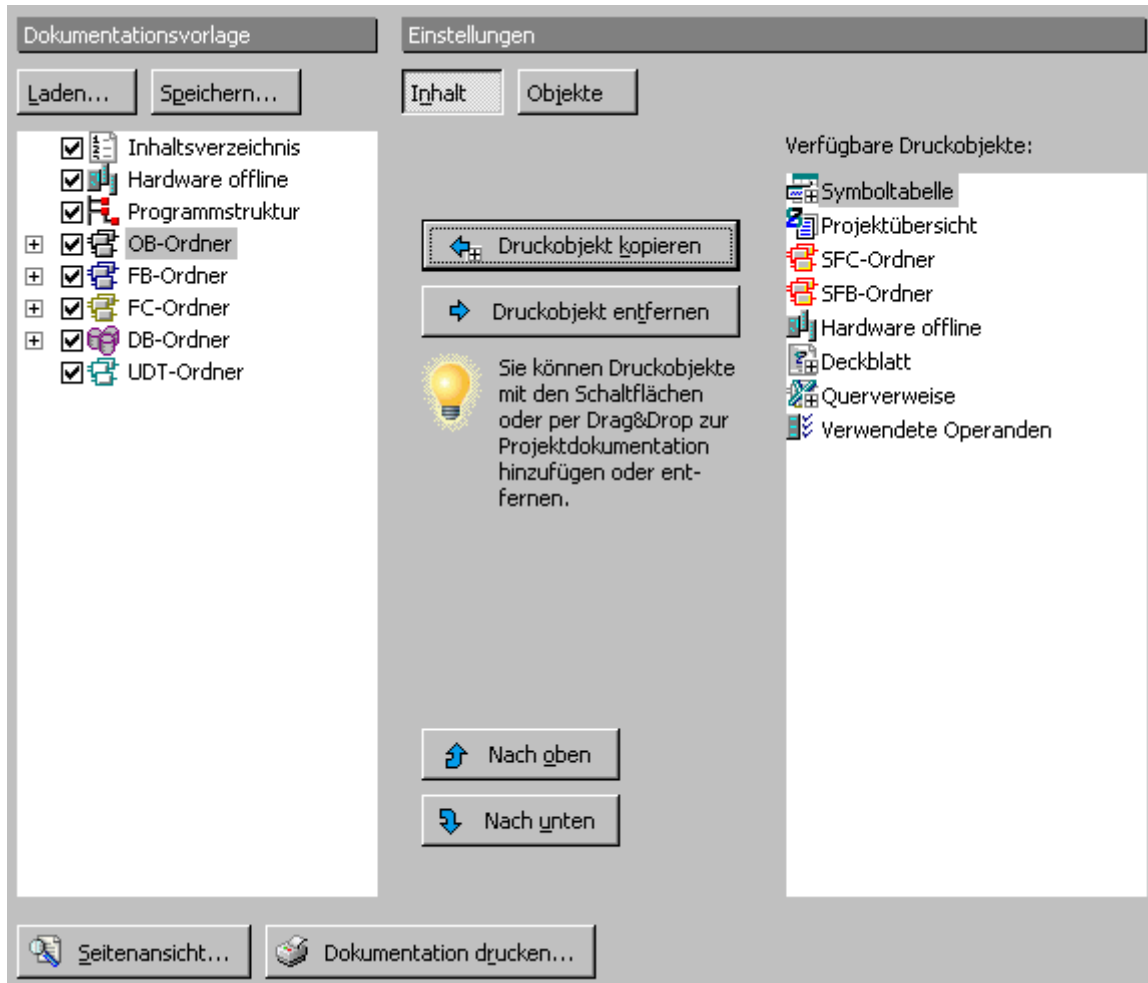
Druckobjekte umsortieren

Die Reihenfolge der Druckobjekte in der Druckliste (und damit auch im Ausdruck) können Sie frei wählen.

1. Selektieren Sie das Druckobjekt, das Sie an einer anderen Stelle der Projektdokumentation ausdrucken wollen.
2. Betätigen Sie die Schaltfläche "Nach oben" oder "Nach unten", wenn Sie das Druckobjekt an einer anderen Stellen ausdrucken wollen.









Das Inhaltsverzeichnis bildet eine Ausnahme. Das Inhaltsverzeichnis kann nur am Anfang oder am Ende der Projektdokumentation positioniert werden. Vor dem Inhaltsverzeichnis können Sie jedoch beliebig viele Deckblätter als Druckobjekte einfügen.














Bei mehreren Deckblättern bestimmt die Position des "Inhaltsverzeichnisses" in der Druckliste, ab welchem Druckobjekt die Objekte im Inhaltsverzeichnis aufgeführt und nummeriert werden. Befindet sich das "Inhaltsverzeichnis" z.B. zwischen dem 2. und dem 3. Deckblatt, so wird das 3. Deckblatt im Inhaltsverzeichnis aufgeführt und mit in die Nummerierung eingeschlossen. Deckblatt 1 und 2 erscheinen nicht im Inhaltsverzeichnis.



12.3 Druckobjekte

In der folgenden Tabelle sind die druckbaren Objekte der Druckliste aufgelistet. Außerdem finden Sie Informationen, OB eine Druckvorschau verfügbar ist und OB das Druckobjekt mehrfach angelegt werden kann.

Druckobjekt	Symbol	Druckvorschau verfügbar?	Mehrfach druckbar?	Bemerkung
Deckblatt		Ja	Ja	Texteingabe und Schriftartauswahl möglich Das Symbol mit dem Pluszeichen befindet sich in der Liste der zur Verfügung stehenden Druckobjekte. Das Pluszeichen deutet an, dass das Objekt mehrfach verwendet werden kann.
Inhaltsverzeichnis		Ja	Nein	Das Inhaltsverzeichnis wird am Anfang oder am Ende der Projektdokumentation ausgedruckt. Bei mehreren Deckblättern bestimmt die Position des "Inhaltsverzeichnisses" in der Druckliste, ab welchem Druckobjekt die Objekte im Inhaltsverzeichnis aufgeführt und nummeriert werden. Befindet sich das "Inhaltsverzeichnis" z.B. zwischen dem 2. und dem 3. Deckblatt, so wird das 3. Deckblatt im Inhaltsverzeichnis aufgeführt und mit in die Nummerierung eingeschlossen. Deckblatt 1 und 2 erscheinen nicht im Inhaltsverzeichnis.
Symboltabelle		Ja	Ja	Filter und Sortierung einstellbar
Querverweise		Ja	Ja	Filter und Sortierung einstellbar Das Symbol mit dem Pluszeichen befindet sich in der Liste der zur Verfügung stehenden Druckobjekte. Das Pluszeichen deutet an, dass das Objekt mehrfach verwendet werden kann.
Verwendete Operanden		Ja	Nein	-
Programmstruktur		Ja	Nein	Druckt die Programmstruktur grafisch als Baumstruktur
Hardware offline		Ja	Nein	Druckt die Hardware-Konfiguration
Projektübersicht		Ja	Nein	Kategorien werden mitgedruckt

Druckobjekt	Symbol	Druckvorschau verfügbar?	Mehrfach druckbar?	Bemerkung
Baustein-Ordner		Ja	Nein	Ziehen Sie die Druckobjekte DB-Ordner, OB-Ordner, FB-Ordner, FC-Ordner, UDT-Ordner, SFB- oder SFC-Ordner aus der Liste der zur Verfügung stehenden Druckobjekte per Drag&Drop oder mit der Schaltfläche "Druckobjekt verschieben" in Ihre Liste der Druckobjekte. Ein Pluszeichen in der Liste der Druckobjekte symbolisiert, dass im Druckobjekt-Ordner noch weitere Druckobjekte zu finden sind. Klicken Sie auf das Pluszeichen um sich alle Bausteine des Ordners als Druckobjekte anzuzeigen zu lassen. Die Optionseinstellungen für den Bausteinordner gelten defaultmäßig auch für die darunter liegenden Bausteine. Möchten Sie für Einzelbausteine abweichende Optionseinstellungen vornehmen, so aktivieren Sie das Optionskästchen "individuelle Einstellungen" und nehmen die gewünschten Einstellungen vor.
DB-Ordner		Ja	Nein	-
Datenbausteine (DBs)		Ja	Nein	-
OB-Ordner		Ja	Nein	-
Organisationsbausteine (OBs)		Ja	Nein	-
FB-Ordner		Ja	Nein	-
Funktionsbausteine (FBs)		Ja	Nein	-
FC-Ordner		Ja	Nein	-
Funktionen (FCs)		Ja	Nein	-
UDT-Ordner		Ja	Nein	-
Datentypen (UDTs)		Ja	Nein	-
SFB-Ordner		Ja	Nein	-
Systemfunktionsbausteine (SFBs)		Ja	Nein	-
SFC-Ordner		Ja	Nein	-
Systemfunktionen (SFCs)		Ja	Nein	-

12.4 Optionen, Schriftart und Seitenlayout festlegen

Die Darstellung und das Aussehen der einzelnen Druckobjekte wird in den Registern "Optionen", "Schriftart" und "Seitenlayout" festgelegt.

Um die "Optionen", die "Schriftart" oder das "Seitenlayout" für einzelne Druckobjekte einzustellen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf die Schaltfläche "Objekte" (unter der Überschrift "Einstellungen").
2. Selektieren Sie das Druckobjekt in der linken Druckliste und wählen das gewünschte Register.

Optionen

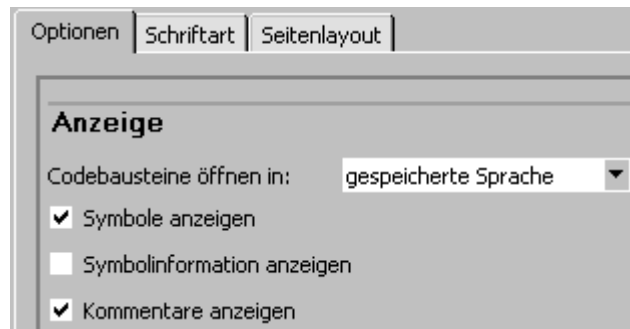
Die Optionen lassen sich festlegen für

- Bausteinordner und für Bausteine
- Symboltabellen
- Hardware offline
- Querverweise
- Deckblätter

Einstellungen für den Bausteinordner gelten auch für alle dazugehörigen Einzelbausteine, sofern bei den Optionen dieser Bausteine das Optionskästchen "individuelle Einstellungen" deaktiviert ist (Voreinstellung). Wenn das Optionskästchen "individuelle Einstellungen" aktiviert ist, können Sie abweichende Optionen festlegen.

Beispiel: Optionen bei Bausteinordnern und Bausteinen

- Codebaustein öffnen in
Sie können den Baustein bzw. die Bausteine in der Sprache ausdrucken, in der der Baustein erstellt wurde oder als Sprache zum Ausdruck "AWL", "KOP" bzw. "FUP" festlegen.
- Symbole anzeigen
Ist das Optionskästchen aktiviert, so werden die symbolischen Bezeichner der Operanden des Druckobjekts im Ausdruck verwendet.
- Symbolinformation anzeigen
Ist das Optionskästchen aktiviert, so werden die Kommentarinformationen des Druckobjekts zum Symbol ausgedruckt.
- Kommentare anzeigen
Ist das Optionskästchen aktiviert, so werden die Anweisungskommentare des Druckobjekts ausgedruckt.



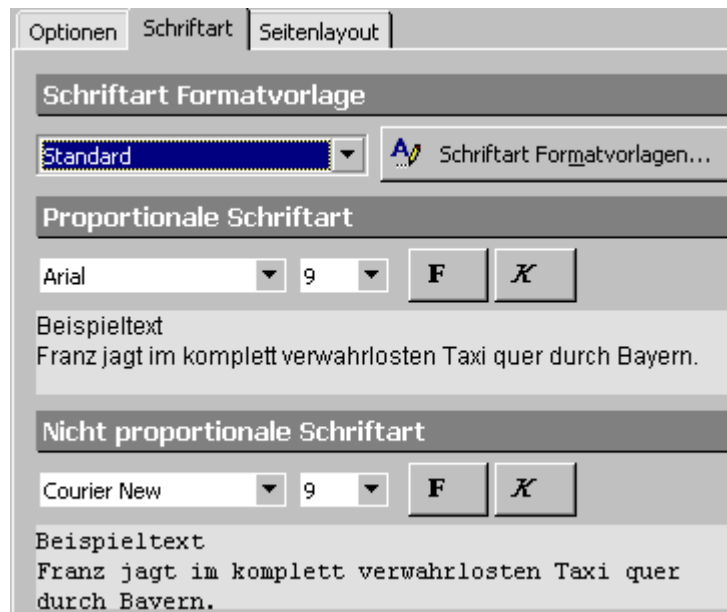
Schriftart

Die Schriftart, die Schriftgröße und der Schriftstil läßt sich für jedes Druckobjekt individuell einstellen

Die Einstellungen können für proportionale und nicht proportionale Schriften getrennt vorgenommen werden. Die Proportionale Schrift kommt in Textpassagen, die bündig dargestellt werden sollen, wie der Codeauflistung, zum Einsatz.

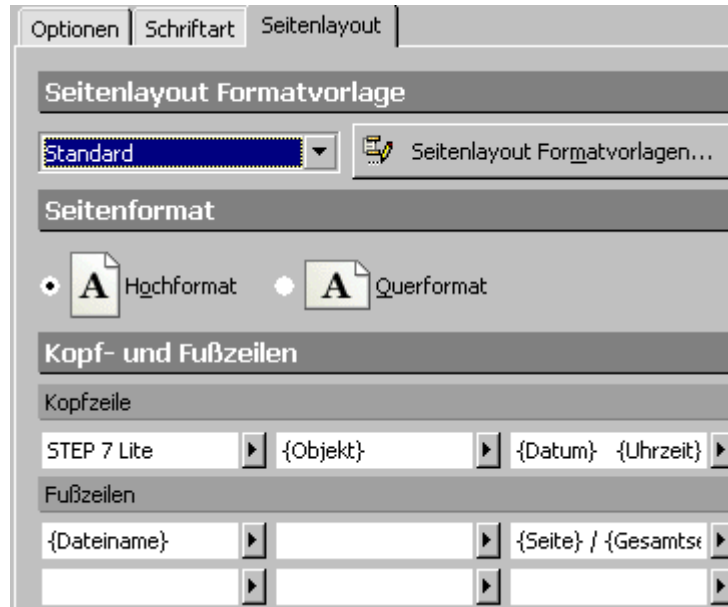
Ein Beispieltext zeigt das zu erwartende Druckergebnis beispielhaft an.

Ist eine individuelle Unterscheidung der Druckobjekte nicht gewünscht, so kann die Schriftart Formatvorlage "Standard" oder eine zuvor abgespeicherte Vorlage verwendet werden.



Seitenlayout

Die Gestaltung der Kopf- und Fußzeilen und deren Inhalt, als auch das Seitenformat "Hochformat" oder "Querformat", lassen sich im Seitenlayout für jedes Druckobjekt individuell einstellen.



Der Kopf- und Fußzeilenbereich ist in einen links bündigen, einen mittigen und einem rechts bündigen Teilbereich aufgeteilt. Die Kopfzeile ist einzeilig, die Fußzeile zweizeilig aufgebaut. Die Inhalte können editiert, bzw. über ein Einfügemenü (Schaltfläche "Rechts") eingefügt werden.



Mit der Objektauswahl aus dem Einfügemenü wird das gewählte Objekt im links befindlichen Feld eingefügt. Während des Ausdrucks wird der Inhalt des Objekts dynamisch aktualisiert.

Je nach gewähltem Druckobjekt stehen folgende dynamischen Objekte zur Verfügung:

- Objekt: Name des Druckobjekt
- Datum: aktuelles Datum zum Druckzeitpunkt
- Uhrzeit: aktuelle Uhrzeit zum Druckzeitpunkt
- Seite: aktuelle Seite (bezogen auf die Gesamtdokumentation)
- Gesamtseiten: Anzahl der Seiten insgesamt
- Projektname / Dateiname: Name des STEP 7 Lite-Projekts
- Autor: Autor des Bausteins
- Version: Version des Bausteins

Neben der Objektauswahl aus dem Einfügemenü kann auch statischer Text in ein Feld eingegeben werden.

Bereits eingefügte Texte und Objekte können nach ihrer Selektion mit der DEL-Taste gelöscht werden.

Wünschen Sie eine einheitliche Gestaltung für die Druckobjekte, so können Sie dem Seitenlayout auch die Seitenlayout Formatvorlage "Standard" oder eine zuvor abgespeicherte Vorlage zuweisen.

12.5 Vorlagen definieren und verwenden

Vorlagen haben den Vorteil, einmal getätigte Einstellungen für die Projektdokumentation zu speichern und bei Bedarf wieder zu verwenden.

Dokumentationsvorlagen

In einer Dokumentationsvorlage werden alle Einstellungen einer Projektdokumentation abgespeichert (ausgenommen sind individuelle Einstellungen zu Einzelbausteinen in Bausteinordnern).

Zu den Einstellungen, die in einer Dokumentationsvorlage abgespeichert werden, gehören die im Projekt erstellten Schriftart Formatvorlagen und die Seitenlayout Formatvorlagen.

Über die Schaltfläche "Speichern" können Sie Ihre individuell definierten Dokumentationseinstellungen als Vorlage speichern und damit für andere Projekte nutzbar machen.

Aus den aktuellen Einstellungen eine Dokumentationsvorlage erstellen

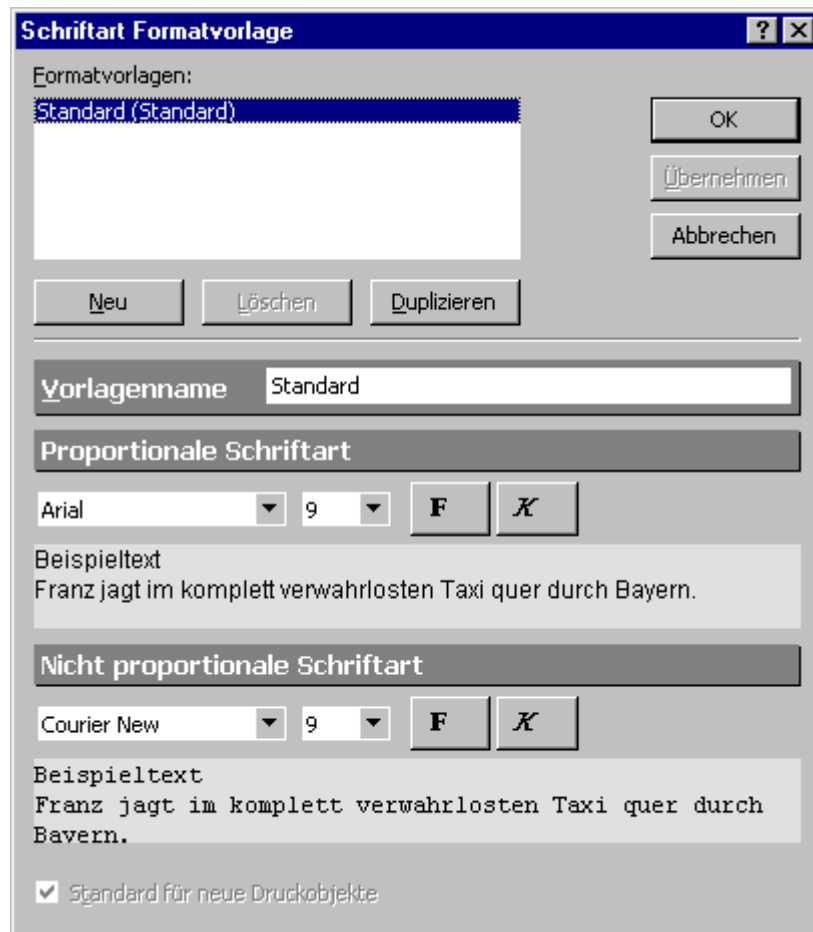
1. Klicken Sie in der Sicht "Projektdokumentation" auf die Schaltfläche "Speichern" (unter der Überschrift "Dokumentationsvorlage").
2. Wählen Sie im Dialogfeld "Dokumentationsvorlage speichern" ein Ziel und einen Namen für die Dokumentationsvorlage.

Dokumentationsvorlage verwenden

1. Klicken Sie in der Sicht "Projektdokumentation" auf die Schaltfläche "Laden" (unter der Überschrift "Dokumentationsvorlage").
2. Navigieren Sie im Dialogfeld "Dokumentationsvorlage laden" zur gewünschten Dokumentationsvorlage und wählen sie aus.

Schriftart Formatvorlage

In den Schriftart Formatvorlagen können Sie immer wiederkehrende Einstellungen für die proportionale und nicht proportionale Schriftart einstellen und speichern.



Neue Schriftart Formatvorlage erstellen

1. Klicken Sie auf die Schaltfläche "Objekte" (unter der Überschrift "Einstellungen").
2. Wählen Sie das Register "Schriftart".
3. Klicken Sie auf die Schaltfläche "Schriftart Formatvorlagen".
4. Klicken Sie auf die Schaltfläche "Neu" und vergeben einen Namen für die neue Formatvorlage.
5. Wählen Sie Schriftart, Schriftgröße und Schriftstil für proportionale und nicht proportionale Schriftart.
6. Falls diese Schriftarten für alle neuen Druckobjekte gelten soll, aktivieren Sie das Optionskästchen "Standard für neue Druckobjekte".
7. Speichern Sie die Schriftart Formatvorlage mit der Schaltfläche "OK" oder "Übernehmen" ("Übernehmen" wählen Sie, wenn das Dialogfeld zur weiteren Bearbeitung geöffnet bleiben soll).

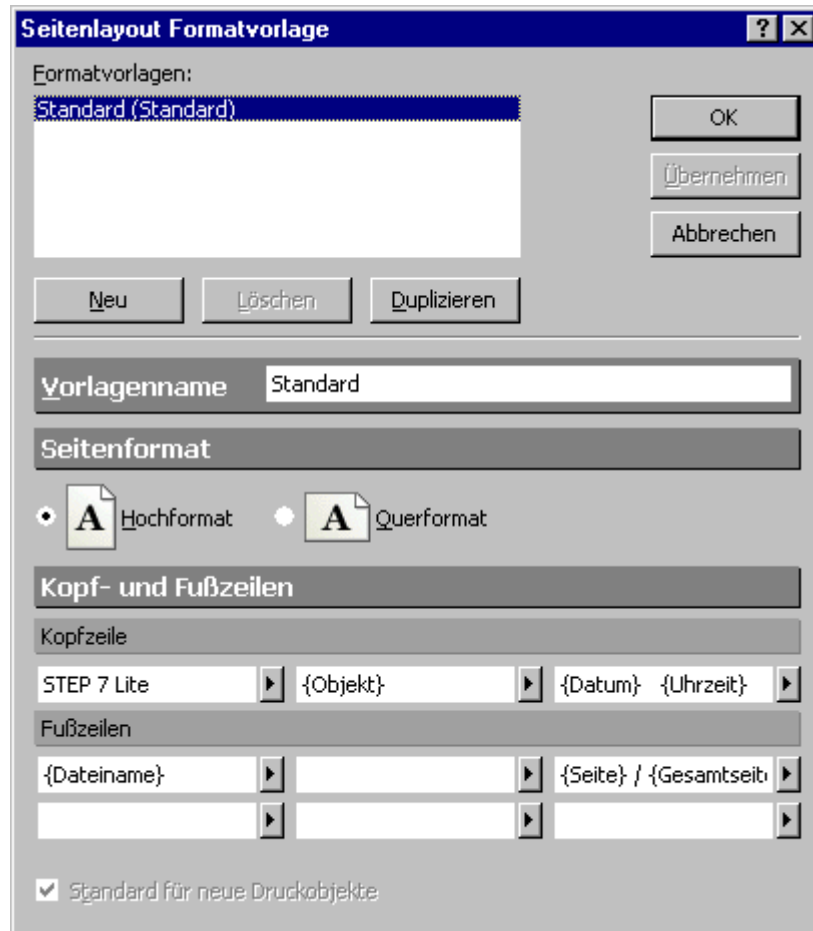
Schriftart Formatvorlagen verwenden

Sie können die nun definierte Schriftart Formatvorlage im Register "Schriftart" für alle Druckobjekte verwenden. Die Schriftarten sind über die Klappliste "Schriftart Formatvorlage" wählbar.

Sobald Sie die Einstellungen einer gewählten Schriftart Formatvorlage ändern, wechselt der Inhalt der Klappliste auf "Individuell".

Seitenlayout Formatvorlage

In den Seitenlayout Formatvorlagen können Sie Einstellungen zum Seitenformat und den Inhalt der Kopf- und Fußzeilen abspeichern.



Neue Seitenlayout Formatvorlage erstellen

1. Klicken Sie auf die Schaltfläche "Objekte" (unter der Überschrift "Einstellungen").
2. Wählen Sie das Register "Seitenlayout".
3. Klicken Sie auf die Schaltfläche "Seitenlayout Formatvorlagen".
4. Klicken Sie auf die Schaltfläche "Neu" und vergeben einen Namen für die neue Formatvorlage.
5. Wählen Sie das Seitenformat und definieren Sie die Kopf- und Fußzeilen.
6. Falls dieses Seitenlayout für alle neuen Druckobjekte gelten soll, aktivieren Sie das Optionskästchen "Standard für neue Druckobjekte".
7. Speichern Sie die Seitenlayout Formatvorlage mit der Schaltfläche "OK" oder "Übernehmen" ("Übernehmen" wählen Sie, wenn das Dialogfeld zur weiteren Bearbeitung geöffnet bleiben soll).

Seitenlayout Formatvorlagen verwenden

Sie können die nun definierte Seitenlayout Formatvorlage im Register "Seitenlayout" für alle Druckobjekte verwenden. Das Seitenlayout ist über die Klappliste "Seitenformat Formatvorlage" wählbar.

Sobald Sie die Einstellungen einer gewählten Seitenlayout Formatvorlage ändern, wechselt der Inhalt der Klappliste auf "Individuell".

12.6 Projektdokumentation drucken

Den Ausdruck der zuvor definierten Projektdokumentation können Sie mit der Schaltfläche "Dokumentation drucken" anstoßen. Im Folgedialog können Sie vor dem Ausdruck noch den Drucker und individuelle Einstellungen für den Drucker auswählen. Die Einstellmöglichkeiten für diese Funktion hängt von dem verwendeten Druckertreiber ab.

Das zu erwartende Druckergebnis kann vorab über die Schaltfläche "Seitenansicht" kontrolliert werden.

Einzelne Objekte drucken

Zum Ausdrucken gehen Sie folgendermaßen vor:

1. Öffnen Sie das geeignete Objekt im Projektfenster, um die zu druckende Information am Bildschirm anzuzeigen.
2. Mit dem Menübefehl **Datei > Drucken** können Sie das einzelne Objekt ausdrucken. Im Folgedialog können Sie vor dem Ausdruck noch den Drucker und individuelle Einstellungen für den Drucker auswählen.
3. Über den Menübefehl **Datei > Seitenansicht** können Sie das zu erwartende Druckergebnis kontrollieren.

Die einzelnen Objekte werden entsprechend ihrer Einstellung in der Projektdokumentation (Optionen, Schriftart und Seitenlayout) ausgedruckt. Falls diese Einstellungen nicht definiert wurden, werden sie entsprechend den Defaulteinstellungen ausgedruckt.

13 Tipps und Tricks

13.1 Baugruppen tauschen in der Hardware-Konfiguration

Wenn Sie eine Stationskonfiguration mit überarbeiten und eine Baugruppen tauschen möchten, z. B. gegen eine mit neuer Bestell-Nr., dann gehen Sie folgendermaßen vor:

1. Ziehen Sie die Baugruppe per Drag&Drop aus dem Katalog über die bereits platzierte ("alte") Baugruppe in der grafischen oder tabellarischen Sicht der HW-Konfiguration.
2. Lassen Sie die neue Baugruppe "fallen"; die neue Baugruppe übernimmt, soweit möglich, die Parameter von der bereits gesteckten.

Diese Vorgehensweise ist schneller als das Austauschen durch Löschen der alten und anschließendem Einfügen und Parametrieren der neuen Baugruppe.

13.2 Testen mit der Variablen-tabelle

Für das Beobachten, Steuern und Forcen von Variablen gibt es eine Reihe von Bearbeitungstipps:

- Symbole und Operanden können Sie sowohl in die Spalte "Symbol" als auch in die Spalte "Operand" eintragen. Der Eintrag wird automatisch in die passende Spalte geschrieben.
- Damit Ihnen der gesteuerte Wert in der Spalte "Statuswert" angezeigt wird, sollten Sie den Triggerpunkt für Beobachten auf "Permanent" einstellen!
- Ist die Schaltfläche "Erweitert" gedrückt:
- Beim Steuern von Ausgängen ist es sinnvoll den Steuermodus auf "Zyklusende permanent" oder "Zyklusende einmalig" einzustellen.
- Beim Steuern von Eingängen ist es sinnvoll den Steuermodus auf "Zyklusbeginn permanent" oder "Zyklusbeginn einmalig" einzustellen.
- Der Steuermodus „Permanent“ kombiniert die beiden obengenannten Eigenschaften.
- Sie können nur Symbole eintragen, die bereits in der Symboltabelle definiert sind. Ein Symbol muss genau so eingegeben werden, wie es in der Symboltabelle definiert ist. Symbolnamen, die Sonderzeichen enthalten, müssen in Hochkommata eingeschlossen werden (z. B. "Motor.Aus", "Motor+Aus", "Motor-Aus").
- Der Beobachtmodus ist während des Beobachtens von Variablen einstellbar.

- Eingeben eines zusammenhängenden Operandenbereichs:
Verwenden Sie den Menübefehl **Einfügen > Operandenbereich**.
- Ändern des Anzeigeformats mehrerer Tabellenzeilen gleichzeitig:
 - Markieren Sie den Bereich der Tabelle, in dem Sie das Anzeigeformat ändern wollen, indem Sie mit gedrückter linker Maustaste über den gewünschten Tabellenbereich ziehen.
 - Wählen Sie die Darstellung mit dem Menübefehl **Ansicht > Anzeigeformat wählen**. Es wird nur für die Zeilen der markierten Tabelle das Format gewechselt, für die der Formatwechsel erlaubt ist.

13.3 Arbeiten ohne Original-Projekt auf dem PG/PC

Wenn Sie ohne Original-Projekt auf dem PG oder PC (Service-Fall) Einstellungen auf der CPU bzw. Baugruppen vornehmen wollen oder Programme anpassen wollen, dann können Sie das folgendermaßen durchführen:

1. Sorgen Sie dafür, dass eine Online-Verbindung zwischen PG/PC und CPU besteht
2. Markieren Sie in der Online-Sicht des Projektfensters (Register "Online-CPU") das Objekt, das Sie bearbeiten wollen; z. B. das Symbol "Online-CPU", wenn Sie umfangreiche Änderungen vornehmen wollen.
3. Wählen Sie den Menübefehl **Datei > Laden in PG**
Wenn noch kein Projekt geöffnet ist, wird automatisch ein neues Projekt angelegt.
Alle markierten Objekte werden in das PG geladen und können anschließend über das Projektfenster (Register "Projekt") geöffnet und bearbeitet werden.
4. Speichern Sie die Änderungen im Projekt (Menübefehl **Datei > Speichern**).
5. Laden Sie die Änderungen in die CPU (Menübefehl **Datei > Laden in CPU**).

Wenn Sie Bausteine und die Konfiguration online öffnen (im Projektfenster Register "Online-CPU"), dann können Sie diese Objekte nicht bearbeiten.

A Anhang

A.1 Betriebszustände

A.1.1 Betriebszustände und Übergänge

Betriebszustände

Betriebszustände beschreiben das Verhalten der CPU zu jedem beliebigen Zeitpunkt. Das Wissen über die Betriebszustände der CPUs ist nützlich für die Programmierung des Anlaufs, den Test der Steuerung sowie für die Fehlerdiagnose.

CPUs können folgende Betriebszustände annehmen:

- STOP
- ANLAUF
- RUN
- HALT

Im Betriebszustand STOP prüft die CPU OB alle konfigurierten bzw. über Default-Adressierung eingesetzten Baugruppen vorhanden sind und setzt die Peripherie in einen vordefinierten Grundzustand. Das Anwenderprogramm wird im Betriebszustand STOP nicht bearbeitet.

Im Betriebszustand "ANLAUF" unterscheidet man zwischen den Anlaufarten "Neustart" (Warmstart) und "Kaltstart":

- Bei Neustart (Warmstart) wird die Programmbearbeitung am Programmanfang mit einer "Grundstellung" der Systemdaten und der Anwenderoperandenbereiche neu begonnen (die nicht remanenten Zeiten, Zähler und Merker werden zurückgesetzt).
- Bei Kaltstart wird das Prozessabbild der Eingänge eingelesen und das STEP 7 Lite- Anwenderprogramm beginnend beim ersten Befehl im OB1 bearbeitet (gilt auch für Neustart (Warmstart)).
 - Per SFC erzeugte Datenbausteine im Arbeitsspeicher werden gelöscht, die übrigen Datenbausteine haben den vorbelegten Wert aus dem Ladespeicher.
 - Das Prozessabbild sowie alle Zeiten, Zähler und Merker werden zurückgesetzt, unabhängig davon, OB sie als remanent parametrieren worden sind oder nicht.

Im Betriebszustand RUN bearbeitet die CPU das Anwenderprogramm, aktualisiert die Ein- und Ausgänge, bearbeitet Alarme und Fehlermeldungen.

Im Betriebszustand HALT wird die Bearbeitung des Anwenderprogramms angehalten, und Sie können das Anwenderprogramm schrittweise testen. Der Betriebszustand HALT kann nicht mit STEP 7 Lite gesetzt werden.

In all diesen Betriebszuständen ist die CPU über die MPI-Schnittstelle kommunikationsfähig.

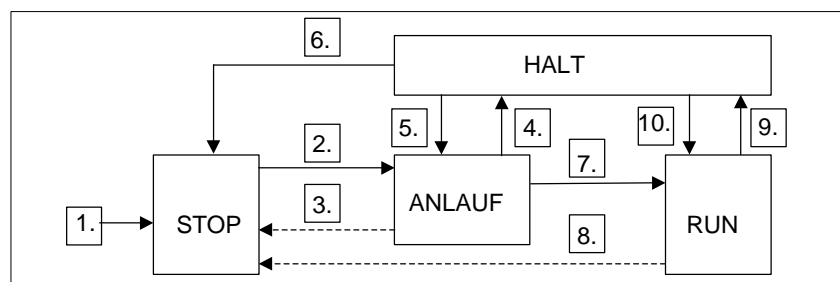
Weitere Betriebszustände

Ist die CPU nicht betriebsbereit, befindet sie sich in einem der beiden Betriebszustände:

- Spannungslos, d. h. die Netzspannung ist ausgeschaltet.
- Defekt, d. h. es ist ein nicht behebbarer Fehler aufgetreten.
Überprüfen Sie, OB die CPU tatsächlich defekt ist: Setzen Sie die CPU auf STOP, und schalten Sie den Netzschalter aus und wieder ein. Wenn die CPU anläuft, lesen Sie den Diagnosepuffer aus, um den Fehler zu analysieren. Wenn die CPU nicht anläuft, muss sie ausgetauscht werden.

Betriebszustandsübergänge

Nachfolgendes Bild zeigt die Betriebszustände und die Betriebszustandsübergänge der S7-300-CPU:



Die Bedingungen, unter welchen die Betriebszustände wechseln, können Sie folgender Tabelle entnehmen.

Übergang	Beschreibung
1.	Nach dem Einschalten der Versorgungsspannung befindet sich die CPU im Betriebszustand STOP.
2.	Die CPU geht in den Betriebszustand "ANLAUF" <ul style="list-style-type: none"> • nachdem die CPU über Schlüsselschalter oder vom PG aus in RUN oder RUN-P gebracht wurde oder • nach automatischer Auslösung einer Anlaufart durch NETZ-EIN. • Wenn die Kommunikationsfunktion "RESUME" oder "START" ausgeführt wird. Der Schlüsselschalter muss in beiden Fällen auf RUN oder RUN-P stehen.
3.	Die CPU geht wieder in den STOP-Zustand, wenn <ul style="list-style-type: none"> • während des Anlaufs ein Fehler erkannt wird • die CPU über Schlüsselschalter oder vom PG aus auf STOP gesetzt wird • ein STOP-Befehl im Anlauf-OB bearbeitet wird oder • die Kommunikationsfunktion STOP ausgeführt wird.
4.	Die CPU geht in den Betriebszustand HALT, wenn im Anlaufprogramm ein Haltepunkt erreicht wird (Haltepunkte sind nicht mit STEP 7 Lite setzbar).
5.	Die CPU geht in den Betriebszustand ANLAUF, wenn der Haltepunkt in einem Anlaufprogramm gesetzt war und der Befehl HALT VERLASSEN ausgeführt wird (Testfunktion).
6.	Die CPU geht wieder in den STOP-Zustand, wenn <ul style="list-style-type: none"> • die CPU über Schlüsselschalter oder vom PG aus auf STOP gesetzt wird oder • der Kommunikationsbefehl STOP ausgeführt wird.
7.	Ist der Anlauf erfolgreich, dann wechselt die CPU in RUN.
8.	Die CPU geht wieder in den STOP-Zustand, wenn <ul style="list-style-type: none"> • im RUN-Zustand ein Fehler erkannt wird und der zugehörige OB nicht geladen ist • die CPU über Schlüsselschalter oder vom PG aus auf STOP gesetzt wird • ein STOP-Befehl im Anwenderprogramm bearbeitet wird oder • die Kommunikationsfunktion STOP ausgeführt wird.
9.	Die CPU geht in den Betriebszustand HALT, wenn im Anwenderprogramm ein Haltepunkt erreicht wird (Haltepunkte sind nicht mit STEP 7 Lite setzbar).
10.	Die CPU geht in den Betriebszustand RUN" wenn ein Haltepunkt gesetzt war und der Befehl HALT VERLASSEN ausgeführt wird.

Priorität der Betriebszustände

Werden mehrere Betriebszustandswechsel gleichzeitig angefordert, wird in den Betriebszustand mit der höchsten Priorität gewechselt. Steht z. B. der Betriebsartenschalter auf RUN, und vom PG aus wird versucht, die CPU in STOP zu schalten, geht die CPU in STOP, weil dieser Betriebszustand die höchste Priorität hat.

Priorität	Betriebszustand
Höchste	STOP
	HALT
	ANLAUF
Niedrigste	RUN

A.1.2 Betriebszustand STOP

Im Betriebszustand "STOP" wird das Anwenderprogramm nicht bearbeitet. Alle Ausgänge werden auf Ersatzwerte gesetzt und damit der gesteuerte Prozess in einen sicheren Betriebszustand gebracht. Die CPU prüft, ob

- Hardwareprobleme vorliegen (z. B. Baugruppen nicht verfügbar sind)
- für die CPU die Defaulteinstellung gelten soll oder Parametersätze vorliegen
- die Randbedingungen für das programmierte Anlaufverhalten stimmen
- Systemsoftwareprobleme vorliegen

Im STOP-Zustand können auch Globaldaten empfangen werden, und es kann passiv einseitige Kommunikation über Kommunikations-SFBs für projektierte Verbindungen und über Kommunikations-SFCs für nichtprojektierte Verbindungen ausgeführt werden.

Urlöschen

Im STOP-Zustand kann die CPU urlöscht werden. Urlöschen kann manuell über den Schlüsselschalter (MRES) oder vom PG aus (z. B. vor dem Laden eines Anwenderprogramms) erfolgen.

Durch Urlöschen wird die CPU in den "Urzustand" versetzt, d. h.

- das gesamte Anwenderprogramm im Arbeitsspeicher und im RAM-Ladespeicher sowie alle Operandenbereiche werden gelöscht.
- die Systemparameter sowie die CPU- und Baugruppenparameter werden auf die Defaulteinstellung zurückgesetzt. Die vor dem Urlöschen eingestellten MPI-Parameter bleiben erhalten.
- wenn eine Memory Card (Flash-EPROM) gesteckt ist, kopiert die CPU das Anwenderprogramm aus der Memory Card in den Arbeitsspeicher (inkl. CPU- und Baugruppenparameter, falls die entsprechenden Konfigurationsdaten sich ebenfalls auf der Memory Card befinden).

Der Diagnosepuffer, die MPI-Parameter, die Uhrzeit und der Betriebsstundenzähler werden nicht zurückgesetzt.

A.1.3 Betriebszustand ANLAUF

Bevor die CPU nach dem Einschalten mit der Bearbeitung des Anwenderprogramms beginnt, wird ein Anlaufprogramm bearbeitet. Im Anlaufprogramm können Sie durch entsprechende Programmierung von Anlauf-OBs bestimmte Voreinstellungen für Ihr zyklisches Programm festlegen.

Es gibt zwei Anlaufarten: Neustart (Warmstart) und Kaltstart.

Im Betriebszustand "ANLAUF":

- wird das Programm im Anlauf-OB (OB 100 für Neustart (Warmstart) und OB 102 für Kaltstart) abgearbeitet,
- ist keine zeit- und alarmgesteuerte Programmbearbeitung möglich,
- werden die Zeiten aktualisiert,
- läuft der Betriebsstundenzähler,
- sind die Digitalausgänge auf Signalbaugruppen gesperrt, können aber über Direktzugriff gesetzt werden.

Neustart (Warmstart)

Ein Neustart (Warmstart) ist immer zulässig, es sei denn, vom System wurde Urlöschen angefordert. In folgenden Fällen ist nur Neustart (Warmstart) möglich, nach:

- Urlöschen,
- dem Laden des Anwenderprogramms im STOP-Zustand der CPU,
- USTACK/BSTACK-Überlauf,
- Abbruch des Neustart (Warmstart) (durch NETZ-AUS oder über Betriebsartenschalter),
- Überschreiten der parametrisierten Unterbrechungszeitgrenze für Wiederanlauf.

Manueller Neustart (Warmstart)

Ein manueller Neustart (Warmstart) kann ausgelöst werden:

- über den Betriebsartenschalter
- über Menübefehl vom PG aus bzw. über Kommunikationsfunktionen
(wenn der Betriebsartenschalter auf RUN oder RUN-P steht)

Automatischer Neustart (Warmstart)

Ein automatischer Neustart (Warmstart) kann ausgelöst werden bei NETZ-EIN, wenn:

- die CPU bei NETZ-AUS nicht im STOP war.
- der Betriebsartenschalter auf RUN oder RUN-P steht.
- die CPU im Neustart (Warmstart) durch Netzausfall unterbrochen wurde (unabhängig von der Parametrierung der Anlaufart).

Ungepufferter automatischer Neustart (Warmstart)

Wird Ihre CPU ohne Pufferbatterie betrieben (falls wartungsfreier Betrieb notwendig ist), wird nach dem Einschalten oder bei Spannungswiederkehr nach NETZ-AUS die CPU automatisch urgelöscht und anschließend ein Neustart (Warmstart) durchgeführt. Das Anwenderprogramm muss auf Flash-EPROM (Memory Card) vorhanden sein.

Remanente Datenbereiche nach Netzausfall

Mit STEP 7 Lite können Sie die Remanenz von Merkern, Zeiten, Zählern und Bereichen in Datenbausteinen im Rahmen der CPU-Parametrierung festlegen, um Datenverlust bei Spannungsausfall zu vermeiden. Die als remanent parametrisierten Bereiche bleiben bei Netzausfall im CPU-internen NV-RAM erhalten.

Die nachfolgende Tabelle "Remanenzverhalten im Arbeitsspeicher (bei EPROM- und RAM-Ladespeicher)" zeigt das Remanenzverhalten der S7-300-CPUs bei Neustart (Warmstart) und Kaltstart.

Remanenzverhalten im Arbeitsspeicher (bei EPROM- und RAM-Ladespeicher):

Anlaufart (S7-300)	Operanden	Operanden nach Anlauf
Neustart (Warmstart) (CPU ist batteriegepuffert)	Code- und Datenbausteine bleiben erhalten
Neustart (Warmstart) (CPU ist batteriegepuffert)	Merker, Zeiten Zähler bleiben nur dann erhalten, wenn sie als remanent parametrier wurden; sonst werden sie zurückgesetzt
Neustart (Warmstart) (CPU ist nicht batteriegepuffert)	Code- und Datenbausteine werden aus EPROM-Ladespeicher in den Arbeitsspeicher kopiert. Die als remanent parametrieren Datenbaustein-Inhalte bleiben erhalten, nachträglich geladene Bausteine oder per Programm erzeugte DBs gehen verloren
Neustart (Warmstart) (CPU ist nicht batteriegepuffert)	Merker, Zeiten Zähler bleiben nur dann erhalten, wenn sie als remanent parametrier wurden; sonst werden sie zurückgesetzt
Kaltstart	Code- und Datenbausteine werden aus EPROM-Ladespeicher in den Arbeitsspeicher kopiert.
Kaltstart	Merker, Zeiten Zähler werden zurückgesetzt (auch wenn sie als remanent parametrier wurden)

Anlauf Tätigkeiten

Welche Tätigkeiten die CPU beim Anlauf durchführt, zeigt nachfolgende Tabelle:

Tätigkeiten in der Bearbeitungsreihenfolge	bei Neustart (Warmstart)	bei Kaltstart
U-Stack/B-Stack löschen	X	X
Nicht remanente Merker, Zeiten, Zähler löschen	X	0
Alle Merker, Zeiten, Zähler löschen	0	X
Prozessabbild der Ausgänge löschen	X	X
Ausgänge der Signalbaugruppen zurücksetzen	X	X
Prozessalarme verwerfen	X	X
Verzögerungsalarme verwerfen	X	X
Diagnosealarme verwerfen	X	X
Systemzustandsliste (SZL) aktualisieren	X	X
Baugruppenparameter auswerten und an Baugruppen übergeben oder Defaultwerte übergeben	X	X
Bearbeitung des jeweiligen Anlauf-OB	X	X
Restzyklus (Teil des Anwenderprogramms, das aufgrund eines NETZ-AUS nicht weiterbearbeitet werden konnte) bearbeiten	0	0
Prozessabbild der Eingänge aktualisieren	X	X
Digitale Ausgänge freigeben (Signal OD aufheben) nach Betriebszustandsübergang in RUN	X	X
X bedeutet wird durchgeführt 0 bedeutet wird nicht durchgeführt		

Abbrechen eines Anlaufs

Treten während des Anlaufs Fehler auf, wird der Anlauf abgebrochen, und die CPU geht oder bleibt in STOP.

Ein abgebrochener Neustart (Warmstart) muss wiederholt werden.

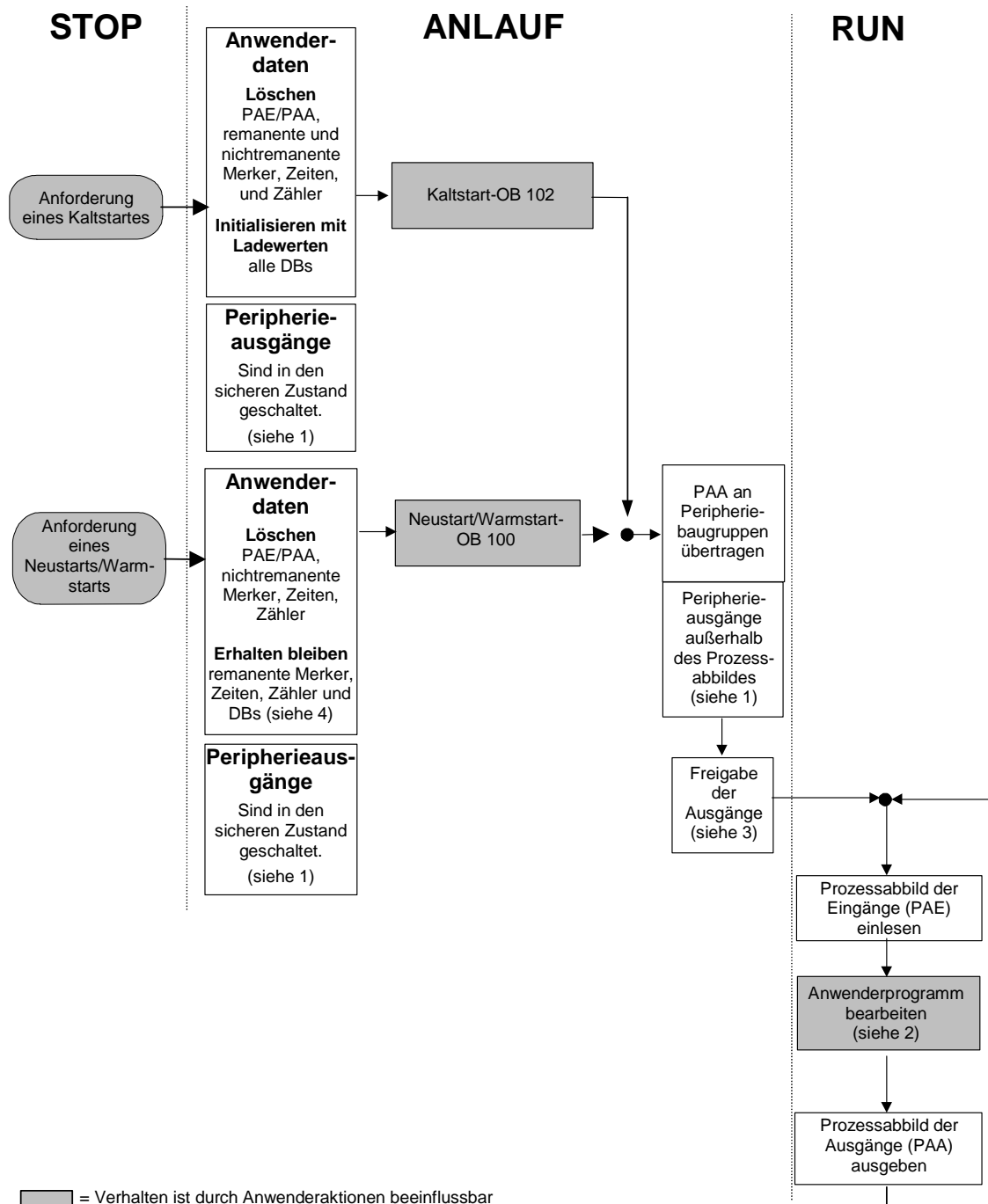
Ein Anlauf (Neustart (Warmstart) oder Wiederanlauf) wird nicht durchgeführt oder er wird abgebrochen, wenn

- der Schlüsselschalter der CPU auf STOP steht,
- Urlöschen angefordert wird,
- eine Memory Card gesteckt ist, deren Anwendungskennung für STEP 7 Lite nicht zulässig ist (z. B. STEP 5),
- das Anwenderprogramm einen OB enthält, den die CPU nicht kennt oder der gesperrt wurde,
- nach dem Einschalten der Netzspannung die CPU feststellt, dass nicht alle Baugruppen, die in der mit STEP 7 Lite erstellten Konfigurationstabelle aufgeführt sind, tatsächlich gesteckt sind (Parametrierung Soll-Ist-Unterschied unzulässig),
- bei der Auswertung der Baugruppenparameter Fehler auftreten.

Ablauf

Nachfolgendes Bild zeigt die Tätigkeiten der CPU in den Betriebszuständen ANLAUF und RUN.

Die dargestellte Anlaufart "Wiederanlauf" ist der Vollständigkeit halber mit aufgenommen (nur S7-400); sie ist nicht relevant für das Baugruppenspektrum, das mit STEP 7 Lite programmiert werden kann.



Legende zum Bild "Tätigkeiten der CPU in den Betriebszuständen ANLAUF und RUN":

1. Alle Peripherieausgänge sind durch die Peripheriebaugruppen hardwareseitig in den sicheren Zustand geschaltet (Defaultwert = "0"). Dies ist unabhängig davon, OB sie im Anwenderprogramm innerhalb oder außerhalb des Prozessabbildbereiches verwendet werden.

Beim Einsatz von ersatzwertfähigen Signalbaugruppen kann das Verhalten der Ausgänge parametrierbar werden, z. B. letzter Wert halten.
2. Auch den Alarm-OBs steht beim erstmaligem Aufruf ein aktuelles Prozessabbild der Eingänge zur Verfügung.
3. Den Zustand der Peripherieausgänge im ersten Zyklus des Anwenderprogramms können Sie durch folgende Maßnahmen festlegen:
 - Parametrierbare Ausgabebaugruppen verwenden, um Ersatzwerte ausgeben zu können oder um den letzten Wert zu halten.
 - Ausgänge im Anlauf-OB (OB 100, OB 102) vorbesetzen.
4. In ungepufferten S7-300-Systemen bleiben nur die als remanent projektierten DB-Bereiche erhalten.

A.1.4 Betriebszustand RUN

Im Betriebszustand "RUN" erfolgt die zyklische, zeit- und alarmgesteuerte Programmbearbeitung:

- das Prozessabbild der Eingänge wird eingelesen
- das Anwenderprogramm wird abgearbeitet
- das Prozessabbild der Ausgänge wird ausgegeben

Der aktive Austausch von Daten zwischen CPUs über die Globaldaten-Kommunikation (Globaldatentabelle) und über Kommunikations-SFBs für projektierte Verbindungen und über Kommunikations-SFCs für nichtprojektierte Verbindungen ist nur im RUN-Zustand möglich.

Nachfolgende Tabelle zeigt beispielhaft, wann Datenaustausch in verschiedenen Betriebszuständen möglich ist:

Art der Kommunikation	Betriebszustand der CPU 1	Richtung des Datenaustauschs	Betriebszustand der CPU 2
Globaldaten-Kommunikation	RUN	↔	RUN
	RUN	→	STOP/HALT
	STOP	←	RUN
	STOP	X	STOP
	HALT	X	STOP/HALT
Einseitige Kommunikation	RUN	→	RUN
über Kommunikations-SFBs	RUN	→	STOP/HALT
Zweiseitige Kommunikation über Kommunikations-SFBs	RUN	↔	RUN
Einseitige Kommunikation	RUN	→	RUN
über Kommunikations-SFCs	RUN	→	STOP/HALT
Zweiseitige Kommunikation über Kommunikations-SFCs	RUN	↔	RUN
↔ bedeutet	Datenaustausch ist in beide Richtungen möglich		
→ bedeutet	Datenaustausch ist nur in eine Richtung möglich		
X	bedeutet Datenaustausch ist nicht möglich		

Betriebsartenschalter und Betriebszustand

- Betriebsartenschalter als **Schlüsselschalter** ausgeführt:
Bei CPUs mit Schlüsselschaltern wird der Betriebszustand in Stellung RUN oder in der Stellung RUN-P des Schlüsselschalters erreicht.
In Stellung RUN ist kein Zugriff auf den Ladespeicher der CPU vom PG/PC möglich, es sei denn, es wurde bei der Parametrierung der CPU ein Passwort für die Schutzstufe 1 parametrierung und das Passwort ist bekannt.
In Stellung RUN-P ist uneingeschränkter Zugriff möglich, sofern kein Passwort bei der Parametrierung der CPU festgelegt wurde
- Betriebsartenschalter als **Kippschalter** ausgeführt (CPU 31xC):
Bei CPUs mit Kippschaltern gibt es nur die Schalterstellung RUN und keine Schalterstellung RUN-P. In Stellung RUN ist uneingeschränkter Zugriff möglich, sofern kein Passwort bei der Parametrierung der CPU festgelegt wurde (entspricht RUN-P bei CPUs mit Schlüsselschalter).

A.1.5 Betriebszustand HALT

Der Betriebszustand "HALT" nimmt eine Sonderstellung ein. Er wird nur zu Testzwecken im Anlauf oder RUN eingenommen. Im Betriebszustand HALT:

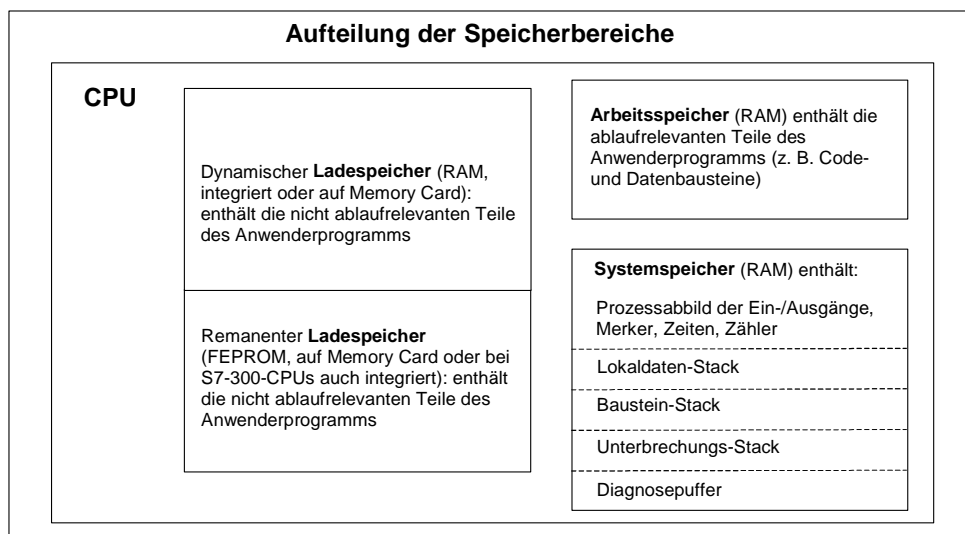
- werden alle Zeiten eingefroren: Zeiten und Betriebsstundenzähler werden nicht bearbeitet, Überwachungszeiten werden angehalten, die Grundtakte der zeitgesteuerten Ebenen werden angehalten
- läuft die Echtzeituhr
- werden Ausgänge nicht freigeschaltet, können aber zu Testzwecken freigegeben werden
- können Ein- und Ausgänge gesteuert werden
- gehen gepufferte CPUs bei Netzausfall und -wiederkehr im Betriebszustand "HALT" in "STOP" und führen keinen automatischen Wiederanlauf oder Neustart (Warmstart) aus. Ungepufferte CPUs führen bei Netzwiederkehr einen ungepufferten automatischen Neustart (Warmstart) aus.
- können auch Globaldaten empfangen und passiv einseitige Kommunikation über Kommunikations-SFBs für projektierte Verbindungen und über Kommunikations-SFCs für nichtprojektierte Verbindungen ausgeführt werden (siehe auch Tabelle in Betriebszustand RUN)

A.2 Speicherbereiche von S7-CPU

A.2.1 Aufteilung der Speicherbereiche

Der Speicher der S7-CPU lässt sich in drei Bereiche aufteilen (siehe nachfolgendes Bild):

- Der Ladespeicher dient zur Aufnahme des Anwenderprogramms ohne symbolische Operandenzuordnung oder Kommentare (diese bleiben im Speicher des PG). Der Ladespeicher kann RAM- oder EPROM-Speicher sein.
- Bausteine, die als nicht ablaufrelevant gekennzeichnet sind, werden ausschließlich in den Ladespeicher aufgenommen.
- Der Arbeitsspeicher (integrierter RAM) dient zur Aufnahme der für den Programmablauf relevanten Teile des Programms. Die Programmbearbeitung erfolgt ausschließlich im Bereich von Arbeitsspeicher und Systemspeicher.
- Der Systemspeicher (RAM) enthält die Speicherelemente, die jede CPU dem Anwenderprogramm zur Verfügung stellt, wie z. B. das Prozessabbild der Ein- und Ausgänge, Merker, Zeiten und Zähler. Außerdem enthält der Systemspeicher den Baustein-Stack und den Unterbrechungs-Stack.
- Der Systemspeicher der CPU stellt außerdem temporären Speicher (Lokaldaten-Stack) zur Verfügung, der dem Programm beim Aufrufen eines Bausteins für dessen temporäre Daten zugeordnet wird. Diese Daten sind nur solange gültig, wie der Baustein aktiv ist.



A.2.2 Lade- und Arbeitsspeicher

Wenn Sie das Anwenderprogramm vom Programmiergerät in die CPU laden, werden nur die Code- und Datenbausteine in den Lade- und den Arbeitsspeicher der CPU geladen.

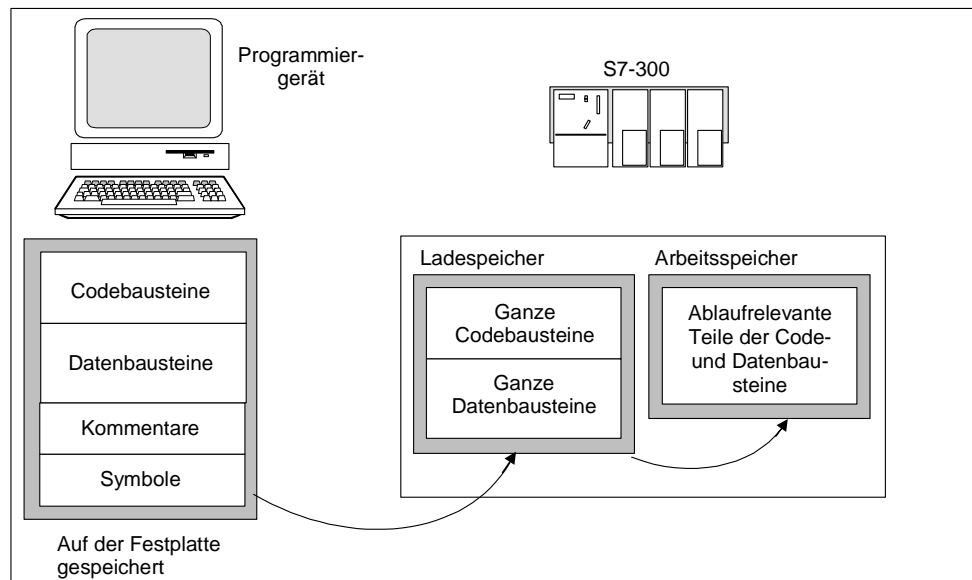
Die symbolische Operandenzuordnung (Symboltabelle) und die Bausteinkommentare bleiben im Speicherbereich des PG.

Aufteilung des Anwenderprogramms

Um eine schnelle Bearbeitung des Anwenderprogramms zu gewährleisten und den nicht erweiterbaren Arbeitsspeicher nicht unnötig zu belasten, werden nur die Teile der Bausteine, die für die Programmbearbeitung relevant sind, in den Arbeitsspeicher geladen.

Bausteinteile, die nicht erforderlich sind, um das Programm ablaufen zu lassen (z. B. Bausteinköpfe), bleiben im Ladespeicher.

Nachfolgendes Bild zeigt das Laden des Programms in den CPU-Speicher.



Hinweis

Datenbausteine, die mit Hilfe von Systemfunktionen (z. B. SFC 22 CREAT_DB) im Anwenderprogramm erstellt werden, speichert die CPU komplett in den Arbeitsspeicher.

Einige CPUs verfügen über getrennt verwaltete Bereiche für Code und Daten im Arbeitsspeicher. Größe und Belegung der Bereiche werden bei diesen CPUs auf der Registerseite "Speicher" des Baugruppenzustands angezeigt.

Kennzeichen von Datenbausteinen als "nicht ablaufrelevant"

Datenbausteine, die als Teil eines AWL-Programms in einer Quelldatei programmiert werden, können als "nicht ablaufrelevant" gekennzeichnet werden (Schlüsselwort UNLINKED). Das bedeutet, beim Laden in die CPU werden diese DBs nur im Ladespeicher abgelegt. Ihr Inhalt kann bei Bedarf mit Hilfe der SFC 20 BLKMOV in den Arbeitsspeicher kopiert werden.

Dadurch kann Platz im Arbeitsspeicher eingespart werden. Der erweiterbare Ladespeicher dient als Zwischenspeicher (z. B. für Rezepturen: nur die Rezeptur, die als nächste bearbeitet werden soll, wird in den Arbeitsspeicher geladen).

Struktur des Ladespeichers

Der Ladespeicher kann durch den Einsatz von Memory Cards erweitert werden. Die maximale Größe des Ladespeichers entnehmen Sie dem Handbuch *Automatisierungssystem S7-300, Aufbauen, CPU-Daten*.

Bei S7-300-CPU's kann der Ladespeicher außer einem integrierten RAM- auch einen integrierten EPROM-Anteil haben. Bereiche in Datenbausteinen können durch Parametrierung mit STEP 7 Lite remanent erklärt werden (siehe Remanente Speicherbereiche in S7-300-CPU's).

Verhalten des Ladespeichers bei RAM- und EPROM-Bereichen

Je nachdem, OB Sie eine RAM- oder EPROM-Memory Card zur Erweiterung des Ladespeichers wählen, kann sich beim Laden, Nachladen und Urlöschen ein unterschiedliches Verhalten des Ladespeichers ergeben.

Nachfolgende Tabelle zeigt die Lademöglichkeiten:

Speicherart	Lademöglichkeiten	Ladeart
RAM	Laden und Löschen von einzelnen Bausteinen	PG-CPU-Verbindung
	Laden und Löschen eines gesamten Programmes	PG-CPU-Verbindung
	Nachladen von einzelnen Bausteinen	PG-CPU-Verbindung
EPROM integriert oder steckbar	Laden von gesamten Programmen	PG-CPU-Verbindung
EPROM steckbar	Laden von gesamten Programmen	Laden des EPROMs auf dem PG und Stecken der Memory Card in die CPU Laden des EPROMs auf der CPU

Programme, die im RAM gespeichert sind, gehen verloren, wenn Sie die CPU urlöschen (MRES) bzw. wenn Sie die CPU oder die RAM-Memory Card ziehen.

Programme, die auf EPROM-Memory-Cards gespeichert sind, gehen beim Urlöschen nicht verloren und bleiben auch ohne Batteriepufferung erhalten (Transport, Sicherungskopien).

A.2.3 Systemspeicher

A.2.3.1 Verwenden der Systemspeicherbereiche

Der Systemspeicher der S7-CPU ist in Operandenbereiche aufgeteilt (siehe nachfolgende Tabelle). Durch Verwendung der entsprechenden Operationen adressieren Sie in Ihrem Programm die Daten direkt in den jeweiligen Operandenbereich.

Operandenbereich	Zugriff über Einheiten der folgenden Größe:	S7- Notation	Beschreibung
Prozessabbild der Eingänge	Eingang (Bit)	E	Zu Beginn jedes Zyklus liest die CPU die Eingänge aus den Eingabebaugruppen und speichert die Werte in das Prozessabbild der Eingänge.
	Eingangsbyte	EB	
	Eingangswort	EW	
	Eingangsdoppelwort	ED	
Prozessabbild der Ausgänge	Ausgang (Bit)	A	Während des Zyklus berechnet das Programm die Werte für die Ausgänge und legt sie im Prozessabbild der Ausgänge ab. Am Ende des Zyklus schreibt die CPU die errechneten Ausgangswerte in die Ausgabebaugruppen.
	Ausgangsbyte	AB	
	Ausgangswort	AW	
	Ausgangsdoppelwort	AD	
Merker	Merker (Bit)	M	Dieser Bereich stellt Speicherplatz für im Programm errechnete Zwischenergebnisse zur Verfügung.
	Merkerbyte	MB	
	Merkerwort	MW	
	Merkerdoppelwort	MD	
Zeiten	Zeit (T)	T	In diesem Bereich stehen Zeiten zur Verfügung.
Zähler	Zähler (Z)	Z	In diesem Bereich stehen Zähler zur Verfügung.
Datenbaustein	Datenbaustein, geöffnet mit "AUF DB":	DB	Datenbausteine speichern Informationen für das Programm. Sie können entweder so definiert sein, dass alle Codebausteine auf sie zugreifen können (globale DBs), oder sie sind einem bestimmten FB oder SFB zugeordnet (Instanz-DB).
	Datenbit	DBX	
	Datenbyte	DBB	
	Datenwort	DBW	
	Datendoppelwort	DBD	

Operandenbereich	Zugriff über Einheiten der folgenden Größe:	S7- Notation	Beschreibung
	Datenbaustein, geöffnet mit "AUF DI":	DI	
	Datenbit	DIX	
	Datenbyte	DIB	
	Datenwort	DIW	
	Datendoppelwort	DID	
Lokaldaten	Lokaldatenbit	L	Dieser Speicherbereich nimmt die temporären Daten eines Bausteins für die Dauer der Bearbeitung dieses Bausteins auf. Der L-Stack stellt auch Speicher zum Übertragen von Bausteinparametern und zum Speichern von Zwischenergebnissen aus KOP-Netzwerken zur Verfügung.
	Lokaldatenbyte	LB	
	Lokaldatenwort	LW	
	Lokaldatendoppelwort	LD	
Peripheriebereich: Eingänge	Peripherieeingangsbyte	PEB	Die Peripheriebereiche der Ein- und Ausgänge erlauben direkten Zugriff auf zentrale und dezentrale Ein- und Ausgabebaugruppen.
	Peripherieeingangswort	PEW	
	Peripherieeingangsdoppelwort	PED	
Peripheriebereich: Ausgänge	Peripherieausgangsbyte	PAB	
	Peripherieausgangswort	PAW	
	Peripherieausgangsdoppelwort	PAD	

Welche Adressbereiche bei Ihrer CPU möglich sind, entnehmen Sie folgender CPU-Beschreibung bzw. Operationsliste:

- Handbuch Automatisierungssystem S7-300, Aufbau, CPU-Daten.
- Operationsliste Automatisierungssystem S7-300.

A.2.3.2 Prozessabbild der Ein-/Ausgänge

Werden im Anwenderprogramm die Operandenbereiche Eingänge (E) und Ausgänge (A) angesprochen, werden nicht die Signalzustände auf den digitalen Signalbaugruppen abgefragt, sondern es wird auf einen Speicherbereich im Systemspeicher der CPU und der dezentralen Peripherie zugegriffen. Diesen Speicherbereich bezeichnet man als Prozessabbild.

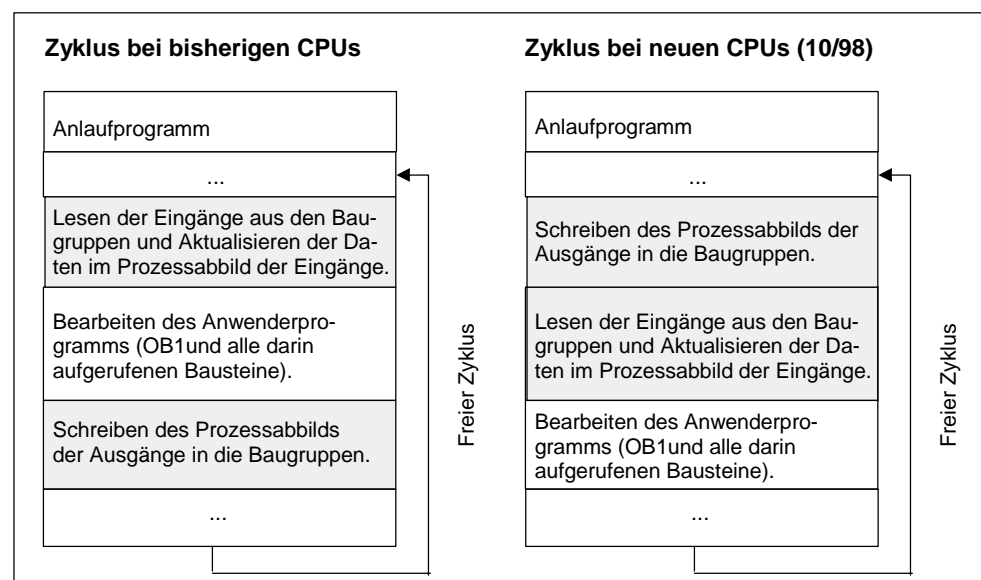
Das Prozessabbild ist in zwei Teile gegliedert: das Prozessabbild der Eingänge und das Prozessabbild der Ausgänge.

Voraussetzung für den Zugriff auf das Prozessabbild

Die CPU kann nur auf das Prozessabbild der Baugruppen zugreifen, die Sie mit STEP 7 Lite konfiguriert haben.

Aktualisieren des Prozessabbilds

Das Prozessabbild wird vom Betriebssystem zyklisch aktualisiert. Nachfolgendes Bild zeigt die Bearbeitungsschritte innerhalb eines Zyklus, unterschieden nach den bisherigen CPUs und nach den neuen ab 10/98 erhältlichen CPUs.



Vorteile des Prozessabbildes

Der Zugriff auf das Prozessabbild hat gegenüber dem direkten Zugriff auf die Ein-/Ausgabebaugruppen den Vorteil, dass der CPU für die Dauer der zyklischen Programmbearbeitung ein konsistentes Abbild der Prozesssignale zur Verfügung steht. Wenn sich während der Programmbearbeitung ein Signalzustand auf einer Eingabebaugruppe ändert, bleibt der Signalzustand im Prozessabbild erhalten bis zur Prozessabbildaktualisierung im nächsten Zyklus. Außerdem benötigt der Zugriff auf das Prozessabbild wesentlich weniger Zeit als der direkte Zugriff auf die Signalbaugruppen, weil sich das Prozessabbild im internen Speicher der CPU befindet.

Peripheriezugriffsfehler (PZF) bei Prozessabbild-Aktualisierung

Die voreingestellte Reaktion der CPU-Familie (S7-300) ist bei einem Fehler während der Prozessabbild-Aktualisierung folgendermaßen:

- Kein Diagnosepuffer-Eintrag, kein OB-Aufruf; die entsprechenden Eingangs-/Ausgangsbytes werden auf 0 gesetzt.

Bei neuen CPUs (ab 4/99) können Sie die Reaktion bei Peripheriezugriffsfehlern umparametrieren:

- OB 85 startet nur bei kommendem und gehendem PZF und erzeugt einen Diagnosepuffer-Eintrag
- kein OB-85-Aufruf (voreingestelltes Verhalten der S7-300).

Wie oft startet der OB 85?

Neben der parametrierten Reaktion auf PZF (kommend/gehend oder bei jedem Peripheriezugriff) hat auch der Adressraum einer Baugruppe Einfluss auf die Häufigkeit des OB-85-Starts:

Bei einer Baugruppe mit einem Adressraum bis zu einem Doppelwort startet der OB 85 einmal, z. B. bei einer Digitalbaugruppe mit bis zu 32 Eingängen oder Ausgängen, oder einer Analogbaugruppe mit 2 Kanälen.

Bei Baugruppen mit größerem Adressraum startet der OB 85 so oft, wie mit Doppelwortbefehlen darauf zugegriffen werden muss, z. B. bei einer 4kanaligen Analogbaugruppe zweimal.

A.2.3.3 Lokaldaten-Stack

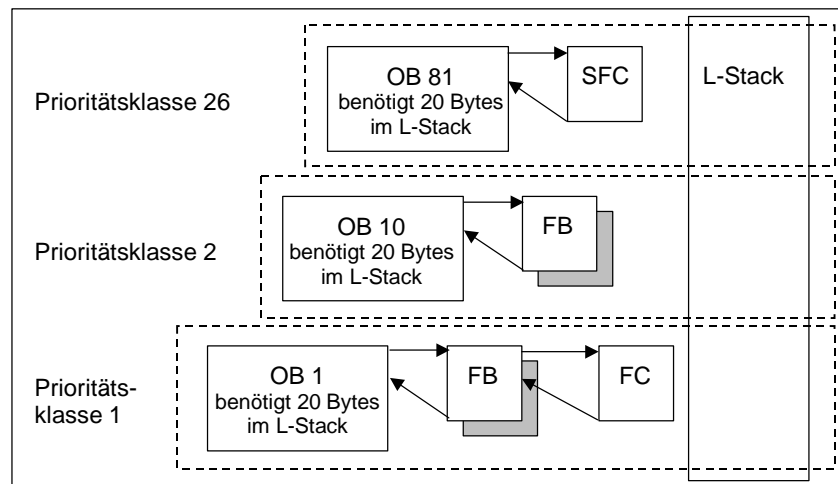
Der L-Stack speichert:

- die temporären Variablen der Lokaldaten von Bausteinen
- die Startinformation der Organisationsbausteine
- Informationen zum Übergeben von Parametern
- Zwischenergebnisse der Logik in Kontaktplan-Programmen

Beim Erstellen von Organisationsbausteinen können Sie temporäre Variablen (temp) deklarieren, die nur während der Bearbeitung des Bausteins zur Verfügung stehen und dann wieder überschrieben werden. Vor dem ersten Zugriff müssen die Lokaldaten initialisiert werden. Außerdem benötigt jeder Organisationsbaustein für seine Startinformation 20 Lokaldaten-Bytes.

Die CPU besitzt einen begrenzten Speicher für die temporären Variablen (Lokaldaten) gerade bearbeiteter Bausteine. Die Größe dieses Speicherbereichs, des Lokaldaten-Stacks, ist CPU-abhängig. Der Lokaldaten-Stack wird zu gleichen Teilen unter den Prioritätsklassen aufgeteilt (Voreinstellung). Das bedeutet, jede Prioritätsklasse verfügt über einen eigenen Lokaldatenbereich. Damit ist gewährleistet, dass auch hochprioritäre Prioritätsklassen und ihre zugeordneten OBs Platz für ihre Lokaldaten zur Verfügung haben.

Nachfolgendes Bild zeigt die Zuordnung von Lokaldaten zu den Prioritätsklassen in einem Beispiel, in dem im L-Stack der OB 1 vom OB 10 unterbrochen wird, der dann wiederum vom OB 81 unterbrochen wird.



Vorsicht

Alle temporäre Variablen (temp) eines OB und seiner unterlagerten Bausteine werden im L-Stack gespeichert. Wenn Sie viele Schachtelungsebenen in Ihrer Bausteinbearbeitung verwenden, kann der L-Stack überlaufen.

S7-CPU's wechseln in den Betriebszustand "STOP", wenn Sie die zulässige L-Stack-Größe für ein Programm überschreiten.

Testen Sie den L-Stack (die temporären Variablen) in Ihrem Programm.

Berücksichtigen Sie den Lokaldatenbedarf von Synchronfehler-OBs.

Zuordnen von Lokaldaten zu Prioritätsklassen

Bei den S7-300-CPU's ist jeder Prioritätsklasse eine feste Anzahl von Lokaldaten zugeordnet (256 Bytes), die nicht verändert werden kann.

A.2.3.4 Unterbrechungs-Stack

Wird die Programmbearbeitung durch einen OB mit höherer Priorität unterbrochen, speichert das Betriebssystem die aktuellen Inhalte der Akkumulatoren und Adressregister sowie die Nummer und die Größe der geöffneten Datenbausteine im U-Stack.

Ist die Bearbeitung des neuen OB beendet, lädt das Betriebssystem die Informationen aus dem U-Stack und nimmt die Bearbeitung des unterbrochenen Bausteins an der Stelle wieder auf, an der die Unterbrechung eintrat.

Im Betriebszustand "STOP" können Sie mit STEP 7 Lite am PG den U-Stack auslesen. Damit können Sie leichter die Ursache dafür erkennen, warum die CPU in den Betriebszustand "STOP" gewechselt hat.

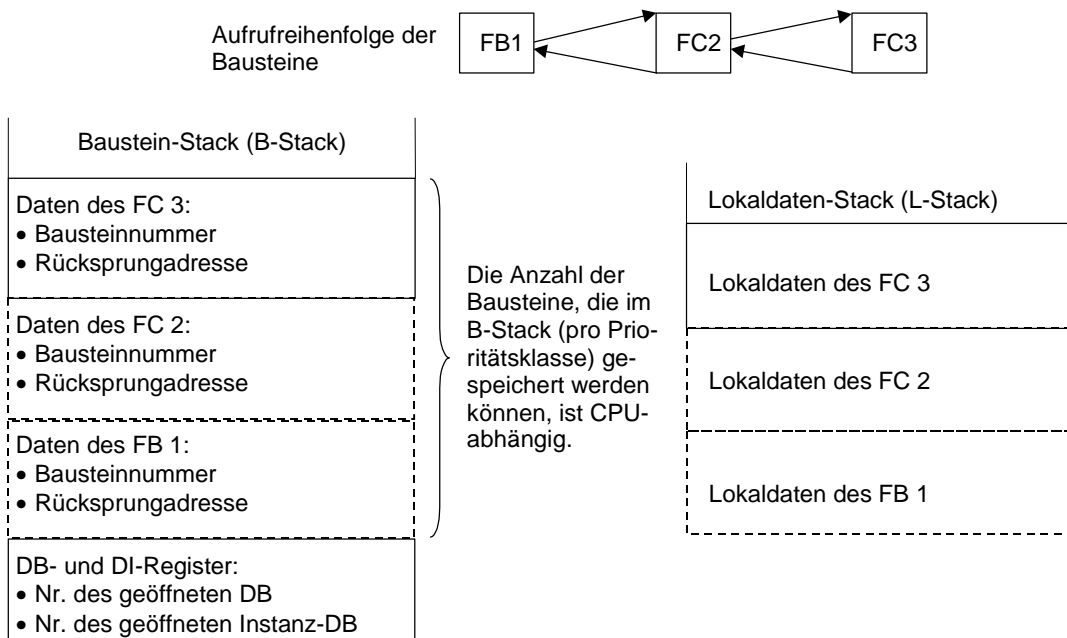
A.2.3.5 Baustein-Stack

Wird die Bearbeitung eines Bausteins durch den Aufruf eines anderen Bausteins oder durch eine höhere Prioritätsklasse (Alarm/Fehlerbehandlung) unterbrochen, speichert der B-Stack folgende Daten:

- Nummer, Art (OB, FB, FC, SFB, SFC) und Rücksprungadresse des Bausteins, der unterbrochen wurde.
- Nummer der Datenbausteine (aus DB- und DI-Register), die zum Zeitpunkt der Unterbrechung geöffnet waren.

Mit den gespeicherten Daten kann das Anwenderprogramm nach der Unterbrechung wieder fortgesetzt werden.

Befindet sich die CPU im Betriebszustand "STOP", können Sie den B-Stack mit STEP 7 Lite am PG anzeigen lassen. Der B-Stack führt alle Bausteine auf, deren Bearbeitung zu dem Zeitpunkt, als die CPU in den Betriebszustand "STOP" versetzt wurde, nicht beendet war. Die Bausteine werden in der Reihenfolge aufgelistet, in der die Bearbeitung begonnen wurde (siehe nachfolgendes Bild).



Datenbausteinregister

Es existieren zwei Datenbausteinregister; sie beinhalten die Nummern der aufgeschlagenen Datenbausteine.

- Im DB-Register steht die Nummer des aufgeschlagenen globalen Datenbausteins.
- Im DI-Register steht die Nummer des aufgeschlagenen Instanz-Datenbausteins.

A.2.3.6 Diagnosepuffer

Im Diagnosepuffer werden die Diagnoseereignisse in der Reihenfolge ihres Auftretens angezeigt. Der erste Eintrag enthält das neueste Ereignis. Die Anzahl der Einträge im Diagnosepuffer ist abhängig von der jeweiligen Baugruppe und ihres aktuellen Betriebszustandes.

Diagnoseereignisse können sein

- Fehler auf einer Baugruppe
- Fehler in der Prozessverdrahtung
- Systemfehler in der CPU
- Betriebszustandsübergänge der CPU
- Fehler im Anwenderprogramm
- Anwenderdefinierte Diagnoseereignisse (über die Systemfunktion SFC 52).

A.2.3.7 Auswertung des Diagnosepuffers

Ein Teil der Systemzustandsliste ist der Diagnosepuffer, in den zu System-Diagnoseereignissen und anwenderdefinierten Diagnoseereignissen in der Reihenfolge ihres Auftretens nähere Informationen eingetragen werden. Die Information, die beim Auftreten eines System-Diagnoseereignisses in den Diagnosepuffer eingetragen wird, ist identisch mit der Startinformation, die an den entsprechenden Organisationsbaustein übergeben wird.

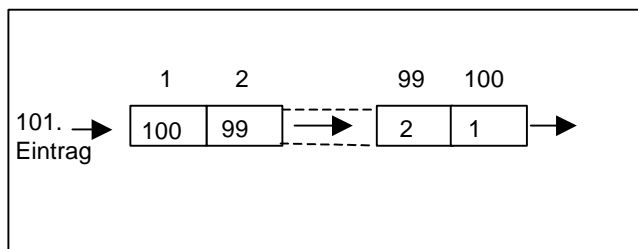
Die Einträge im Diagnosepuffer können nicht gelöscht werden, der Inhalt des Diagnosepuffers bleibt auch nach dem Urlöschen erhalten.

Der Diagnosepuffer bietet die Möglichkeit:

- bei Anlagenstopp die letzten Ereignisse vor dem STOP auszuwerten und die STOP-Ursache zu finden
- Fehlerursachen schneller zu erkennen und dadurch die Verfügbarkeit der Anlage zu erhöhen
- das dynamische Anlagenverhalten auszuwerten und zu optimieren

Organisation des Diagnosepuffers

Der Diagnosepuffer ist als Ringpuffer für eine baugruppenabhängige maximale Anzahl von Einträgen ausgelegt. Wird die maximale Anzahl von Einträgen erreicht, so wird bei einem neuen Diagnosepufferereignis der älteste Eintrag gelöscht. Alle Einträge rücken entsprechend weiter. Dadurch steht der jüngste Diagnoseeintrag immer an erster Stelle. Für die S7-300 CPU 314 sind dies z. B. 100 Einträge:



Die Anzahl der angezeigten Einträge im Diagnosepuffer ist abhängig von der Baugruppe und ihrem aktuellen Betriebszustand. Bei bestimmten CPUs ist die Länge des Diagnosepuffers parametrierbar.

Inhalt des Diagnosepuffers

Das **obere** Listenfeld enthält eine Liste aller aufgetretenen Diagnoseereignisse mit folgenden Informationen:

- Laufende Nummer des Eintrags (das neueste Ereignis hat die Nr. 1)
- Uhrzeit und Datum des Diagnoseereignisses: Angezeigt werden Uhrzeit und Datum der Baugruppe, falls auf der Baugruppe eine Uhr vorhanden ist. Um diese Zeitangaben sinnvoll nutzen zu können, ist es wichtig, dass Sie die Uhrzeit und das Datum auf der Baugruppe einstellen und gelegentlich überprüfen.
- Ereignistext (Kurzdarstellung)

Im **unteren** Textfeld werden zusätzliche Informationen zu dem im oberen Fenster ausgewählten Ereignis angezeigt. Dazu gehören z. B.:

- Ereignisnummer,
- Bezeichnung des Ereignisses,
- Betriebszustandswechsel, der durch das Diagnoseereignis verursacht wurde,
- Verweis auf Fehlerstelle in einem Baustein (Bausteintyp, -nummer, Relativadresse), die zum Eintrag des Ereignisses führte,
- Kommendes oder gehendes Ereignis,
- ereignisspezifische Zusatzinformationen.

Mit der Schaltfläche "Hilfe zum Ereignis" können Sie sich zusätzliche Information zu dem im Listenfeld markierten Ereignis anzeigen lassen.

Erläuterungen zu den Ereignis-IDs finden Sie in der Referenzhilfe zu Systembausteinen und Systemfunktionen (Sprünge in Sprachbeschreibungen, Baustein-Hilfen, Systemattribute).

Speichern des Inhalts in einer Textdatei

Mit der Schaltfläche "Speichern unter" im Register "Diagnosepuffer" des Dialogfelds "Baugruppenzustand" können Sie den Inhalt des Diagnosepuffers als ASCII-Text speichern.

Diagnosepuffer auslesen

Sie können sich den Inhalt des Diagnosepuffers über das Dialogfeld "Baugruppenzustand", Register "Diagnosepuffer" am PG/PC anzeigen lassen oder in einem Programm über die SFC 51 RDSYSST auslesen.

Letzter Eintrag vor STOP

Sie können festlegen, dass der letzte Diagnosepuffereintrag vor dem Übergang von RUN in STOP automatisch an ein angemeldetes Beobachtungsgerät (z. B. PG, OP, TD) gesendet wird, um sicherzustellen, dass die Ursache für den Betriebszustandswechsel in STOP schneller gefunden und behoben wird.

A.2.3.8 Remanente Speicherbereiche in S7-300-CPU

Generelle Remanenz

Diagnosepuffer, MPI-Parameter und Betriebsstundenzähler sind generell remanent. Diese Daten bleiben sowohl nach Netzausfall als auch nach Umräumen erhalten.

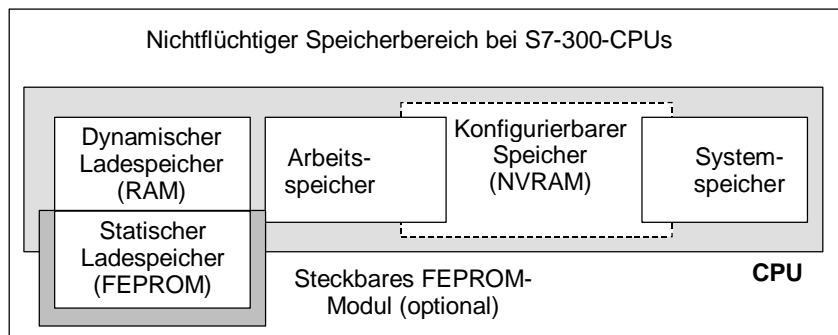
Pufferung der Daten bei CPU 31x

Wenn der Speicher (dynamischer Ladespeicher (RAM), Arbeitsspeicher und Systemspeicher) einer S7-300-CPU nicht gepuffert ist, gehen alle Daten, die in diesen Bereichen gespeichert sind, bei Stromausfall verloren. Sie haben folgende Möglichkeiten, Programm bzw. Daten zu puffern:

- Sie können alle Daten, die sich im Ladespeicher, Arbeitsspeicher und in Teilen des Systemspeichers befinden, durch eine Batterie puffern.
- Sie können Ihr Programm im EPROM speichern (entweder Memory Card oder in die CPU integriert, siehe Handbuch *Automatisierungssystem S7-300, Aufbauen, CPU-Daten*).
- Sie können eine CPU-abhängige Menge Daten in einem Bereich des nichtflüchtigen NVRAM sichern.

Verwenden des NVRAM

Ihre S7-300-CPU stellt einen Bereich im NVRAM (nichtflüchtiger RAM) zur Verfügung (siehe nachfolgendes Bild). Wenn Sie Ihr Programm im EPROM des Ladespeichers abgelegt haben, können Sie einige Daten (bei Stromausfall oder beim Übergang der CPU von STOP in RUN) auch durch eine entsprechende Konfiguration speichern.



Hierzu stellen Sie Ihre CPU so ein, dass die folgenden Daten in dem nichtflüchtigen RAM gespeichert werden:

- Daten, die in einem DB gespeichert sind (dies ist nur nützlich, wenn Sie auch Ihr Programm in einem EPROM des Ladespeichers abgelegt haben)
- Werte von Zeiten und Zählern
- Daten, die in Merkern gespeichert sind

Bei jeder CPU können Sie eine bestimmte Anzahl an Zeiten, Zählern und Merkern puffern. Außerdem wird eine spezifische Anzahl an Bytes zur Verfügung gestellt, in denen die Daten, die in DBs abgelegt sind, gespeichert werden können.

Verwenden der Batteriepufferung zum Sichern von Daten

Durch Batteriepufferung werden der Ladespeicher und der Arbeitsspeicher bei Stromausfall remanent. Wenn durch Ihre Konfiguration Zeiten, Zähler und Merker im NVRAM gespeichert werden, gehen unabhängig von der Batteriepufferung auch diese Informationen nicht verloren.

Konfigurieren der Daten des NVRAM

Wenn Sie die CPU mit STEP 7 Lite konfigurieren, können Sie festlegen, welche Speicherbereiche remanent sein sollen.

Wieviel Speicher im NVRAM konfiguriert werden kann, ist CPU-abhängig. Sie können nicht mehr Daten puffern, als die für Ihre CPU angegebene Menge.

Pufferung von Daten bei CPU 31xC ("Kompakt-CPU")

Der Ladespeicher der CPU 31xC ist komplett auf einer Micro Memory Card (MMC) untergebracht. Seine Größe entspricht genau der MMC.

Das Laden von Programmen und der Betrieb der CPUs 31xC ist nur mit gesteckter MMC möglich.

Remanente Objekte:

- Anwenderprogramm im Ladespeicher (auf der MMC),
- die als remanent parametrisierten Merker, Zeiten, Zähler (im Systemspeicher)
- Inhalte von Datenbausteinen (werden bei Netzausfall vom Arbeitsspeicher auf MMC gesichert).

Bei CPUs, welche die Retaineigenschaft von DBs unterstützen (z. B. CPU 317 V2.1) ist dies der Fall, wenn Sie die Option "Non-Retain" (Eigenschaft des Datenbausteins) deaktiviert haben. Wenn Sie das Optionskästchen "Non-Retain" aktivieren, werden nach jedem Netz-Aus- und Netz-Einschalten und nach jedem STOP-RUN-Übergang der CPU die Datenbaustein-Inhalte auf die Ladewerte zurückgesetzt.

A.3 Daten- und Parametertypen

A.3.1 Einführung zu Daten- und Parametertypen

Alle in einem Anwenderprogramm verwendeten Daten müssen durch einen Datentyp gekennzeichnet sein. Man unterscheidet zwischen:

- elementaren Datentypen, die Ihnen STEP 7 Lite zur Verfügung stellt
- zusammengesetzten Datentypen, die Sie erzeugen können, indem Sie elementare Datentypen verknüpfen, und
- Parametertypen, mit denen Sie Parameter definieren, die an FBs oder FCs übergeben werden sollen.

Allgemeine Information

AWL-, FUP- oder KOP-Operationen arbeiten mit Datenobjekten bestimmter Größen. Bitverknüpfungsoperationen arbeiten beispielsweise mit Bits. Lade- und Transferoperationen (AWL) sowie Übertragungsoperationen (FUP und KOP) arbeiten mit Bytes, Wörtern und Doppelwörtern.

Bit nennt man eine Binärziffer "0" oder "1". Ein Byte besteht aus 8 Bits, ein Wort aus 16 Bits und ein Doppelwort aus 32 Bits.

Arithmetische Operationen arbeiten mit Bytes, Wörtern oder Doppelwörtern. In diesen Byte-, Wort- oder Doppelwortoperanden können Sie Zahlen verschiedener Formate codieren, wie z. B. Ganzzahlen und Gleitpunktzahlen.

Wenn Sie die symbolische Adressierung verwenden, dann definieren Sie Symbole und geben einen Datentyp für diese Symbole an (siehe nachfolgende Tabelle). Verschiedene Datentypen haben verschiedene Formatoptionen und Zahlendarstellungen.

Dieses Kapitel beschreibt nur einige der Schreibweisen, die für Zahlen und Konstanten möglich sind. Nachfolgende Tabelle listet Formate von Zahlen und Konstanten auf, die nicht ausführlich eingegangen wird.

Format	Größe in Bits	Zahlendarstellung
Hexadezimal	8, 16 und 32	B#16#, W#16# und DW#16#
Binär	8, 16 und 32	2#
IEC-Datum	16	D#
IEC-Zeit	32	T#
Uhrzeit	32	TOD#
Zeichen	8	'A'

A.3.2 Elementare Datentypen

Jeder elementare Datentyp hat eine definierte Länge. Nachfolgende Tabelle listet die elementaren Datentypen auf.

Typ und Beschreibung	Größe in Bits	Formatoptionen	Bereich und Zahlendarstellung (niedrigster bis höchster Wert)	Beispiel
BOOL (Bit)	1	Bool-Text	TRUE/FALSE	TRUE
BYTE (Byte)	8	Hexadezimalzahl	B#16#0 bis B#16#FF	L B#16#10 L byte#16#10
WORD (Wort)	16	Dualzahl Hexadezimalzahl BCD Dezimalzahl ohne Vorzeichen	2#0 bis 2#1111_1111_1111_1111 W#16#0 bis W#16#FFFF C#0 bis C#999 B#(0,0) bis B#(255,255)	L 2#0001_0000_0000_0000 L W#16#1000 L word#16#1000 L C#998 L B#(10,20) L byte#(10,20)
DWORD (Doppelwort)	32	Dualzahl Hexadezimalzahl Dezimalzahl ohne Vorzeichen	2#0 bis 2#1111_1111_1111_1111_1111_1111_1111_1111 DW#16#0000_0000 bis DW#16#FFFF_FFFF B#(0,0,0,0) bis B#(255,255,255,255)	2#1000_0001_0001_1000_1011_1011_0111_1111 L DW#16#00A2_1234 L dword#16#00A2_1234 L B#(1, 14, 100, 120) L byte#(1,14,100,120)
INT (Ganzzahl)	16	Dezimalzahl mit Vorzeichen	-32768 bis 32767	L 1
DINT (Ganzzahl, 32 Bit)	32	Dezimalzahl mit Vorzeichen	L#-2147483648 bis L#2147483647	L L#1
REAL (Gleitpunktzahl)	32	IEEE Gleitpunktzahl	Obere Grenze: $\pm 3.402823e+38$ 0 Untere Grenze: $\pm 1.175 495e-38$	L 1.234567e+13
S5TIME (SIMATIC-Zeit)	16	S7-Zeit in Schritten von 10 ms (Defaultwert)	S5T#0H_0M_0S_10MS bis S5T#2H_46M_30S_0MS und S5T#0H_0M_0S_0MS	L S5T#0H_1M_0S_0MS L S5TIME#0H_1H_1M_0S_0MS
TIME (IEC-Zeit)	32	IEC-Zeit in Schritten von 1 ms, Ganzzahl mit Vorzeichen	-T#24D_20H_31M_23S_648MS bis T#24D_20H_31M_23S_647MS	L T#0D_1H_1M_0S_0MS L TIME#0D_1H_1M_0S_0MS
DATE (IEC-Datum)	16	IEC-Datum in Schritten von 1 Tag	D#1990-1-1 bis D#2168-12-31	L D#1994-3-15 L DATE#1994-3-15
TIME_OF_DAY (Uhrzeit)	32	Uhrzeit in Schritten von 1 ms	TOD#0:0:0.0 bis TOD #23:59:59.999	L TOD#1:10:3.3 L TIME_OF_DAY#1:10:3.3
CHAR (Zeichen)	8	ASCII-Zeichen	'A','B' usw.	L 'E'

A.3.2.1 Format des Datentyps INT (16-Bit-Ganzzahlen)

Eine Ganzzahl hat ein Vorzeichen, das angibt, OB es sich um eine positive oder negative ganze Zahl handelt. Der Platz, den eine Ganzzahl (16 Bit) im Speicher belegt, beträgt ein Wort. Nachfolgende Tabelle zeigt den Bereich einer Ganzzahl (16 Bit).

Format	Bereich
Ganzzahl (16 Bit)	-32 768 bis +32 767

Nachfolgendes Bild zeigt die Ganzzahl +44 als Dualzahl.

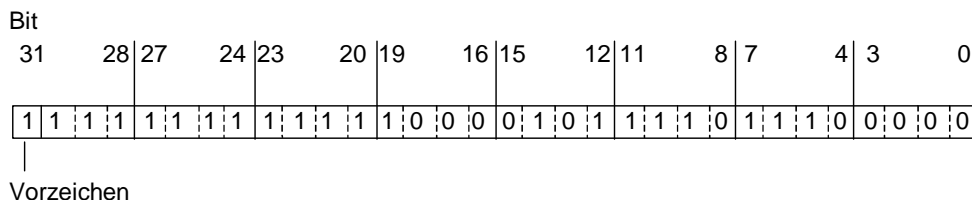


A.3.2.2 Format des Datentyps DINT (32-Bit-Ganzzahlen)

Eine Ganzzahl hat ein Vorzeichen, das angibt, OB es sich um eine positive oder negative ganze Zahl handelt. Der Platz, den eine Ganzzahl (32 Bit) im Speicher belegt, beträgt zwei Wörter. Nachfolgende Tabelle zeigt den Bereich einer Ganzzahl (32 Bit).

Format	Bereich
Ganzzahl (32 Bit)	-2 147 483 648 bis +2 147 483 647

Nachfolgendes Bild zeigt die Ganzzahl -500 000 als Dualzahl. Im Dualsystem wird die negative Form einer Ganzzahl als Zweierkomplement der positiven Ganzzahl dargestellt. Das Zweierkomplement einer Ganzzahl erhalten Sie, indem Sie die Signalzustände aller Bits umkehren und dann +1 zum Ergebnis addieren.



A.3.2.3 Format des Datentyps REAL (Gleitpunktzahlen)

Zahlen im Gleitpunktformat werden in der allgemeinen Form "Zahl = $m \cdot b^{\text{hoch } E}$ " dargestellt. Die Basis "b" und der Exponent "E" sind ganze Zahlen, die Mantisse "m" eine rationale Zahl.

Diese Zahlendarstellung hat den Vorteil, dass auf begrenztem Raum sehr große und sehr kleine Werte dargestellt werden können. Mit der begrenzten Anzahl von Bits für Mantisse und Exponent lässt sich ein weiter Zahlenbereich abdecken.

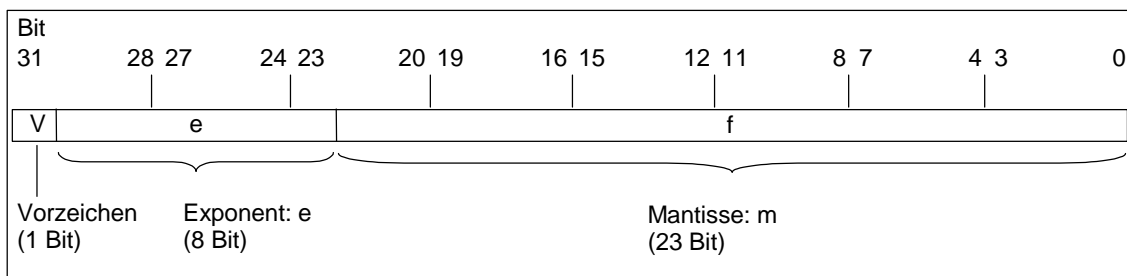
Der Nachteil liegt in der begrenzten Rechengenauigkeit: Beispielsweise müssen bei der Summenbildung zweier Zahlen die Exponenten durch Verschieben der Mantisse (gleitender Dezimalpunkt) angeglichen werden (Addition der Mantissen zweier Zahlen mit gleichem Exponent).

Gleitpunktformat in STEP 7 Lite

Gleitpunktzahlen in STEP 7 Lite entsprechen dem Grundformat mit einfacher Breite, wie in der Norm ANSI/IEEE Standard 754–1985, *IEEE Standard for Binary Floating-Point Arithmetic* beschrieben. Sie bestehen aus den folgenden Komponenten:

- dem Vorzeichen V
- dem um eine Konstante (Bias = +127) erhöhten Exponenten $e = E + \text{Bias}$
- dem gebrochenen Teil der Mantisse m
Der ganzzahlige Anteil der Mantisse wird nicht mit abgelegt, da er innerhalb des gültigen Zahlenbereichs immer = 1 ist.

Die drei Bestandteile belegen zusammen ein Doppelwort (32 Bit):



Die folgende Tabelle zeigt die Wertigkeit der einzelnen Bits im Gleitpunktformat.

Bestandteil der Gleitpunktzahl	Bit-Nummer	Wertigkeit
Vorzeichen V	31	
Exponent e	30	$2^{\text{hoch } 7}$
...
Exponent e	24	$2^{\text{hoch } 1}$
Exponent e	23	$2^{\text{hoch } 0}$
Mantisse m	22	$2^{\text{hoch } -1}$
...
Mantisse m	1	$2^{\text{hoch } -22}$
Mantisse m	0	$2^{\text{hoch } -23}$

Mit den drei Komponenten **V**, **e** und **m** ist der Wert einer in diesem Format dargestellten Zahl definiert über die Formel:

$$\text{Zahl} = 1.\mathbf{m} * 2^{\text{hoch}(\mathbf{e}\text{-Bias})}$$

Dabei gilt:

- e: $1 \leq e \leq 254$
- Bias: Bias = 127. Somit entfällt ein extra Vorzeichen für den Exponent.
- V: für eine positive Zahl ist V = 0 und für negative Zahl ist V = 1.

Wertebereich der Gleitpunktzahlen

Anhand des oben dargestellten Gleitpunktformats ergeben sich die

- kleinste Gleitpunktzahl = $1.0 * 2^{\text{hoch}(1-127)} = 1.0 * 2^{\text{hoch}(-126)} = 1.175\ 495\text{E}-38$ und
- größte Gleitpunktzahl = $2 * 2^{\text{hoch}(-23)} * 2^{\text{hoch}(254-127)} = 2 * 2^{\text{hoch}(-23)} * 2^{\text{hoch}(+127)} = 3.402\ 823\text{E}+38$

Die Zahl Null wird mit e = m = 0 dargestellt; e = 255 und m = 0 steht für "unendlich".

Format	Bereich ¹⁾
Gleitpunktzahlen nach ANSI/IEEE Standard	-3.402 823E+38 bis -1.175 495E-38 und 0 und +1.175 495E-38 bis +3.402 823E+38

Die nächste Tabelle zeigt den Signalzustand der Bits im Statuswort für die Ergebnisse von Operationen mit Gleitpunktzahlen, die nicht innerhalb des gültigen Bereichs liegen.

Ungültiger Bereich für ein Ergebnis	A1	A0	OV	OS
-1.175494E-38 < Ergebnis < -1.401298E-45 (negative Zahl) Unterschreitung	0	0	1	1
+1.401298E-45 < Ergebnis < +1.175494E-38 (positive Zahl) Unterschreitung	0	0	1	1
Ergebnis < -3.402823E+38 (negative Zahl) Überlauf	0	1	1	1
Ergebnis > 3.402823E+38 (positive Zahl) Überlauf	1	0	1	1
keine gültige Gleitpunktzahl oder unzulässige Operation (Eingangswert außerhalb des gültigen Wertebereichs)	1	1	1	1

Vorsicht bei mathematischen Operationen:

Das Ergebnis "Keine gültige Gleitpunktzahl" erhält man beispielsweise bei dem Versuch, die Quadratwurzel aus -2 zu ziehen. Werten Sie deshalb bei mathematischen Operationen stets zuerst die Statusbits aus, bevor Sie mit dem Ergebnis weiterrechnen.

Vorsicht beim "Steuern von Variablen":

Legt man die Werte für Gleitkommaoperationen z.B. in Merkerdoppelwörtern ab, so kann man diese Werte mit beliebigen Bitmustern verändern. Jedoch ist nicht jedes Bitmuster eine gültige Zahl!

Genauigkeit beim Rechnen mit Gleitpunktzahlen**Vorsicht**

Bei umfangreichen Berechnungen mit Zahlen, die große Größenunterschiede (mehrere 10-er Potenzen) aufweisen, können Ungenauigkeiten im Rechenergebnis entstehen.

Die Gleitpunktzahlen in STEP 7 Lite sind auf 6 Dezimalstellen genau. Sie können deshalb bei der Angabe von Gleitpunkt konstanten nur maximal 6 Nachkommastellen angeben.

Hinweis

Die Rechengenauigkeit von 6 Dezimalstellen bedeutet z. B., dass die Addition von $Zahl1 + Zahl2 = Zahl1$ ergibt, wenn $Zahl1$ größer $Zahl2 * 10$ hoch y , mit $y > 6$ ist:

$$100\,000\,000 + 1 = 100\,000\,000.$$

Beispiele für Zahlen in Gleitpunktdarstellung

Nachfolgendes Bild zeigt das Gleitpunktzahlenformat für die folgenden dezimalen Werte:

- 10,0
- p (3,141593)
- Quadratwurzel aus 2 ($p_2 = 1,414214$)

Die Zahl **10.0** im ersten Beispiel ergibt sich aus ihrem Gleitpunktformat (Darstellung in HEX: 4120 0000) wie folgt:

$$e = 2 \text{ hoch } 1 + 2 \text{ hoch } 7 = 2 + 128 = 130$$

$$m = 2 \text{ hoch } (-2) = 0,25$$

Damit ergibt sich: $1.m * 2 \text{ hoch } (e - \text{Bias}) = 1.25 * 2 \text{ hoch } (130 - 127) = 1.25 * 2 \text{ hoch } 3 = 10.0.$

Dezimalwert 10,0

Hexadezimalwert	4	1	2	0	0	0	0	0								
Bits	31	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0

0 1 0 0 0 0 0 1 0 0 1 0

Mantissen- Exponent: e Mantisse: f
 vorzeichen: (8 Bit) (23 Bit)
 V (1 Bit)

$e = 27 + 21 = 130$
 $1.f \cdot 2^{e-bias} = 1,25 \cdot 2^{-23} = 10,0$
 $[1,25 \cdot 2^{-(130-127)} = 1,25 \cdot 2^{-3} = 10,0]$

$f = 2^{-2} = 0,25$

Dezimalwert 3,141593

Hexadezimalwert	4	0	4	9	0	F	D	C								
Bits	31	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0

0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 0 0

Mantissen- Exponent: e Mantisse: f
 vorzeichen: (8 Bit) (23 Bit)
 V (1 Bit)

Dezimalwert 1,414214

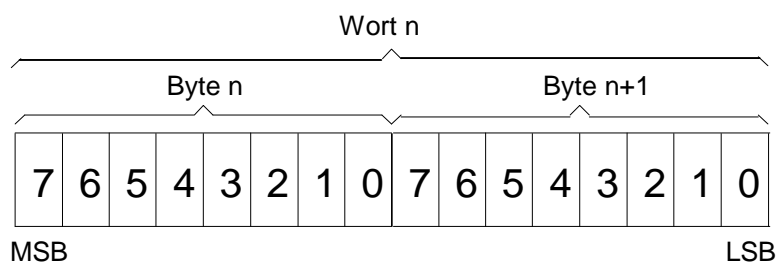
Hexadezimalwert	3	F	B	5	0	4	F	7								
Bits	31	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0

0 0 1 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 0 0 0 1 0 0 1 1 1 1 1 0 1 1 1

Mantissen- Exponent: e Mantisse: f
 vorzeichen: (8 Bit) (23 Bit)
 V (1 Bit)

A.3.2.4 Format des Datentyps WORD

Datentyp	Länge (Bit)	Format	Beispiele für das Format	
			Min.	Max.
WORD	16	Binär	2#0	2#1111111111111111
		Hexadezimal	W#16#0	W#16#FFFF
		Vorzeichenlose Bytes	B#(0,0)	B#(255,255)

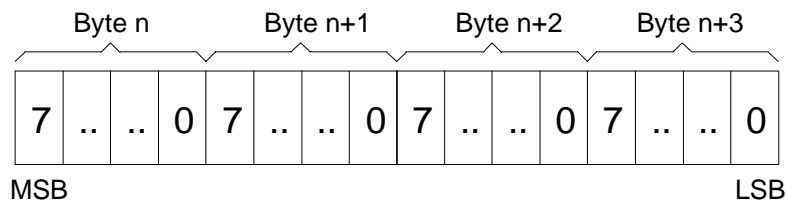


MSB: Most Significant Bit

LSB: Least Significant Bit

A.3.2.5 Format des Datentyps DWORD

Datentyp	Länge (Bit)	Format	Beispiele für das Format	
			Min.	Max.
DWORD	32	Binär, Min.	2#0	
		Binär, Max.	2#11111111111111111111111111111111	
		Hexadezimal	DW#16#0	DW#16#FFFFFFFF
		Vorzeichenlose Bytes	B#(0,0,0,0)	B#(255,255,255,255)



MSB: Most Significant Bit

LSB: Least Significant Bit

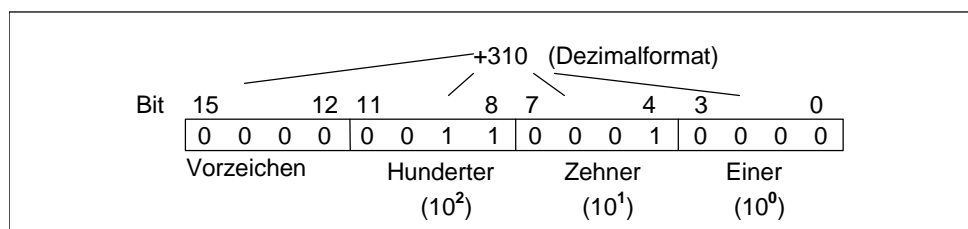
A.3.2.6 Format der Datentypen WORD und DWORD bei binär-codierten Dezimalzahlen

Die binär-codierte Darstellung (BCD) stellt eine Dezimalzahl durch Gruppen von Binärziffern (Bits) dar. Eine Gruppe von 4 Bits stellt eine Ziffer einer Dezimalzahl oder das Vorzeichen der Dezimalzahl dar. Die Gruppen von 4 Bits bilden ein Wort (16 Bits) oder Doppelwort (32 Bits). Die vier höchstwertigen Bits geben das Vorzeichen der Zahl an ("1111" bedeutet Minus und "0000" Plus). Befehle mit BCD-codierten Operanden werten nur das höchstwertige Bit (15 bei Wort-, 31 bei Doppelwortformat) aus. Nachfolgende Tabelle zeigt Format und Bereich für die beiden Arten der BCD-Zahlen.

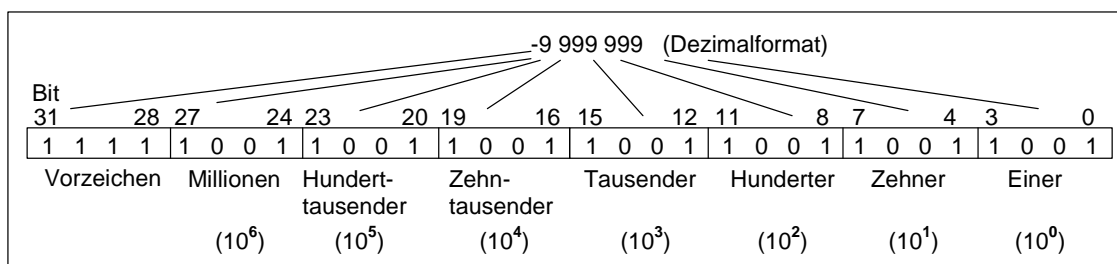
Format	Bereich
Wort (16 Bits, 3stellige BCD-Zahl mit Vorzeichen)	-999 bis +999
Doppelwort (32 Bits, 7stellige BCD-Zahl mit Vorzeichen)	-9 999 999 bis +9 999 999

Die nachfolgenden Bilder zeigen Beispiele für eine binär-codierte Dezimalzahl in folgenden Formaten:

- Wortformat

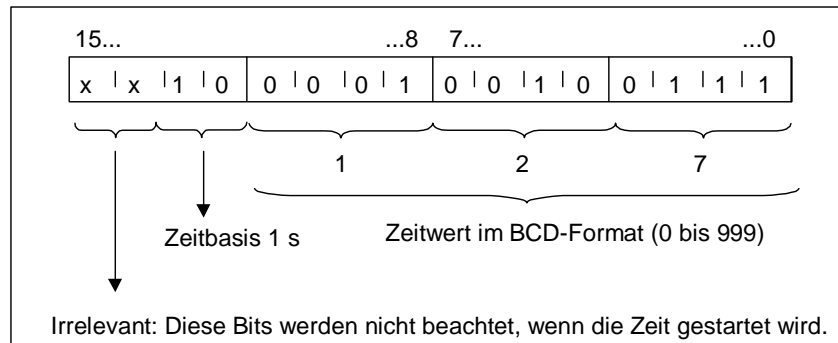


- Doppelwortformat



A.3.2.7 Format des Datentyps S5TIME (Zeitdauer)

Wenn Sie die Zeitdauer mit dem Datentyp S5TIME eingeben, werden Ihre Eingaben im BCD-Format gespeichert. Nachfolgendes Bild zeigt den Inhalt des Zeitoperanden bei einem Zeitwert von 127 und einer Zeitbasis von 1 s.



Wenn Sie mit S5TIME arbeiten, geben Sie einen Zeitwert im Bereich von 0 bis 999 ein und bezeichnen eine Zeitbasis (siehe nachfolgende Tabelle). Die Zeitbasis gibt das Intervall an, in dem eine Zeit den Zeitwert um je eine Einheit vermindert, bis er "0" erreicht.

Zeitbasis für S5TIME

Zeitbasis	Binärkode für Zeitbasis
10 ms	00
100 ms	01
1 s	10
10 s	11

Mit der folgenden Syntax können Sie einen vordefinierten Zeitwert laden:

- L¹⁾ W#16#wxyz
 - mit: w = Zeitbasis (d. h. Zeitintervall oder Auflösung)
 - xyz = Zeitwert im BCD-Format
- L¹⁾ S5T#aH_bbM_ccS_dddMS
 - mit: a = Stunden, bb = Minuten, cc = Sekunden und ddd = Millisekunden.
 - Die Zeitbasis wird automatisch gewählt und der Wert zur nächstniederen Zahl mit dieser Zeitbasis gerundet.

Sie können einen Zeitwert von max. 9 990 Sekunden bzw. 2H_46M_30S eingeben.

¹⁾ = L nur bei AWL-Programmierung angeben

A.3.2.8 Format des Datentyps TIME

Datentyp	Länge (Bit)	Format
TIME	32	Zeitraum mit Vorzeichen: + oder - Tage, Stunden, Minuten, Sekunden, Millisekunden

Beispiele für das Format (obere und untere Grenzwerte)

Max.	T#+24d20h31m23s647ms
Min.	T# -24d20h31m23s648ms

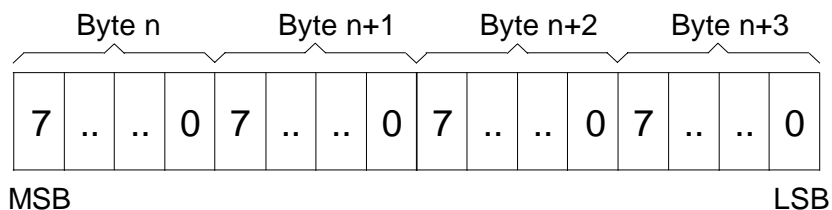
Es ist nicht erforderlich, alle Zeiteinheiten anzugeben (z.B.: T#5h10s ist gültig).

Wenn nur eine Einheit angegeben wird, darf der absolute Wert an Tagen, Stunden und Minuten die oberen oder unteren Grenzwerte nicht überschreiten.

T# -65535 und T#+65535 sind die oberen und unteren Grenzwerte für Sekunden und Millisekunden.

Wenn mehr als eine Zeiteinheit angegeben wird, darf die Einheit

- Stunden den Wert 23,
- Minuten den Wert 59,
- Sekunden den Wert 59,
- Millisekunden den Wert 999 nicht überschreiten.



MSB: Most Significant Bit

LSB: Least Significant Bit

Hinweis

Der Datentyp TIME wird als Ganzzahl mit Vorzeichen in Millisekunden als Zweierkomplement gespeichert.

A.3.3 Zusammengesetzte Datentypen

Zusammengesetzte Datentypen definieren Datengruppen, die größer sind als 32 Bits bzw. Datengruppen, die sich aus anderen Datentypen zusammensetzen. STEP 7 Lite lässt die folgenden zusammengesetzten Datentypen zu:

- DATE_AND_TIME
- STRING
- ARRAY (Feld)
- STRUCT (Struktur)
- UDT (anwenderdefinierte Datentypen)
- FB und SFB

Nachfolgende Tabelle beschreibt die zusammengesetzten Datentypen. Sie definieren Strukturen und Felder entweder in der Variablendeklaration des Codebausteins oder in einem Datenbaustein.

Datentyp	Beschreibung
DATE_AND_TIME DT	Definiert einen Bereich mit 64 Bits (8 Bytes). Dieser Datentyp speichert in binärcodiertem Dezimalformat.
STRING	Definiert eine Gruppe von maximal 254 Zeichen (Datentyp CHAR). Der Standardbereich, der für einen Zeichenkette reserviert ist, besteht aus 256 Bytes. Dies ist der Platz, der benötigt wird, um 254 Zeichen und einen Kopf von 2 Bytes zu speichern. Sie können den Speicherplatz für eine Zeichenkette verringern, indem Sie auch die Anzahl der Zeichen definieren, die in der Zeichenkette gespeichert werden sollen (z. B. string[9] 'Siemens').
ARRAY	Definiert eine multidimensionale Gruppierung eines Datentyps (entweder elementar oder zusammengesetzt). Beispiel: "ARRAY [1..2,1..3] OF INT" definiert ein Feld im Format 2 x 3 aus Ganzzahlen. Sie greifen auf die Daten, die in einem Feld gespeichert sind, über den Index ("[2,2]") zu. Sie können maximal 6 Dimensionen in einem Feld definieren. Der Index kann ein beliebiger ganzzahliger Wert (-32768 bis 32767) sein.
STRUCT	Definiert eine Gruppierung von beliebig kombinierten Datentypen. Sie können z. B. ein Feld aus Strukturen oder eine Struktur aus Strukturen und Feldern definieren.
UDT	Ermöglicht es, große Datenmengen zu strukturieren und das Eingeben der Datentypen zu vereinfachen, wenn Datenbausteine erzeugt oder Variablen in der Variablendeklaration deklariert werden sollen. In STEP 7 Lite können Sie zusammengesetzte und elementare Datentypen verknüpfen und so Ihren eigenen "anwenderdefinierten" Datentyp erstellen. UDTs haben einen eigenen Namen und sind deshalb mehrfach verwendbar.
FB, SFB	Bestimmen die Struktur des zugeordneten Instanz-Datenbausteins und ermöglichen die Übergabe von Instanzdaten für mehrere FB-Aufrufe in einem Instanz-DB.

Strukturierte Datentypen werden auf Wortgrenzen ausgerichtet (WORD aligned) abgelegt.

A.3.3.1 Format des Datentyps DATE_AND_TIME (Datum und Uhrzeit)

Wenn Sie Datum und Uhrzeit mit dem Datentyp DATE_AND_TIME (DT) eingeben, werden Ihre Eingaben im BCD-Format in 8 Bytes gespeichert. Der Datentyp DATE_AND_TIME umfasst den folgenden Bereich:

DT#1990-1-1-0:0:0.0 bis DT#2089-12-31-23:59:59.999

Die folgenden Beispiele zeigen die Syntax, mit der Sie das Datum und die Uhrzeit für Donnerstag, den 25. Dezember 1993, 8:01 Uhr und 1,23 Sekunden eingeben. Die folgenden beiden Formate sind möglich:

- DATE_AND_TIME#1993-12-25-8:01:1.23
- DT#1993-12-25-8:01:1.23

Die folgenden IEC-Standardfunktionen (International Electrotechnical Commission) stehen für das Arbeiten mit dem Datentyp DATE_AND_TIME zur Verfügung:

- Umwandeln des Datums und der Uhrzeit in das Format DATE_AND_TIME
FC3: D_TOD_DT
- Herauslösen des Datums aus dem Format DATE_AND_TIME
FC6: DT_DATE
- Herauslösen des Wochentags aus dem Format DATE_AND_TIME
FC7: DT_DAY
- Herauslösen der Uhrzeit aus dem Format DATE_AND_TIME
FC8: DT_TOD

In nachfolgender Tabelle ist der Inhalt der Bytes aufgeführt, die die Information zu Datum und Uhrzeit enthalten. Das Beispiel zeigt Datum und Uhrzeit für Donnerstag, den 25. Dezember 1993, 8:01 Uhr und 1,23 Sekunden.

Byte	Inhalt	Beispiel
0	Jahr	B#16#93
1	Monat	B#16#12
2	Tag	B#16#25
3	Stunde	B#16#08
4	Minute	B#16#01
5	Sekunde	B#16#01
6	Die beiden höchstwertigen Ziffern von MSEC	B#16#23
7 (4MSB)	Die niederwertigen Ziffern von MSEC	B#16#0
7 (4LSB)	Wochentag 1 = Sonntag 2 = Montag ... 7 = Samstag	B#16#5

Der erlaubte Bereich für den Datentyp "DATE_AND_TIME" ist:

- min.: DT#1990-1-1-0:0:0.0
- max.: DT#2089-12-31-23:59:59.999

	Möglicher Wertebereich	BCD-Code
Jahr	1990 – 1999	90h – 99h
	2000 – 2089	00h – 89h
Monat	1 – 12	01h – 12h
Tag	1 – 31	01h – 31h
Stunde	00 – 23	00h – 23h
Minute	00 – 59	00h – 59h
Sekunde	00 – 59	00h – 59h
Millisekunde	0 – 999	000h – 999h
Wochentag	Sonntag – Samstag	1h – 7h

A.3.3.2 Format des Datentyps STRING

Eine Zeichenkette (STRING) umfasst eine Gruppe von maximal 254 Zeichen (Datentyp CHAR). Der Standardbereich, der für diesen Datentyp reserviert wird, besteht aus 256 Bytes (254 Byte für die Zeichen und 2 Byte für den Kopf des Datentyps STRING). Sie können den Speicherbedarf für eine Variable dieses Typs verringern, indem Sie hinter dem Schlüsselwort STRING auch die Anzahl der Zeichen definieren, die in der Zeichenkette gespeichert werden sollen.

Beispiel: STRING [7] = 'SIEMENS'. Die Zeichenkette muss in einfachen Hochkommata angegeben werden.

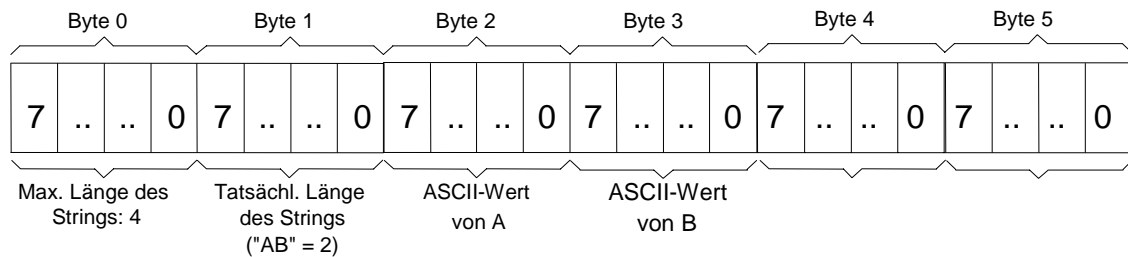
Datentyp	Länge (Byte)	Format
STRING[n] oder STRING	n+2	ASCII-Zeichenkette beliebiger Länge. n gibt die Länge der Zeichenkette an. Maximale Länge sind 254 Zeichen. Wird keine Länge angegeben, ist die Voreinstellung 254 Zeichen.

Datentyp	Beispiele für das Format
STRING[2]	'AB'
STRING[55]	'Die Zeichenkette kann aus maximal 55 Zeichen bestehen.'

Hinweis

Sie müssen Ihre Zeichenkette in einzelne Hochkommata einschließen.

Das folgende Beispiel zeigt die Reihenfolge der Bytes bei der Angabe des Datentyps STRING[4] mit dem Ausgangswert 'AB'.

Beispiel:

Dynamische Lokaldaten vom Datentyp STRING müssen vom Anwender vor dem ersten Verwenden initialisiert werden, z. B. durch eine AWL-Sequenz der Form:

```
LAR1    P#lokal_string_var    // lokal_string_var ist in var_temp deklariert
                                //als STRING[200]

L  200                                // oben genannte STRING-Länge
T  LB [AR1, P#0.0]                // im MAX Len Byte des Strings eintragen
L  5                                  // tatsächliche Länge des Strings
T  LB [AR1, P#1.0]                // tatsächliche Länge des Strings eintragen
```

Hinweis

Wird der Inhalt eines Strings vom Anwenderprogramm geändert, muss auch das Byte "Tatsächliche Länge" beschrieben bzw. aktualisiert werden, damit der String vom PG angezeigt werden kann.

Wurde eine temporäre Variable vom Datentyp STRING definiert, muss das Byte "Max. Länge" vor der Verwendung der Variablen im Anwenderprogramm mit der definierten Länge beschrieben werden.

A.3.3.3 Format des Datentyps ARRAY

Ein Feld (= ARRAY) ist ein zusammengesetzter Datentyp mit bis zu 6 Dimensionen. Alle Elemente in einem Feld können beliebige Datentypen sein (außer Parametertypen), es müssen jedoch alle Elemente vom gleichen Datentyp sein. ARRAYS können nicht geschachtelt werden und müssen aus mindestens zwei Elementen bestehen.

Beispiel: "ARRAY [1..2,1..3] OF INT" definiert ein Feld im Format 2 x 3 aus Ganzzahlen.

Sie greifen auf die Daten über den Index ("[m,n]") zu, wobei $1 \leq m \leq 2$ und $1 \leq n \leq 3$ zu beachten ist.

Der Index kann ein beliebiger ganzzahliger Wert sein. In der Deklaration müssen die ARRAY-Grenzen so beschaffen sein, dass das ARRAY insgesamt maximal 65535 Elemente umfasst. Die Grenzwerte einer Dimension (z. B.: x1 und x2) können negativ, Null oder positiv sein, die Wertangabe für die obere Grenze der Dimension (x2) muss aber höher sein, als die für die untere (x1).

Angeben von Dimensionen

Beispiele: Eindimensional: ARRAY[x1..x2]

ARRAY[-2..-1]

ARRAY[0..1]

ARRAY[1..2]

Zusätzliche Dimensionen werden durch Komma abgetrennt.

Beispiel: Dreidimensional: ARRAY[x1..x2, y1..y2, z1..z2]

Bearbeiten der Variablenliste

ARRAYs können im Programm verwendet werden. In der Initialisierungsspalte können Sie Anfangswerte vorgeben.

Anfangswerte

Elemente werden mit einer Liste von Werten initialisiert, die durch Komma voneinander getrennt sind. Sie können einen Wiederholfaktor verwenden, beispielsweise "4(10)" ("ordnet den folgenden vier Elementen den Wert 10 zu"), um in einem Feld Anfangswerte zuzuordnen. Elemente, die über keinen Anfangswert verfügen, haben standardmäßig den Defaultwert Null.

Symbolische Adressierung

Ihre Programmanweisungen können über den Variablennamen auf die Feldwerte zugreifen.

Beispiel:

L	#feld[3]	Lade den Wert, der im Element 3 des ARRAYs mit dem Namen "feld" steht, in den AKKU1.
---	----------	--

A.3.3.4 Format des Datentyps STRUCT

Eine Struktur (STRUCT) ist ein zusammengesetzter Datentyp, der in bis zu 8 Ebenen geschachtelt werden kann. Elemente einer Struktur können beliebige gültige Datentypen sein. Der Datentyp STRUCT muss aus mindestens zwei Komponenten bestehen, die sich zwischen STRUCT und END_STRUCT befinden. Eine Struktur kann in der Variablendeklarationstabelle eines Codebausteins oder in einem anwenderdefinierten Datentyp (UDT) deklariert werden.

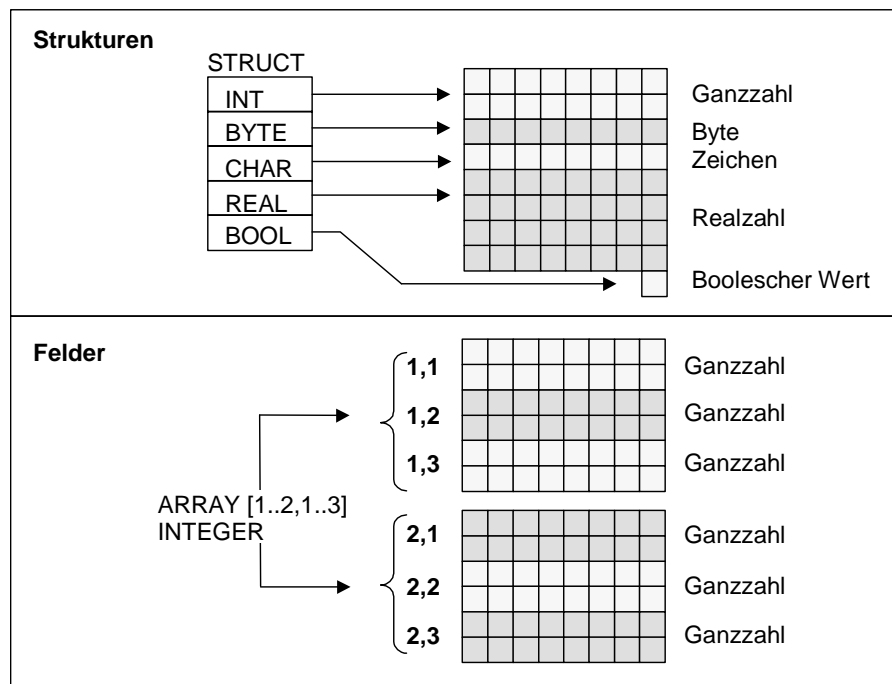
Über <Strukturname.Variablenname> können Sie auf die einzelnen Variablen einer Struktur im Anwenderprogramm zugreifen. Eine Struktur innerhalb einer Struktur zählt nur als eine Komponente.

A.3.3.5 Verwenden von zusammengesetzten Datentypen

Sie können neue Datentypen erstellen, indem Sie elementare und zusammengesetzte Datentypen zu den folgenden zusammengesetzten Datentypen verbinden:

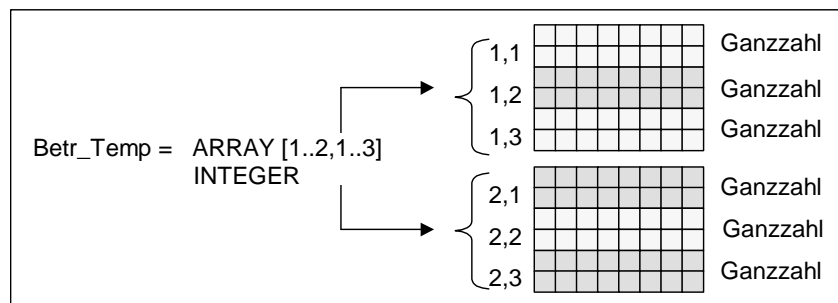
- **Feld (Datentyp ARRAY):** Ein Feld verknüpft eine Gruppe eines Datentyps zu einer Einheit.
- **Struktur (Datentyp STRUCT):** Eine Struktur verknüpft verschiedene Datentypen zu einer Einheit.
- **Zeichenkette (Datentyp STRING):** Eine Zeichenkette definiert ein eindimensionales Feld mit maximal 254 Zeichen (Datentyp CHAR). Eine Zeichenkette kann nur als Einheit übergeben werden. Die Länge der Zeichenkette muss bei dem Formal- und dem Aktualparameter des Bausteins übereinstimmen.
- **Datum und Zeit (Datentyp DATE_AND_TIME):** Das Datum und die Zeit speichern Jahr, Monat, Tag, Stunden, Minuten, Sekunden, Millisekunden und Wochentag.

Nachfolgendes Bild zeigt, wie Felder und Strukturen Datentypen in einem Bereich strukturieren, um so Informationen zu speichern. Sie definieren ein Feld oder eine Struktur entweder in einem DB oder in der Variablendeklaration eines FB, OB oder einer FC.



Beispiel 2

Ein Feld kann auch eine multidimensionale Gruppe von Datentypen beschreiben. Nachfolgendes Bild zeigt ein zweidimensionales Feld aus Ganzzahlen.



Sie greifen auf die Daten in einem multidimensionalen Feld über den Index zu. Bei dem Beispiel ist die erste Ganzzahl `Betr_Temp[1,1]`, die dritte ist `Betr_Temp[1,3]`, die vierte ist `Betr_Temp[2,1]` und die sechste ist `Betr_Temp[2,3]`.

Sie können maximal sechs Dimensionen (sechs Indizes) für ein Feld definieren. Sie könnten beispielsweise die Variable `Betr_Temp` folgendermaßen als sechsdimensionales Feld definieren:

```
ARRAY [1..3,1..2,1..3,1..4,1..3,1..4]
```

Der Index des ersten Elements in diesem Feld ist `Betr_Temp[1,1,1,1,1,1]`. Der Index des letzten Elements ist `Betr_Temp[3,2,3,4,3,4]`.

Erstellen von Feldern

Sie definieren Felder, wenn Sie Daten in einem DB oder in der Variablendeklaration deklarieren. Wenn Sie das Feld deklarieren, dann geben Sie das Schlüsselwort (ARRAY) an, danach die Größe in eckigen Klammern:

```
[unterer Grenzwert..oberer Grenzwert]
```

Bei einem multidimensionalen Feld geben Sie die zusätzlichen oberen und unteren Grenzwerte an und trennen die einzelnen Dimensionen durch ein Komma. Nachfolgendes Bild zeigt die Deklaration zum Erstellen eines Felds im Format 2 x 3.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Waerme_2x3	ARRAY [1..2,1..3]		
*2.0		INT		
=6.0		END STRUCT		

Eingeben von Anfangswerten für ein Feld

Sie können beim Erstellen der Felder jedem Feldelement einen Anfangswert zuordnen. STEP 7 Lite bietet Ihnen zwei Möglichkeiten zum Eingeben von Anfangswerten:

- Eingeben von individuellen Werten: Für jedes Element des Felds geben Sie einen gültigen Wert an (für den Datentyp des Felds). Geben Sie die Werte in der Reihenfolge der Elemente an: [1,1]. Beachten Sie, dass Sie die einzelnen Elemente durch Komma voneinander trennen müssen.
- Angeben eines Wiederholfaktors: Bei sequenziellen Elementen, die über den gleichen Anfangswert verfügen, können Sie die Anzahl der Elemente (den Wiederholfaktor) und den Anfangswert für diese Elemente angeben. Das Format zum Eingeben des Wiederholfaktors ist $x(y)$, wobei x der Wiederholfaktor und y der Wert ist, der wiederholt werden soll.

Wenn Sie das Feld verwenden, das im vorherigen Bild deklariert wird, können Sie den Anfangswert für alle sechs Elemente folgendermaßen angeben: 17, 23, -45, 556, 3342, 0. Sie könnten den Anfangswert von allen sechs Elementen auch auf 10 setzen, indem Sie 6(10) angeben. Sie könnten für die ersten beiden Elemente bestimmte Werte angeben und dann die übrigen vier Elemente auf 0 setzen, indem Sie 17, 23, 4(0) angeben.

Zugreifen auf Daten in einem Feld

Sie greifen auf die Daten in einem Feld über den Index des bestimmten Elements in dem Feld zu. Der Index wird mit dem symbolischen Namen verwendet.

Beispiel: Wenn das im vorherigen Bild deklarierte Feld an dem ersten Byte von DB20 (Motor) beginnt, greifen Sie auf das zweite Element in dem Feld über folgende Adresse zu:

```
Motor.Waerme_2x3[1,2].
```

Verwenden von Feldern als Parameter

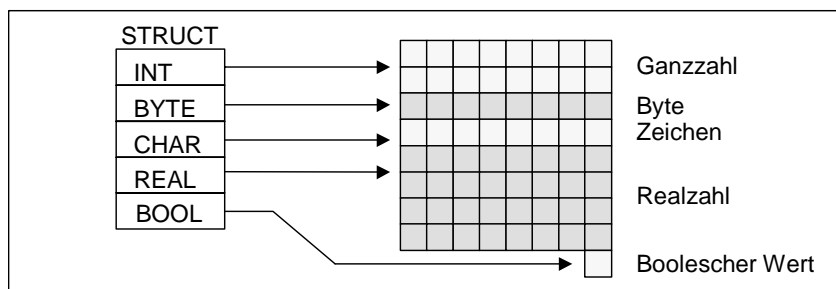
Sie können Felder als Parameter übergeben. Wenn ein Parameter in der Variablendeklaration als ARRAY deklariert ist, müssen Sie das ganze Feld übergeben (und nicht nur einzelne Elemente). Ein Element eines Felds kann jedoch einem Parameter zugeordnet werden, wenn Sie einen Baustein aufrufen, sofern das Element des Felds dem Datentyp des Parameters entspricht.

Wenn Sie Felder als Parameter verwenden, müssen die Felder nicht den gleichen Namen haben (sie brauchen gar keinen Namen zu haben). Beide Felder (der Formalparameter und der Aktualparameter) müssen aber die gleiche Struktur vorweisen. Ein Feld im Format 2 x 3 aus Ganzzahlen zum Beispiel kann als Parameter nur übergeben werden, wenn der Formalparameter des Bausteins ein Feld im Format 2 x 3 aus Ganzzahlen definiert und auch der Aktualparameter, der über die Aufrufoperation bereitgestellt wird, ein Feld im Format 2 x 3 aus Ganzzahlen ist.

A.3.3.7 Verwenden von Strukturen für den Datenzugriff

Strukturen

Eine Struktur verknüpft verschiedene Datentypen (elementare und zusammengesetzte Datentypen, einschließlich Felder und Strukturen) zu einer Einheit. So können Sie Daten entsprechend der Steuerung für Ihren Prozess gruppieren. Sie können so auch Parameter als eine Dateneinheit übergeben, und nicht als einzelne Elemente. Nachfolgendes Bild zeigt eine Struktur, die aus einer Ganzzahl, einem Byte, einem Zeichen, einer Gleitpunktzahl und einem booleschen Wert besteht.



Eine Struktur kann in maximal 8 Ebenen geschachtelt sein (z. B. eine Struktur aus Strukturen, die Felder enthält).

Erstellen einer Struktur

Sie definieren Strukturen beim Deklarieren von Daten innerhalb eines DB oder in der Variablendeklaration eines Codebausteins.

Nachfolgendes Bild zeigt die Deklaration einer Struktur (*Stapel_1*), die aus folgenden Elementen besteht: eine Ganzzahl (zum Speichern der Menge), ein Byte (zum Speichern der Ursprungsdaten), ein Zeichen (zum Speichern des Steuercode), eine Gleitpunktzahl (zum Speichern der Temperatur) und ein boolescher Merker (zum Beenden des Signals).

Adresse	Name	Typ	Anfangswert	Kommentar
0.0	Stapel_1	STRUCT		
+0.0	Menge	INT	100	
+2.0	Ursprungsdaten	BYTE		
+4.0	Steuercode	CHAR		
+6.0	Temperatur	REAL	120	
+8.1	Ende	BOOL	FALSE	
=10.0		END_STRUCT		

Zuordnen von Anfangswerten für eine Struktur

Wenn Sie jedem Element einer Struktur einen Anfangswert zuordnen möchten, geben Sie jeweils einen gültigen Wert für den Datentyp und den Namen des Elements ein. Sie können beispielsweise (der im vorherigen Bild deklarierten Struktur) die folgenden Anfangswerte zuordnen:

Menge	=	100
Ursprungsdaten	=	B#(0)
Steuercode	=	'Z'
Temperatur	=	120
Ende	=	False

Speichern von und Zugreifen auf Daten in Strukturen

Sie greifen auf die einzelnen Elemente einer Struktur zu. Sie können symbolische Adressen (z. B. *Stapel_1.Temperatur*) verwenden. Sie können aber auch die absolute Adresse angeben, unter der das Element gespeichert ist (Beispiel: ist *Stapel_1* in DB20 mit Beginn bei Byte 0 gespeichert, dann ist die absolute Adresse für die Menge *DB20.DBW0* und die Adresse für die Temperatur ist *DB20.DBD6*).

Verwenden von Strukturen als Parameter

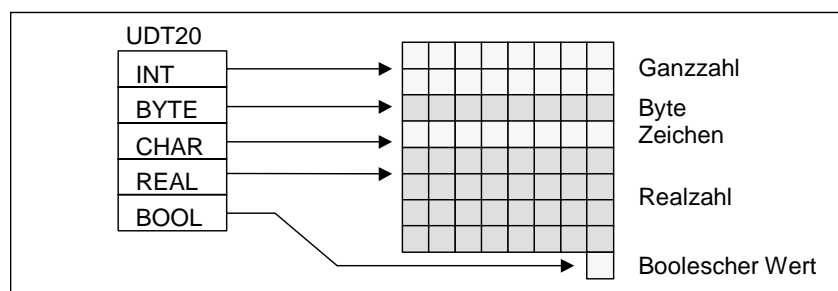
Sie können Strukturen als Parameter übergeben. Wenn ein Parameter in der Variablendeklaration als STRUCT deklariert ist, müssen Sie eine Struktur mit dem gleichen Aufbau übergeben. Ein Element einer Struktur kann jedoch einem Parameter zugeordnet werden, wenn Sie einen Baustein aufrufen, sofern das Element der Struktur dem Datentyp des Parameters entspricht.

Wenn Sie Strukturen als Parameter verwenden, müssen beide Strukturen (für den Formalparameter und den Aktualparameter) den gleichen Aufbau der Daten vorweisen, d. h. die gleichen Datentypen müssen in der gleichen Reihenfolge angeordnet sein.

A.3.3.8 Verwenden von anwenderdefinierten Datentypen für den Datenzugriff

Anwenderdefinierte Datentypen

Anwenderdefinierte Datentypen (UDT) können elementare und zusammengesetzte Datentypen verknüpfen. Sie können UDTs mit einem Namen versehen und mehrfach verwenden. Nachfolgendes Bild zeigt die Struktur eines UDT, der aus einer Ganzzahl, einem Byte, einem Zeichen, einer Gleitpunktzahl und einem booleschen Wert besteht.



Anstatt alle Datentypen einzeln oder als Struktur einzugeben, müssen Sie nur "UDT20" als Datentyp angeben und STEP 7 Lite ordnet automatisch den entsprechenden Speicherplatz zu.

Erstellen eines anwenderdefinierten Datentyps

UDTs definieren Sie mit STEP 7 Lite. Nachfolgendes Bild zeigt einen UDT, der aus den folgenden Elementen besteht: eine Ganzzahl (zum Speichern der Menge), ein Byte (zum Speichern der Ursprungsdaten), ein Zeichen (zum Speichern des Steuercodes), eine Gleitpunktzahl (zum Speichern der Temperatur) und ein boolescher Merker (zum Beenden des Signals). Sie können dem UDT in der Symboltabelle einen symbolischen Namen zuordnen (z. B. Prozessdaten).

Adresse	Name	Typ	Anfangswert	Kommentar
0.0	Stapel 1	STRUCT		
+0.0	Menge	INT	100	
+2.0	Ursprungsdaten	BYTE	B#16#0	
+3.0	Steuercode	CHAR		
+4.0	Temperatur	REAL	1.200000e+002	
+8.0	Ende	BOOL	FALSE	
=10.0		END STRUCT		

Nachdem Sie einen UDT erstellt haben, können Sie den UDT wie einen Datentyp verwenden, so z. B. wenn Sie für eine Variable den Datentyp *UDT200* in einem DB deklarieren (oder in der Variablendeklaration eines FB).

Nachfolgendes Bild zeigt einen DB mit der Variablen *Prozessdaten_1* mit dem Datentyp UDT200. Sie geben nur *UDT200* und *Prozessdaten_1* an. Die kursiv dargestellten Felder werden erstellt, wenn Sie den DB übersetzen.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+10.0	Prozessdaten 1	UDT200		
=10.0		END STRUCT		

Zuordnen von Anfangswerten für einen anwenderdefinierten Datentyp

Wenn Sie jedem Element eines UDT Anfangswerte zuordnen möchten, geben Sie für den Datentyp und den Namen eines jeden Elements einen gültigen Wert an. Sie können beispielsweise (dem im vorherigen Bild deklarierten UDT) die folgenden Anfangswerte zuordnen:

```
Menge           =      100
Ursprungsdaten  =      B#16#0
Steuercode      =      'Z'
Temperatur      =      1.200000e+002
Ende            =      False
```

Wenn Sie eine Variable als UDT deklarieren, dann sind die Anfangswerte der Variablen die Werte, die Sie beim Erstellen des UDT angegeben haben.

Speichern von und Zugreifen auf Daten in einem anwenderdefinierten Datentyp

Sie greifen auf die einzelnen Elemente eines UDT zu. Sie können symbolische Adressen (z. B. *Stapel_1.Temperatur*) verwenden. Sie können aber auch die absolute Adresse angeben, unter der das Element gespeichert ist (Beispiel: ist *Stapel_1* in DB20 mit Beginn bei Byte 0 gespeichert, dann ist die absolute Adresse für die *Menge* *DB20.DBW0* und die Adresse für die *Temperatur* ist *DB20.DBD6*).

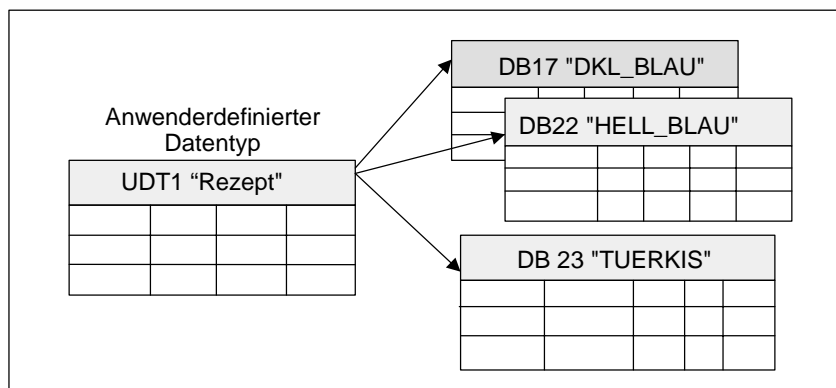
Verwenden von anwenderdefinierten Datentypen als Parameter

Sie können Variablen vom Datentyp UDT als Parameter übergeben. Wenn der Parameter in der Variablendeklaration als UDT deklariert ist, müssen Sie einen UDT übergeben, der die gleiche Struktur der Datenelemente hat. Ein Element eines UDT kann aber auch beim Aufrufen eines Bausteins einem Parameter zugeordnet werden, sofern das Element des UDT mit dem Datentyp des Parameters übereinstimmt.

Vorteile von DBs mit zugeordnetem UDT

Mit Hilfe von einmal erstellten UDTs können Sie eine Vielzahl von Datenbausteinen erzeugen, die die gleiche Datenstruktur haben. Diese Datenbausteine können von Ihnen durch Eingabe von unterschiedlichen Aktualwerten genau für die entsprechende Aufgabe angepasst werden.

Strukturieren Sie z. B. einen UDT für ein Rezept (z. B. für das Mischen von Farben), dann können Sie diesem UDT mehrere DBs zuordnen, die jeweils die anderen Mengenangaben beinhalten:



Die Struktur des Datenbausteins wird durch den zugeordneten UDT festgelegt.

A.3.4 Parametertypen

Zusätzlich zu elementaren und zusammengesetzten Datentypen können Sie Parametertypen für Formalparameter definieren, die zwischen Bausteinen übergeben werden. STEP 7 Lite kennt folgende Parametertypen:

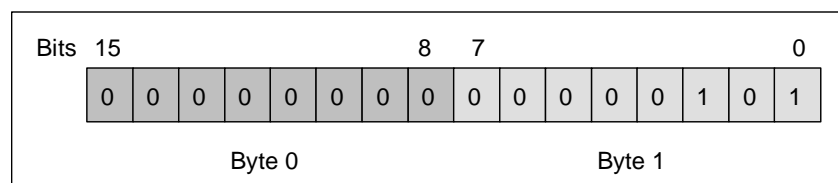
- **TIMER** oder **COUNTER**: legen eine bestimmte Zeit oder einen bestimmten Zähler fest, der bei der Bearbeitung verwendet werden soll. Wenn Sie einen Formalparameter vom Parametertyp **TIMER** oder **COUNTER** versorgen, muss der zugehörige Aktualparameter eine Zeit oder ein Zähler sein, d. h. Sie geben ein "T" oder "Z" an, auf das eine positive Ganzzahl folgt.
- **BLOCK**: legt einen bestimmten Baustein fest, der als Eingang oder als Ausgang verwendet werden soll. Die Deklaration des Parameters bestimmt die Bausteinart (FB, FC, DB usw.), die verwendet werden soll. Wenn Sie einen Formalparameter vom Parametertyp **BLOCK** versorgen, geben Sie eine Bausteinadresse als Aktualparameter an. Beispiel: "FC 101" (bei absoluter Adressierung) oder "Ventil" (bei symbolischer Adressierung).
- **POINTER**: referenziert die Adresse einer Variablen. Ein Pointer enthält eine Adresse anstelle eines Werts. Wenn Sie einen Formalparameter vom Parametertyp **POINTER** versorgen, geben Sie als Aktualparameter die Adresse an. In STEP 7 Lite können Sie einen Pointer im Pointerformat oder einfach als Adresse (z. B. M 50.0) angeben. Beispiel für ein Pointerformat zum Adressieren der Daten, die in M 50.0 beginnen: P#M50 . 0.
- **ANY**: wird verwendet, wenn der Datentyp des Aktualparameters nicht bekannt ist oder wenn ein beliebiger Datentyp eingesetzt werden kann. Nähere Informationen zum Parameter **ANY** entnehmen Sie den Abschnitten "Format des Parametertyps ANY bzw. Verwenden des Parametertyps ANY".

Ein Parametertyp kann auch ein anwenderdefinierter Datentyp (UDT) sein. Nähere Informationen zu UDTs entnehmen Sie dem Abschnitt "Verwenden von anwenderdefinierten Datentypen für den Datenzugriff".

Parameter	Größe	Beschreibung
TIMER	2 Bytes	Kennzeichnet eine bestimmte Zeit, die vom Programm im aufgerufenen Codebaustein verwendet werden soll. Format: T1
COUNTER	2 Bytes	Kennzeichnet einen bestimmten Zähler, der vom Programm im aufgerufenen Codebaustein verwendet werden soll. Format: Z10
BLOCK_FB BLOCK_FC BLOCK_DB BLOCK_SDB	2 Bytes	Kennzeichnet einen bestimmten Baustein, der vom Programm im aufgerufenen Codebaustein verwendet werden soll. Format: FC101 DB42
POINTER	6 Bytes	Kennzeichnet die Adresse. Format: P#M50.0
ANY	10 Bytes	Wird verwendet, wenn der Datentyp des Aktualparameters nicht bekannt ist. Format: P#M50.0 BYTE 10 ANY-Format bei Datentypen P#M100.0 WORD 5 L#1COUNTER 10 ANY-Format bei Parametertypen

A.3.4.1 Format der Parametertypen BLOCK, COUNTER und TIMER

STEP 7 Lite speichert die Parametertypen BLOCK, COUNTER und TIMER als Binärzahlen in einem Wort (32 Bits). Nachfolgendes Bild zeigt das Format dieser Parametertypen.



Die zulässige Anzahl der Bausteine, Zeiten und Zähler ist von Ihrer Variante der S7-CPU abhängig. Nähere Informationen zu der zulässigen Anzahl an Zeiten und Zählern sowie zu der maximalen Anzahl an verfügbaren Bausteinen entnehmen Sie den Datenblättern zu Ihrer CPU im Handbuch *Automatisierungssystem S7-300, Aufbau, CPU-Daten*.

Damit können Sie dem Codebaustein die Nummer eines Zeitglieds, Zählgliedes, Funktionsbausteins, Datenbausteins, Systemdatenbausteins oder einer Funktion übergeben, das er zur Bearbeitung verwenden soll. Bei der Versorgung dieses vereinbarten Formalparameters geben Sie als Aktualparameter ein "T", "Z", "FB", "DB", "SDB" oder "FC" an, auf das eine positive Ganzzahl folgt.

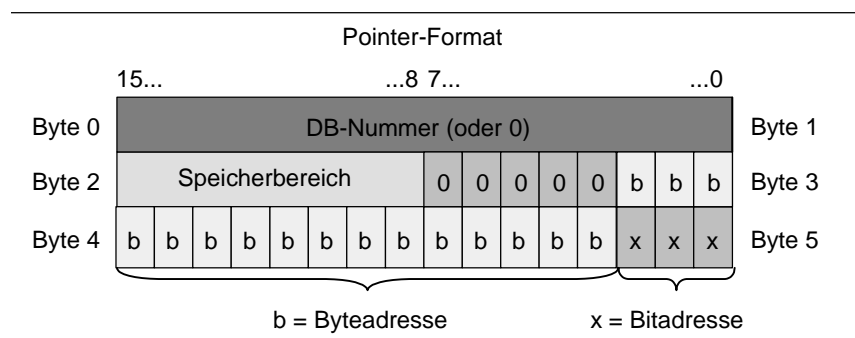
Beispiel:

```

Call FB 10, DB110 (
:
Eingangs_Var_Counter:= Z5, // FB 10 wird den
                        // Zähler 5 für die Bearbeitung
                        // verwenden.
:
);
    
```

A.3.4.2 Format des Paramertyps POINTER

STEP 7 Lite speichert den Paramertyp POINTER in 6 Bytes (48 Bits).
 Nachfolgendes Bild zeigt die Art der Daten, die in jedem Byte gespeichert werden.



Der Paramertyp POINTER speichert die folgenden Informationen:

- DB-Nummer (bzw. 0, wenn die Daten nicht in einem DB gespeichert werden)
- Speicherbereich in der CPU (Nachfolgende Tabelle zeigt die Hexadezimalcodes der Speicherbereiche für den Paramertyp POINTER)

Hexadezimalcode	Speicherbereich	Beschreibung
b#16#81	E	Speicherbereich der Eingänge
b#16#82	A	Speicherbereich der Ausgänge
b#16#83	M	Speicherbereich der Merker
b#16#84	DB	Datenbaustein
b#16#85	DI	Instanz-Datenbaustein
b#16#86	L	Lokaldaten (L-Stack)
b#16#87	V	Vorherige Lokaldaten

- Adresse der Daten (im Format Byte.Bit)

STEP 7 Lite bietet das Pointer-Format: p#Speicherbereich Byte.Bit_Adresse.
 (Wenn der Formalparameter als Paramertyp POINTER deklariert wurde, dann brauchen Sie nur den Speicherbereich und die Adresse anzugeben.
 STEP 7 Lite formatiert Ihren Eintrag dann automatisch in das Pointer-Format um.) Die folgenden Beispiele zeigen, wie Sie den Paramertyp POINTER für die Daten, die bei M 50.0 beginnen, eingeben:
- P#M50.0
- M50.0 (wenn der Formalparameter als POINTER deklariert wurde)

Beispiel:

```

Call FB 10, DB110 (
:
Eingangs_Var_Addr:= P#M20.0,    //FB 10 wird die Adresse des
    //Merkers 20.0 für die
    //Bearbeitung
    //verwenden.
:
);

```

A.3.4.3 Verwenden des Parametertyps POINTER

Ein Pointer wird verwendet, um auf einen Operanden zu zeigen. Der Vorteil dieser Adressierung ist, dass Sie den Operanden der Anweisung während der Programmbearbeitung dynamisch modifizieren können.

Pointer zur speicherindirekten Adressierung

Programmanweisungen, die mit der speicherindirekten Adressierung arbeiten, setzen sich zusammen aus einer Operation, einem Operandenkennzeichen und einem Versatz (der Versatz muss in eckigen Klammern angegeben werden).

Beispiel für einen Pointer im Doppelwortformat:

L	P#8.7	Lade den Wert des Pointers in AKKU 1.
T	MD2	Transferiere den Pointer nach MD2.
U	E [MD2]	Frage den Signalzustand an Eingang E 8.7 ab,
=	A [MD2]	und weise den Signalzustand Ausgang A 8.7 zu.

Pointer zur bereichsinternen und bereichsübergreifenden Adressierung

Die Programmanweisungen, die mit diesen Adressierungen arbeiten, setzen sich zusammen aus einer Operation und den folgenden Teilen:
Operandenkennzeichen, Adressregister-Kennung, Versatz.

Das Adressregister (AR1/2) und der Versatz müssen zusammen in eckigen Klammern angegeben werden.

Beispiel zur bereichsinternen Adressierung

Der Pointer enthält keine Angabe zu einem Speicherbereich:

L	P#8.7	Lade den Wert des Pointers in AKKU 1.
LAR1		Lade den Pointer von AKKU 1 in AR1.
U	E [AR1, P#0.0]	Frage den Signalzustand an Eingang E 8.7 ab und
=	A [AR1, P#1.1]	weise den Signalzustand dem Ausgang zu A 10.0 zu

Der Versatz 0.0 hat keinen Einfluss. Ausgang 10.0 errechnet sich aus 8.7 (AR1) plus den Versatz 1.1. Das Ergebnis ist 10.0 und nicht 9.8, siehe Pointer-Format.

Beispiel zur bereichsübergreifenden Adressierung

Bei der bereichsübergreifenden Adressierung ist im Pointer der Speicherbereich angegeben (im Beispiel E bzw. A).

L	P# E8.7	Lade den Wert des Pointers und Bereichskennung in AKKU 1.
LAR1		Lade Speicherbereich E und die Adresse 8.7 in AR1.
L	P# A8.7	Lade den Wert des Pointers und Bereichskennung in AKKU 1.
LAR2		Lade Speicherbereich A und die Adresse 8.7 in AR2.
U	[AR1, P#0.0]	Frage den Signalzustand an Eingang E 8.7 ab und
=	[AR2, P#1.1]	weise den Signalzustand Ausgang A 10.0 zu.

Der Versatz 0.0 hat keinen Einfluss. Ausgang 10.0 errechnet sich aus 8.7 (AR2) plus 1.1 (Versatz). Das Ergebnis ist 10.0 und nicht 9.8, siehe Pointer-Format.

A.3.4.4 Baustein zum Verändern des Pointers

Mit Hilfe des Beispiel-Bausteins FC 3 "Rangieren von Pointern" ist es möglich die Bit- oder Byteadresse eines Pointers zu verändern. Der zu verändernde Pointer wird beim FC-Aufruf an die Variable "Zeiger" übergeben (es können bereichsinterne und bereichsübergreifende Pointer im Doppelwortformat verwendet werden).

Mit dem Parameter "Bit-Byte" können Sie die Bit- oder die Byteadresse des Pointers ändern (0: Bitadresse, 1: Byteadresse). Bei der Variable "Inc_Wert" (im Format Integer) wird die Zahl angegeben, die zum Adressinhalt addiert bzw. subtrahiert werden soll. Hier können Sie auch negative Zahlen zum Dekrementieren der Adresse angeben.

Bei Bitadressenänderung erfolgt ein Übertrag in die Byteadresse (auch bei Dekrementierung); z. B.:

- P#M 5.3, Bit_Byte = 0, Inc_Wert = 6 => P#M 6.1 oder
- P#M 5.3, Bit_Byte = 0, Inc_Wert = -6 => P#M 4.5.

Die Bereichsinformation des Pointers wird durch die Funktion nicht beeinflusst.

Der FC fängt einen Über-/Unterlauf des Zeigers ab. In diesem Fall wird der Zeiger nicht verändert und die ausgangsseitige Variable "RET_VAL" (Fehlerbehandlung möglich) wird auf "1" gesetzt (bis zur nächsten korrekten Bearbeitung der FC 3). Dies ist der Fall, wenn:

- 1. Bitadresse gewählt ist und Inc_Wert >7, oder <-7
- 2. Bit- oder Byteadresse gewählt ist und die Änderung eine "negative" Byteadresse zur Folge hätte
- 3. Bit- oder Byteadresse gewählt ist und die Änderung eine unzulässig große Byteadresse zur Folge hätte

Beispiel-Baustein in AWL zum Verändern des Pointers:

```

FUNCTION FC 3: BOOL
TITLE =Rangieren von Pointern
//Die FC 3 kann zum Verändern von Pointern verwendet werden.
AUTHOR : AUT1CS1
FAMILY : INDADR
NAME : ADRPOINT
VERSION : 0.0

VAR_INPUT
    Bit_Byte : BOOL ; //0: Bitadresse, 1: Byteadresse
    Inc_Wert : INT ; //Inkrement (wenn Wert neg. =>
//Dekrement/wenn Wert pos. =>
//Inkrement)

END_VAR

VAR_IN_OUT
    Zeiger : DWORD ; //der zu veränderten Pointer
END_VAR
VAR_TEMP
    Inc_Wert1 : INT ; //Zwischenwert Inkrement
    Zeiger1 : DWORD ; //Zwischenwert Zeiger
    Zw_Wert : DWORD ; //Hilfsvariable
END_VAR
BEGIN
NETWORK
TITLE =
//Der Baustein fängt Veränderungen, welche die
//Bereichsinformationen des Zeigers
//verändern, oder zu "negativen" Pointern führen
//automatisch ab!
    SET ; //Setze VKE auf 1 und
    R #RET_VAL; //setze Überlauf zurück
    L #Zeiger; //Versorgen des temporären
    T #Zeiger1; //Zwischenwertes Zeiger
    L #Inc_Wert; //Versorgen des temporären
    T #Inc_Wert1; //Zwischenwertes Inkrement
    U #Bit_Byte; //wenn =1, dann
Byteadressenoperation
    SPB Byte; //Sprung zur Byteadressenberechnung
    L 7; //Wenn Wert des Inkrementes > 7,
    L #Inc_Wert1;
    <I ;
    S #RET_VAL; //dann setze RET_VAL und
    SPB Ende; //springe zum Ende
    L -7; //Wenn Wert des Inkrementes < -7,
    <I ;
    S #RET_VAL; //dann setze RET_VAL und
    SPB Ende; //springe zum Ende
    U L 1.3; //wenn Bit 4 des Wertes = 1
//Inc_Wert neg) ist,
    SPB neg; //dann springe zu Bitadressen-
//Subtraktion

```

```

    L      #Zeiger1;      //Lade Adressinformation des
Zeigers
    L      #Inc_Wert1;    //und addiere das Inkrement
    +D     ;
    SPA    test;         //Springe zum Test auf
                        //negatives Ergebnis
neg: L #Zeiger1;        //Lade Adressinformation
                        //des Zeigers
    L      #Inc_Wert1;    //Lade das Inkrement
    NEGI   ;             //negiere den negativen Wert,
    -D     ;             //subtrahiere den Wert
    SPA    test;         //und springe zum Test
Byte: L 0;              //Beginn der Byteadressenänderung
    L      #Inc_Wert1;    //Wenn Inkrement >= 0, dann
    <I     ;
    SPB    pos;         //springe zu Addition, sonst
    L      #Zeiger1;    //lade Adressinformation des
Zeigers,
    L      #Inc_Wert1;    //lade das Inkrement,
    NEGI   ;             //negiere den negativen Wert,
    SLD    3;           //schiebe das Inkrement 3 Stellen
                        //nach links,
    -D     ;             //subtrahiere den Wert
    SPA    test;         //und springe zum Test
pos: SLD 3;            //schiebe Inkrement 3 Stellen
                        //nach links
    L      #Zeiger1;    //lade Adressinformation des
Zeigers
    +D     ;             //addiere Inkrement
test: T #Zw_Wert;      //Transfer. Ergebnis Berechnungen
                        //in Zw_Wert,
    U      L 7.3;        //Wenn ungültige Byteadresse
                        //(zu groß oder
    S      #RET_VAL;    //negativ), dann setze RET_VAL
    SPB    Ende;        //und springe zum Ende,
    L      #Zw_Wert;    //sonst transferiere Ergebnis
    T      #Zeiger;     //in Zeiger
Ende: NOP 0;
END_FUNCTION
```

A.3.4.5 Format des Paramertyps ANY

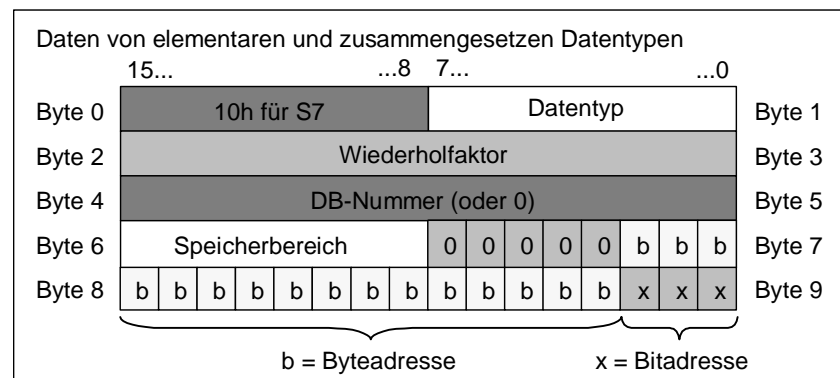
STEP 7 Lite speichert die Daten des Paramertyps ANY in 10 Bytes. Beim Aufbau eines Parameters vom Typ ANY müssen Sie darauf achten, dass alle 10 Bytes belegt sind, da der aufgerufene Baustein den gesamten Inhalt des Parameters auswertet. Wenn Sie beispielsweise in Byte 4 eine DB-Nummer spezifizieren, müssen Sie auch den Speicherbereich in Byte 6 explizit angeben.

STEP 7 Lite verwaltet die Daten von elementaren und zusammengesetzten Datentypen anders als die Daten für Paramertypen.

ANY-Format bei Datentypen

Bei elementaren und zusammengesetzten Datentypen speichert STEP 7 Lite die folgenden Daten:

- Datentypen
- Wiederholfaktor
- DB-Nummer
- Speicherbereich, in dem die Informationen gespeichert werden
- Anfangsadresse der Daten



Der Wiederholfaktor kennzeichnet eine Menge des gekennzeichneten Datentyps zum Übergeben durch den Paramertyp ANY. Dadurch können Sie einen Datenbereich angeben und auch Felder und Strukturen in Verbindung mit dem Paramertyp ANY verwenden. STEP 7 Lite kennzeichnet Felder und Strukturen als Anzahl (mit Hilfe des Wiederholfaktors) an Datentypen. Sollen z. B. 10 Wörter übergeben werden, muss beim Wiederholfaktor der Wert 10 und beim Datentyp der Wert 04 eingetragen sein.

Die Adresse wird im Format Byte.Bit gespeichert, wobei die Byteadresse in den Bits 0 bis 2 von Byte 7, in den Bits 0 bis 7 von Byte 8 und in den Bits 3 bis 7 von Byte 9 gespeichert wird. Die Bitadresse wird in den Bits 0 bis 2 von Byte 9 gespeichert.

Beim Nullpointer vom Datentyp NIL sind ab Byte 1 alle Bytes mit 0 belegt.

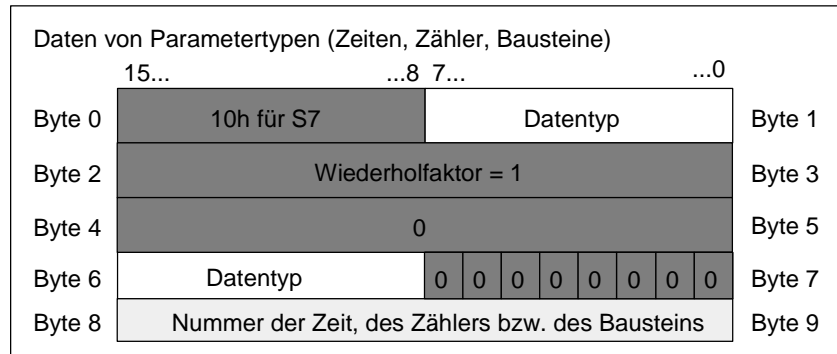
Nachfolgenden Tabellen zeigen die Codierung der Datentypen bzw. der Speicherbereiche für den Paramertyp ANY.

Codierung der Datentypen		
Hexadezimalcode	Datentyp	Beschreibung
b#16#00	NIL	Nullpointer
b#16#01	BOOL	Bits
b#16#02	BYTE	Bytes (8 Bits)
b#16#03	CHAR	Zeichen (8 Bits)
b#16#04	WORD	Wörter (16 Bits)
b#16#05	INT	Ganzzahlen (16 Bits)
b#16#06	DWORD	Wörter (32 Bits)
b#16#07	DINT	Ganzzahlen (32 Bits)
b#16#08	REAL	Gleitpunktzahlen (32 Bits)
b#16#09	DATE	Datum
b#16#0A	TIME_OF_DAY (TOD)	Uhrzeit
b#16#0B	TIME	Zeit
b#16#0C	S5TIME	Datentyp S5TIME
b#16#0E	DATE_AND_TIME (DT)	Datum und Zeit (64 Bits)
b#16#13	STRING	Zeichenkette

Codierung der Speicherbereiche		
Hexadezimalcode	Bereich	Beschreibung
b#16#81	E	Speicherbereich der Eingänge
b#16#82	A	Speicherbereich der Ausgänge
b#16#83	M	Speicherbereich der Merker
b#16#84	DB	Datenbaustein
b#16#85	DI	Instanz-Datenbaustein
b#16#86	L	Lokaldaten (L-Stack)
b#16#87	V	Vorherige Lokaldaten

ANY-Format bei Parametertypen

Bei Parametertypen speichert STEP 7 Lite den Datentyp und die Adresse der Parameter. Der Wiederholfaktor ist immer 1. Byte 4, 5 und 7 sind immer 0. Byte 8 und 9 geben die Nummer der Zeit, des Zählers oder des Bausteins an.



Nachfolgende Tabelle zeigt die Codierung der Datentypen für den Parametertyp ANY bei Parametertypen.

Hexadezimalcode	Datentyp	Beschreibung
b#16#17	BLOCK_FB	FB-Nummer
b#16#18	BLOCK_FC	FC-Nummer
b#16#19	BLOCK_DB	DB-Nummer
b#16#1A	BLOCK_SDB	SDB-Nummer
b#16#1C	COUNTER	Nummer des Zählers
b#16#1D	TIMER	Nummer der Zeit

Beispiel:

```

Call FB 10, DB110(
:
:
Eingangs_Var_Any:= MW100, // Hier wird FB 10 ein Wort
(MW100)
// für die Bearbeitung
// verwenden.
:
);
:
Call FB 10, DB110(
:
:
Eingangs_Var_Any:= M1.3, // Hier wird FB 10 ein Bit (M1.3)
// für die Bearbeitung
// verwenden.
:
);

```

A.3.4.6 Verwenden des Parametertyps ANY

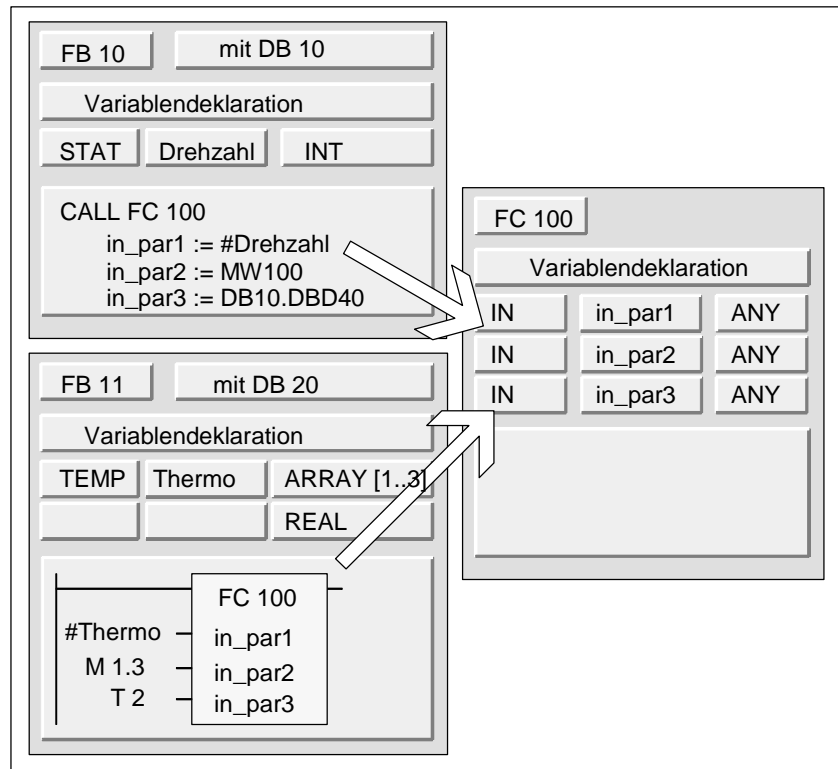
Sie können für einen Baustein Formalparameter definieren, die für Aktualparameter mit beliebigen Datentypen geeignet sind. Dies ist vor allem dann nützlich, wenn der Datentyp des Aktualparameters, der beim Aufrufen des Bausteins bereitgestellt wird, unbekannt ist oder variieren kann (und wenn ein beliebiger Datentyp zulässig ist). In der Variablendeklaration des Bausteins deklarieren Sie den Parameter als Datentyp ANY. In STEP 7 Lite können Sie dann einen Aktualparameter eines beliebigen Datentyps zuordnen.

STEP 7 Lite ordnet einer Variablen vom Datentyp ANY 80 Bits Speicherplatz zu. Wenn Sie diesem Formalparameter einen Aktualparameter zuordnen, codiert STEP 7 Lite die Startadresse, den Datentyp und die Länge des Aktualparameters in den 80 Bits. Der aufgerufene Baustein analysiert die 80 Bits an Daten, die für den Parameter ANY gespeichert wurden, und erhält so Informationen, die für die zusätzliche Bearbeitung verwendet werden können.

Zuordnen eines Aktualparameters zu einem Parameter ANY

Wenn Sie für einen Parameter den Datentyp ANY deklarieren, können Sie diesem Formalparameter einen Aktualparameter mit beliebigem Datentyp zuordnen. In STEP 7 Lite können Sie die folgenden Datentypen als Aktualparameter zuordnen:

- Elementare Datentypen: Sie geben die absolute Adresse oder den symbolischen Namen des Aktualparameters an.
- Zusammengesetzte Datentypen: Sie geben den symbolischen Namen der Daten mit zusammengesetztem Datentyp an (z. B. Felder und Strukturen).
- Zeiten, Zähler und Bausteine: Sie geben die Nummer an (z. B. T1, Z20 oder FB6).
- Nachfolgendes Bild zeigt, wie Daten an eine FC mit Parametern vom Datentyp ANY übergeben werden können.



In diesem Beispiel hat FC 100 drei Parameter (*in_par1*, *in_par2* und *in_par3*), die als Datentyp ANY deklariert wurden.

- Wenn FB 10 die FC 100 aufruft, übergibt FB10 eine Ganzzahl (die statische Variable "Drehzahl"), ein Wort (MW 100) und ein Doppelwort in DB10 (DB10.DBD40).
- Wenn FB 11 die FC 10 aufruft, übergibt FB11 ein Feld aus Realzahlen (die temporäre Variable "Thermo"), einen booleschen Wert (M 1.3) und eine Zeit (T2).

Angeben eines Datenbereichs für einen Parameter ANY

Sie können einem Parameter ANY aber nicht nur individuelle Operanden (z. B. MW100) zuordnen, sondern Sie können auch einen Datenbereich angeben. Wenn Sie einen Datenbereich als Aktualparameter zuordnen möchten, verwenden Sie das folgende Format einer Konstanten, um die Datenmenge anzugeben, die übergeben werden soll:

p# Bereichskennung Byte.Bit Datentyp Wiederholfaktor

Sie können für das Element *Datentyp* in dem Format für Konstanten alle elementaren Datentypen und den Datentyp DATE_AND_TIME angeben. Handelt es sich bei dem Datentyp nicht um BOOL, dann muss die Bitadresse von 0 (x.0) angegeben werden. Nachfolgende Tabelle zeigt Beispiele für das Format zum Angeben von Speicherbereichen, die an einen Parameter ANY übergeben werden sollen.

Aktualparameter	Beschreibung
p# M 50.0 BYTE 10	Gibt 10 Bytes im Speicherbereich Merker an: MB50 bis MB59.
p# DB10.DBX5.0 S5TIME 3	Gibt 3 Einheiten von Daten vom Datentyp S5TIME an, die in DB10 gespeichert sind: DB Byte 5 bis DB Byte 10.
p# A 10.0 BOOL 4	Gibt 4 Bits im Speicherbereich Ausgänge an: A 10.0 bis A 10.3.

Beispiel für die Verwendung des Parametertyps ANY

Das folgende Beispiel zeigt, wie Sie mit Hilfe des Parametertyps ANY und der Systemfunktion SFC 20 BLKMOV einen Speicherbereich von 10 Bytes kopieren können.

AWL	Erläuterung
FUNCTION FC 10:VOID	
VAR_TEMP	
Quelle : ANY;	
Ziel : ANY;	
END_VAR	
BEGIN	
LAR1 P#Quelle;	//Lade die Anfangsadresse des ANY-Pointers in AR1.
L B#16#10;	Lade die Syntax-ID und
T LB[AR1,P#0.0];	transferiere sie in den ANY-Pointer.
L B#16#02;	Lade Datentyp Byte und
T LB[AR1,P#1.0];	transferiere ihn in den ANY-Pointer.
L 10;	Lade 10 Byte und
T LW[AR1,P#2.0];	transferiere sie in den ANY-Pointer.
L 22;	Quelle ist der DB22, DBB11
T LW[AR1,P#4.0];	
L P#DBX11.0;	
T LD[AR1,P#6.0];	
LAR1 P#Ziel;	Lade die Anfangsadresse des ANY-Pointers in AR1.
L B#16#10;	Lade die Syntax-ID und
T LB[AR1,P#0.0];	transferiere sie in den ANY-Pointer.

AWL	Erläuterung
L B#16#02; T LB[AR1,P#1.0];	Lade Datentyp Byte und transferiere ihn in den ANY-Pointer.
L 10; T LW[AR1,P#2.0];	Lade 10 Bytes und transferiere sie in den ANY-Pointer.
L 33; T LW[AR1,P#4.0];	Ziel ist der DB33, DBB202
L P#DBX202.0; T LD[AR1,P#6.0];	
CALL SFC 20 (SRCBLK := Quelle, RET_VAL := MW 12, DSTBLK := Ziel); END_FUNCTION	Aufruf der Systemfunktion Blockmove Auswertung des BIE-Bit und des MW 12

A.3.4.7 Zuordnen von Datentypen zu Lokaldaten von Codebausteinen

STEP 7 Lite begrenzt die Datentypen (elementare und zusammengesetzte Datentypen sowie Parametertypen), die in der Variablendeklaration den Lokaldaten eines Bausteins zugeordnet werden können.

Gültige Datentypen für die Lokaldaten eines OB

Nachfolgende Tabelle zeigt die Einschränkungen (–) beim Deklarieren von Lokaldaten für einen OB. Weil Sie einen OB nicht aufrufen können, kann ein OB auch nicht über Parameter (Ein-, Aus- oder Durchgänge) verfügen. Weil ein OB keinen Instanz-DB hat, können Sie für einen OB keine statischen Variablen deklarieren. Bei den Datentypen der temporären Variablen eines OB kann es sich um elementare oder zusammengesetzte Datentypen sowie um ANY handeln.

Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Deklarations- typ	Elementare Datentypen	Zusammen- gesetzte Datentypen	Parame- tertyp	Parame- tertyp	Parame- tertyp	Parame- tertyp	Parame- tertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang	—	—	—	—	—	—	—
Ausgang	—	—	—	—	—	—	—
Durchgang	—	—	—	—	—	—	—
Statisch	—	—	—	—	—	—	—
Temporär	●(1)	●(1)	—	—	—	—	●(1)

⁽¹⁾ Im L-Stack des OB gespeichert.

Gültige Datentypen für die Lokaldaten eines FB

Nachfolgende Tabelle zeigt die Einschränkungen (–) beim Deklarieren von Lokaldaten für einen FB. Wegen des Instanz-DB gibt es weniger Einschränkungen beim Deklarieren von Lokaldaten für einen FC. Für die Deklaration von Eingangsparametern gibt es keinerlei Einschränkungen, für einen Ausgangsparameter können Sie keine Parametertypen deklarieren und für Durchgangparameter sind nur die Parametertypen POINTER und ANY zulässig. Temporäre Variablen können Sie als Datentyp ANY deklarieren. Alle anderen Parametertypen sind unzulässig.

Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Para-	Para-	Para-	Para-	Para-
			meter-	meter-	meter-	meter-	meter-
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang	●	●	●	●	●	●	●
Ausgang	●	●	—	—	—	—	—
Durchgang	●	●(1)(3)	—	—	—	●	●
Statisch	●	●	—	—	—	—	—
Temporär	●(2)	●(2)	—	—	—	—	●(2)

⁽¹⁾ Als Verweis (48-Bit-Pointer) im Instanz-DB gespeichert.

⁽²⁾ Im L-Stack des FB gespeichert.

⁽³⁾ STRINGS können nur in der Standardlänge definiert werden.

Gültige Datentypen für die Lokaldaten einer FC

Nachfolgende Tabelle zeigt die Einschränkungen (–) beim Deklarieren von Lokaldaten für eine FC. Weil eine FC keinen Instanz-DB hat, verfügt sie auch nicht über statische Variablen. Für Eingangs-, Ausgangs- und Durchgangparameter einer FC sind die Parametertypen POINTER und ANY zulässig. Sie können auch temporäre Variablen vom Parametertyp ANY deklarieren.

Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Para-	Para-	Para-	Para-	Para-
			meter-	meter-	meter-	meter-	meter-
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang	●	●(2)	●	●	●	●	●
Ausgang	●	●(2)	—	—	—	●	●
Durchgang	●	●(2)	—	—	—	●	●
Temporär	●(1)	●(1)	—	—	—	—	●(1)

⁽¹⁾ Im L-Stack der FC gespeichert.

⁽²⁾ STRINGS können nur in der Standardlänge definiert werden.

A.3.4.8 Zulässige Datentypen beim Übergeben von Parametern

Regeln für die Parameterübergabe zwischen Bausteinen

Wenn Sie Formalparametern Aktualparameter zuordnen, dann können Sie entweder eine absolute Adresse, einen symbolischen Namen oder eine Konstante angeben. STEP 7 Lite begrenzt die jeweils gültigen Zuordnungen bei den verschiedenen Parametern. Ausgangs- und Durchgangparameter können zum Beispiel keinem konstanten Wert zugeordnet werden (weil der Zweck eines Ausgangs bzw. eines Durchgangs das Ändern des Werts ist). Diese Einschränkungen gelten besonders bei Parametern mit zusammengesetzten Datentypen, denen weder eine absolute Adresse noch eine Konstante zugeordnet werden kann.

Nachfolgende Tabellen zeigen die Einschränkungen (–) bei den Datentypen von Aktualparametern, die Formalparametern zugeordnet werden.

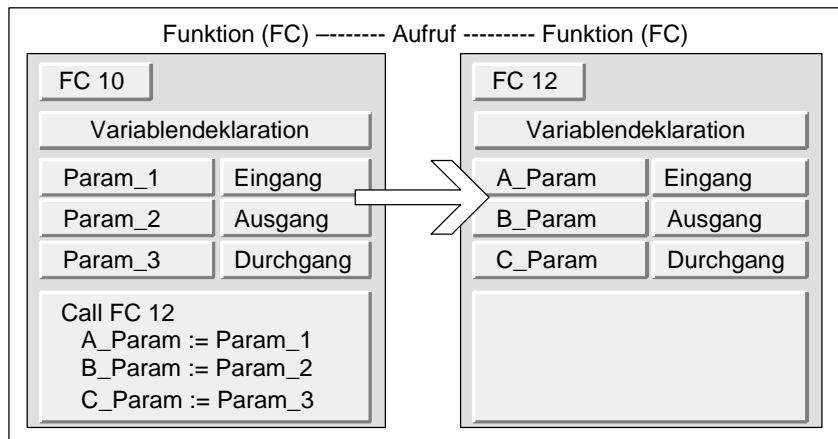
Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Elementare Datentypen				
Deklara- tionstyp	Absolute Adresse	Symbolischer Name (in der Symboltabelle)	Bausteinlokales Symbol	Konstante
Eingang	●	●	●	●
Ausgang	●	●	●	—
Durchgang	●	●	●	—

Zusammengesetzte Datentypen				
Deklara- tionstyp	Absolute Adresse	Symbolischer Name des Elements des DB (in der Symboltabelle)	Bausteinlokales Symbol	Konstante
Eingang	—	●	●	—
Ausgang	—	●	●	—
Durchgang	—	●	●	—

Zulässige Datentypen beim Aufruf einer FC durch eine andere FC

Sie können die Formalparameter einer aufrufenden FC den Formalparametern einer aufgerufenen FC zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FC 10, die als Aktualparameter den Formalparametern von FC 12 zugeordnet werden.



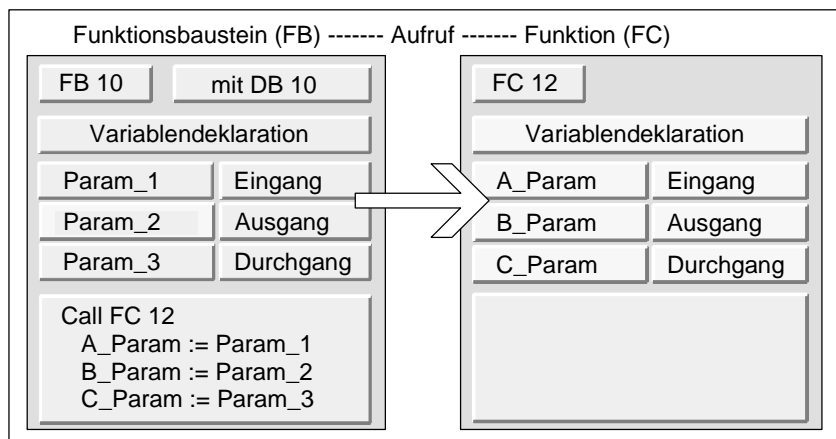
STEP 7 Lite begrenzt die Zuordnung von Formalparametern einer FC als Aktualparameter zu den Formalparametern einer anderen FC. Sie können beispielsweise keine Parameter mit zusammengesetztem Datentyp oder Parametertyp als Aktualparameter zuordnen.

Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn eine FC eine andere FC aufruft.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang → Eingang	●	—	—	—	—	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	—	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

Zulässige Datentypen beim Aufruf einer FC durch einen FB

Sie können die Formalparameter eines aufrufenden FB den Formalparametern einer aufgerufenen FC zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FB 10, die als Aktualparameter den Formalparametern von FC 12 zugeordnet werden.

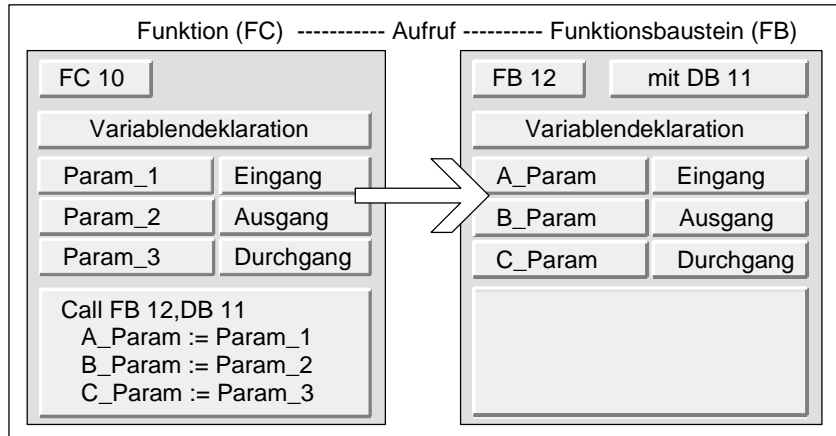


STEP 7 Lite begrenzt die Zuordnung der Formalparameter eines FB zu den Formalparametern einer FC. Sie können beispielsweise keine Parameter mit Parametertyp als Aktualparameter zuordnen. Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn ein FB eine FC aufruft.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parameter-typ	Parame-tertyp	Parame-tertyp	Parame-tertyp	Parame-tertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang → Eingang	●	●	—	—	—	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	●	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

Zulässige Datentypen beim Aufruf eines FB durch eine FC

Sie können die Formalparameter einer aufrufenden FC den Formalparametern eines aufgerufenen FB zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FC 10, die als Aktualparameter den Formalparametern von FB 12 zugeordnet werden.



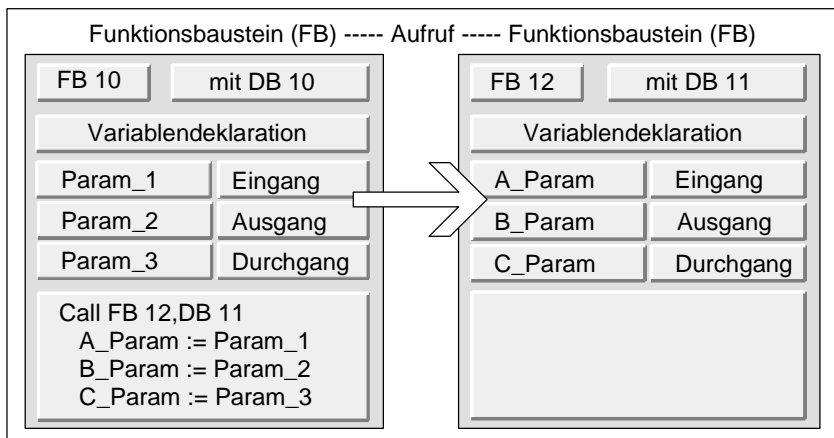
STEP 7 Lite begrenzt die Zuordnung von Formalparametern einer FC zu den Formalparametern eines FB. Sie können beispielsweise Parameter mit zusammengesetztem Datentyp nicht als Aktualparameter zuordnen. Sie können aber Eingangsparameter mit dem Parametertyp TIMER, COUNTER oder BLOCK den Eingangsparametern des aufgerufenen FB zuordnen.

Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn eine FC einen FB aufruft.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang → Eingang	●	—	●	●	●	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	—	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

Zulässige Datentypen beim Aufruf eines FB durch einen anderen FB

Sie können die Formalparameter eines aufrufenden FB den Formalparametern des aufgerufenen FB zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FB 10, die als Aktualparameter den Formalparametern von FB 12 zugeordnet werden.



STEP 7 Lite begrenzt die Zuordnung von Formalparametern eines FB zu den Formalparametern eines anderen FB. Sie können beispielsweise keine Eingangs- und Ausgangsparameter mit zusammengesetztem Datentyp als Aktualparameter den Eingangs- und Ausgangsparametern eines aufgerufenen FB zuordnen. Sie können aber Eingangsparameter mit dem Parametertyp TIMER, COUNTER oder BLOCK den Eingangsparametern des aufgerufenen FB zuordnen.

Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn ein FB einen anderen FB aufruft.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang → Eingang	●	●	●	●	●	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	●	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

A.3.4.9 Übergabe an In_out-Parameter eines FB

Bei der Übergabe von zusammengesetzten Datentypen an In_out-Parameter eines Funktionsbausteins (FB) wird die Operandenadresse der Variable übergeben (call by reference).

Bei der Übergabe von elementaren Datentypen an In_out-Parameter eines FB werden die Werte in den Instanz-Datenbaustein kopiert, bevor der FB gestartet wird, und aus dem Instanz-Datenbaustein herauskopiert, nachdem der FB beendet ist.

Dadurch können In_out-Variablen elementaren Datentyps mit einem Wert initialisiert werden.

Es ist jedoch nicht möglich, in einem Aufruf eine Konstante als Aktualparameter an der Position einer In_out-Variablen anzugeben, da auf eine Konstante nicht geschrieben werden kann.

Variablen vom Datentyp STRUCT oder ARRAY können nicht initialisiert werden, da in diesem Fall im Instanz-Datenbaustein nur eine Adresse liegt.

A.4 Beispielprogramme

A.4.1 Beispielprojekte und Beispielprogramme

Die Installations-CD enthält eine Anzahl von Beispielprojekten. Bei den Projekten, die nicht in diesem Kapitel beschrieben werden, ist eine Beschreibung im entsprechenden OB 1 enthalten.

Beispiele und Beispielprojekte	Auf CD enthalten	In diesem Kapitel beschrieben
Projekte "Erste_Schritte_KOP", "Erste_Schritte_FUP" und "Erste_Schritte_AWL"	•	Eigenes Handbuch "Erste Schritte mit STEP 7 Lite"
Beispiel für einen industriellen Mischprozess		•
Beispiel zur Hantierung mit Uhrzeitalarmen		•
Beispiel zur Hantierung mit Verzögerungsalarmen		•

Der Schwerpunkt der Beispiele liegt nicht auf der Vermittlung eines bestimmten Programmierstils oder technischen Fachwissens, das Sie zur Steuerung eines bestimmten Prozesses benötigen. Mit Hilfe dieser Beispiele können Sie vielmehr die Schritte nachvollziehen, die beim Programmwurf ausgeführt werden müssen.

Löschen und Installieren von mitgelieferten Beispielprojekten

Die mitgelieferten Beispielprojekte können Sie löschen und danach wieder neu installieren. Für die Installation starten Sie das Programm Setup von STEP 7 Lite. Die Beispielprojekte lassen sich selektiv nachinstallieren.

Sie finden die installierten Beispielprojekte auf dem Laufwerk, auf dem Sie STEP 7 Lite installiert haben unter Laufwerk:\Siemens\S7lite\Examples.

Achtung

Die mitgelieferten Beispielprojekte werden, falls diese nicht abgewählt sind, bei einer STEP 7 Lite-Installation kopiert. Wenn Sie mitgelieferte Beispielprojekte bearbeitet haben, werden bei einer erneuten STEP 7 Lite-Installation die von Ihnen geänderten Beispielprojekte mit dem Original überschrieben.

Deswegen sollten Sie mitgelieferte Beispielprojekte vor Änderungen kopieren und nur die Kopie bearbeiten.

A.4.2 Beispiel für das Maskieren und Demaskieren von Synchronfehlerereignissen

Im folgenden Beispiel für ein Anwenderprogramm zeigen wir Ihnen das Maskieren und Demaskieren von Synchronfehlerereignissen. Mit dem SFC 36 "MSK_FLT" werden in der Programmierfehlermaske folgende Fehler maskiert:

- Bereichslängenfehler beim Lesen
- Bereichslängenfehler beim Schreiben

Mit einem 2. Aufruf des SFC 36 "MSK_FLT" wird zusätzlich ein Zugriffsfehler maskiert:

- Peripheriezugriffsfehler beim Schreiben

Mit dem SFC 38 "READ_ERR" werden die maskierten Synchronfehlerereignisse abgefragt. Der "Peripheriezugriffsfehler beim Schreiben" wird mit dem SFC 37 "DMSK_FLT" wieder demaskiert.

Anweisung

Im folgenden sehen Sie den OB 1, in dem das Beispiel für das Anwenderprogramm in AWL programmiert wurde.

AWL (Network 1)	Erläuterung
UN M 255.0	//nicht remanenter Merker M 255.0 (nur beim //ersten Durchlauf=0)
SPENB m001	
CALL SFC 36	//SFC 36 MSK_FLT (Maskieren von Synchron- //fehlerereignissen)
PRGFLT_SET_MASK :=DW#16#C	//Bit2=Bit3=1 (BLFL und BLFS werden maskiert)
ACCFLT_SET_MASK :=DW#16#0	//alle Bits=0 (es wird kein Zugriffs//fehler //maskiert)
RET_VAL :=MW 100	//Returnwert
PRGFLT_MASKED :=MD 10	//Ausgabe der aktuellen Programmierfehlermaske //in MD 10
ACCFLT_MASKED :=MD 14	//Ausgabe der aktuellen Zugriffsfehlermaske in //MD 14
m001: U BIE	//Setzen von M255.0, wenn Maskieren
S M 255.0	//erfolgreich
AWL (Network 2)	Erläuterung
CALL SFC 36	//SFC 36 MSK_FLT (Maskieren von Synchron- //fehlerereignissen)
PRGFLT_SET_MASK :=DW#16#0	//alle Bits=0 (es wird kein weiterer //Programmfehler maskiert)
ACCFLT_SET_MASK :=DW#16#8	//Bit3=1 (schreibende Zugriffsfehler werden //maskiert)
RET_VAL :=MW 102	//Returnwert
PRGFLT_MASKED :=MD 20	//Ausgabe der aktuellen Programmier- //fehlermaske in MD 20
ACCFLT_MASKED :=MD 24	//Ausgabe der aktuellen Zugriffsfehlermaske in //MD 24
AWL (Network 3)	Erläuterung
UN M 27.3	//Baustein-Ende, wenn schreibender
BEB	//Zugriffsfehler (Bit3 in ACCFLT_MASKED) nicht //maskiert
AWL (Network 4)	Erläuterung
L B#16#0	
T PAB 16	//Schreibender Zugriff (mit Wert 0) auf PAB 16
AWL (Network 5)	Erläuterung
CALL SFC 38	//SFC 38 READ_ERR (Abfragen von //Synchronfehlerereignissen)
PRGFLT_QUERY :=DW#16#0	//alle Bits=0 (es wird kein Programmierfehler //abgefragt)
ACCFLT_QUERY :=DW#16#8	//Bit3=1 (schreibender Zugriffsfehler wird //abgefragt)
RET_VAL :=MW 104	//Returnwert
PRGFLT_CLR :=MD 30	//Ausgabe der aktuellen Programmierfehlermaske //in MD 30
ACCFLT_CLR :=MD 34	//Ausgabe der aktuellen Zugriffsfehlermaske in //MD 34
U BIE	//kein Fehler aufgetreten und schreibenden //Zugriffsfehler festgestellt
U M 37.3	//VKE invertieren
NOT	//M 0.0=1, wenn PAB 16 vorhanden
= M 0.0	

AWL (Network 6)		Erläuterung
L	B#16#0	
T	PAB 17	Schreibender Zugriff (mit Wert 0) auf PAB 17

AWL (Network 7)		Erläuterung
CALL	SFC 38	//SFC 38 READ_ERR (Abfragen von Synchron- //fehlerereignissen)
PRGFLT_QUERY	:=DW#16#0	//alle Bits=0 (es wird kein Programmierfehler //abgefragt)
ACCFLT_QUERY	:=DW#16#8	//Bit3=1 (schreibender Zugriffsfehler wird //abgefragt)
RET_VAL	:=MW 104	//Returnwert
PRGFLT_CLR	:=MD 30	//Ausgabe der aktuellen Programmierfehlermaske //in MD 30
ACCFLT_CLR	:=MD 34	//Ausgabe der aktuellen Zugriffsfehlermaske in //MD 34
U	BIE	//kein Fehler aufgetreten und schreibenden //Zugriffsfehler festgestellt
U	M 37.3	//VKE invertieren
NOT		//M 0.1=1, wenn PAB 17 vorhanden
=	M 0.1	

AWL (Network 8)		Erläuterung
L	B#16#0	
T	PAB 18	//Schreibender Zugriff (mit Wert 0) auf PAB 18

AWL (Network 9)		Erläuterung
CALL	SFC 38	//SFC 38 READ_ERR (Abfragen von //Synchronfehlerereignissen)
PRGFLT_QUERY	:=DW#16#0	//alle Bits=0 (es wird kein Programmierfehler //abgefragt)
ACCFLT_QUERY	:=DW#16#8	//Bit3=1 (schreibender Zugriffsfehler wird //abgefragt)
RET_VAL	:=MW 104	//Returnwert
PRGFLT_CLR	:=MD 30	//Ausgabe der aktuellen Programmierfehlermaske //in MD 30
ACCFLT_CLR	:=MD 34	//Ausgabe der aktuellen Zugriffsfehlermaske in //MD 34
U	BIE	//kein Fehler aufgetreten und schreibenden //Zugriffsfehler festgestellt
U	M 37.3	//VKE invertieren
NOT		//M 0.2=1, wenn PAB 18 vorhanden
=	M 0.2	

AWL (Network 10)		Erläuterung
L	B#16#0	
T	PAB 19	//Schreibender Zugriff (mit Wert 0) auf PAB 19

AWL (Network 11)		Erläuterung
CALL	SFC 38	//SFC 38 READ_ERR (Abfragen von Synchron- //fehlerereignissen)
PRGFLT_QUERY	:=DW#16#0	//alle Bits=0 (es wird kein Programmierfehler //abgefragt)
ACCFLT_QUERY	:=DW#16#8	//Bit3=1 (schreibender Zugriffsfehler wird //abgefragt)
RET_VAL	:=MW 104	//Returnwert
PRGFLT_CLR	:=MD 30	//Ausgabe der aktuellen Programmierfehlermaske //in MD 30
ACCFLT_CLR	:=MD 34	//Ausgabe der aktuellen Zugriffsfehlermaske in //MD 34
U	BIE	//kein Fehler aufgetreten und schreibenden //Zugriffsfehler festgestellt
U	M 37.3	//VKE invertieren
NOT		//M 0.3=1, wenn PAB 19 vorhanden
=	M 0.3	

AWL (Network 12)		Erläuterung
CALL	SFC 37	//SFC 37 DMSK_FLT (Demaskieren von
		//Synchronfehlerereignissen)
PRGFLT_RESET_MASK	:=DW#16#0	//alle Bits=0 (es wird kein
		//Programmierfehler demaskiert)
ACCFLT_RESET_MASK	:=DW#16#8	//Bit3=1 (schreibender Zugriffsfehler
		//wird demaskiert)
RET_VAL	:=MW 102	//Returnwert
PRGFLT_MASKED	:=MD 20	//Ausgabe der aktuellen Programmier-
		//fehlermaske in MD 20
ACCFLT_MASKED	:=MD 24	//Ausgabe der aktuellen
		//Zugriffsfehlermaske in MD 24
AWL (Network 13)		Erläuterung
U	M 27.3	//Baustein-Ende, wenn schreibender
BEB		//Zugriffsfehler (Bit3 in ACCFLT MASKED)
		//nicht demaskiert
AWL (Network 14)		Erläuterung
U	M 0.0	
SPBNB	m002	
L	EB 0	//EB 0 nach PAB 16 transferieren, wenn
T	PAB 16	//vorhanden
m002: NOP	0	
AWL (Network 15)		Erläuterung
U	M 0.1	
SPBNB	m003	
L	EB 1	//EB 1 nach PAB 17 transferieren, wenn
T	PAB 17	//vorhanden
m003: NOP	0	
AWL (Network 16)		Erläuterung
U	M 0.2	
SPBNB	m004	
L	EB 2	//EB 2 nach PAB 18 transferieren, wenn
T	PAB 18	//vorhanden
m004: NOP	0	
AWL (Network 17)		Erläuterung
U	M 0.3	
SPBNB	m005	
L	EB 3	//EB 3 nach PAB 19 transferieren, wenn
T	PAB 19	//vorhanden
m005: NOP	0	

A.4.3 Beispiel zum Sperren und Freigeben von Alarm- und Asynchronereignissen (SFC 39 und 40)

In diesem Beispiel für ein Anwenderprogramm wird ein Programmteil angenommen, der nicht durch Alarme unterbrochen werden darf. Für diesen Programmteil werden mit dem SFC 39 "DIS_IRT" OB 35-Aufrufe (Uhrzeitalarm) gesperrt und mit dem SFC 40 "EN_IRT" werden die OB 35-Aufrufe wieder freigegeben.

Im OB 1 werden der SFC 39 und der SFC 40 aufgerufen:

AWL (OB 1)	Erläuterung
U M 0.0	//Programmteil, der problemlos unterbrochen
S M 90.1	//werden kann:
U M 0.1	
S M 90.0	
:	
:	//Programmteil, der nicht durch Interrupts
	//unterbrochen werden darf:
CALL SFC 39	//Alarme sperren und verwerfen
MODE :=B#16#2	//Mode 2: einzelne Alarm-OBs sperren
OB_NR :=35	//OB 35 sperren
RET_VAL :=MW 100	
:	
:	
L PEW 100	
T MW 200	
L MW 90	
T MW 92	
:	
:	
CALL SFC 40	//Alarme freigeben
MODE :=B#16#2	//Mode 2: einzelne Alarm-OBs freigeben
OB_NR :=35	//OB 35 freigeben
RET_VAL :=MW 102	
U M 10.0	//Programmteil, der problemlos unterbrochen
S M 190.1	//werden kann:
U M 10.1	
S M 190.0	
:	
:	

A.4.4 Beispiel zur verzögerten Bearbeitung von Alarm- und Asynchronereignissen (SFC 41 und 42)

In diesem Beispiel für ein Anwenderprogramm wird ein Programmteil angenommen, der nicht durch Alarme unterbrochen werden darf. Für diesen Programmteil werden mit dem SFC 41 "DIS_AIRT" Alarme verzögert und später mit dem SFC 42 "EN_AIRT" wieder freigegeben.

In dem OB 1 werden der SFC 41 und der SFC 42 aufgerufen:

AWL (OB 1)	Erläuterung
U M 0.0	//Programmteil, der problemlos
S M 90.1	//unterbrochen werden kann:
U M 0.1	
S M 90.0	
:	
:	//Programmteil, der nicht durch
	//Interrupts unterbrochen werden darf:
CALL SFC 41	//Alarme sperren und verzögern
RET_VAL :=MW 100	
L PEW 100	
T MW 200	
L MW 90	
T MW 92	
:	
:	
:	
CALL SFC 42	//Alarm freigegeben
RET_VAL :=MW 102	
L MW 100	//Im Returnwert steht die Anzahl der
	//eingelegten Alarmsperren
DEC 1	
L MW 102	//im Returnwert steht die Anzahl der
	//eingelegten Alarmsperren
<>I	//die Anzahl muss nach der Alarmfreigabe
	//den gleichen Wert haben
SPB fehl	//wie vor der Alarmsperre (hier "0")
U M 10.0	//Programmteil, der problemlos
S M 190.1	//unterbrochen werden kann:
U M 10.1	
S M 190.0	
:	
:	
BEA	
fehl: L MW 102	//Anzahl der eingelegten Alarmsperren
T AW 12	//wird angezeigt

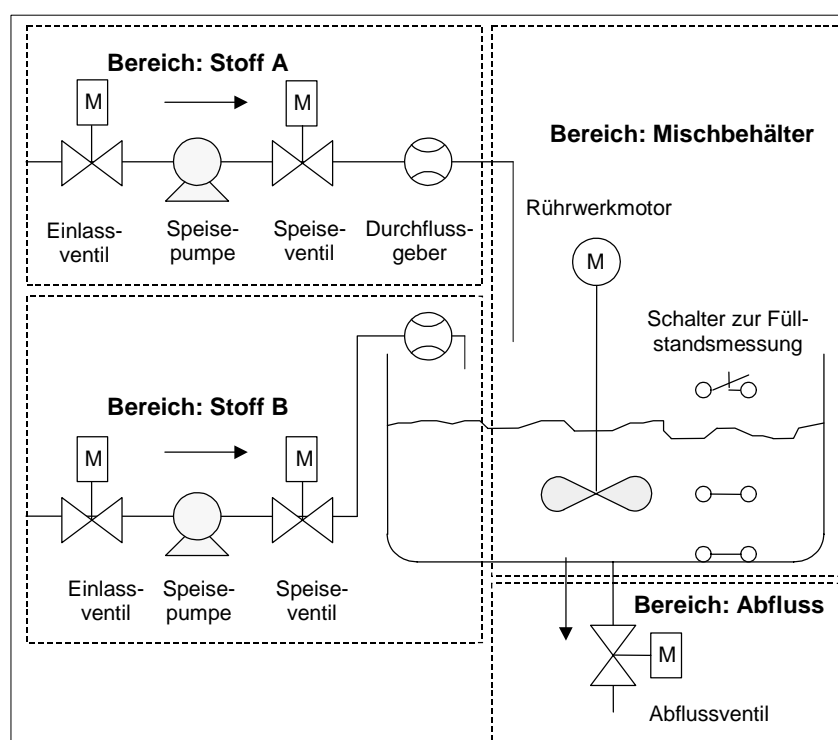
A.4.5 Beispielprogramm für einen industriellen Mischprozess

A.4.5.1 Beispielprogramm für einen industriellen Mischprozess

Das Beispielprogramm baut auf den Informationen auf, die bereits im Teil 1 des Handbuchs über das Steuern eines industriellen Mischprozesses vermittelt wurden.

Aufgabenstellung

Zwei Stoffe (Stoff A und Stoff B) sollen in einem Mischbehälter durch ein Rührwerk vermengt werden. Die Masse soll durch ein Abflussventil aus dem Mischbehälter abgelassen werden. Im nachfolgenden Bild sehen Sie ein Diagramm des Beispielprozesses.



Beschreibung der Teilprozesse

Im Teil 1 des Handbuchs wurde beschrieben, wie Sie den Beispielprozess in Funktionsbereiche und einzelne Aufgaben zerlegen können. Im folgenden finden Sie eine Beschreibung der einzelnen Teilbereiche.

Bereiche Stoff A und Stoff B:

- Die Stoffzuleitungen sind jeweils mit einem Einlass- und einem Speiseventil sowie einer Speisepumpe zu versehen.
- In den Zuleitungen befinden sich Durchflussgeber.
- Das Einschalten der Speisepumpen muss verriegelt sein, wenn der Füllstandsmesser "Behälter ist voll" anzeigt.
- Das Einschalten der Speisepumpen muss verriegelt sein, wenn das Abflussventil geöffnet ist.
- Frühestens 1 Sekunde nach Ansteuerung der Speisepumpe dürfen Einlass- und Speiseventil geöffnet werden.
- Sofort nach dem Stoppen der Speisepumpen (Signal des Durchflussgebers) müssen die Ventile geschlossen werden, um ein Abfließen des Stoffes aus der Pumpe zu verhindern.
- Die Ansteuerung der Speisepumpen ist zeitüberwacht, d. h. innerhalb von 7 Sekunden nach Ansteuerung muss vom Durchflussgeber ein Durchfluss gemeldet werden.
- Die Speisepumpen müssen in kürzester Zeit abgeschaltet werden, wenn die Durchflussgeber während des Laufs der Speisepumpen keinen Durchfluss mehr melden.
- Die Anzahl der Starts der Speisepumpen müssen gezählt werden (Wartungsintervall).

Bereich Mischbehälter:

- Das Einschalten des Rührwerkmotors muss verriegelt sein, wenn der Füllstandsmesser "Behälter unter Minimum" anzeigt oder das Abflussventil offen ist.
- Der Rührwerkmotor gibt ein Rückmeldesignal nach Erreichen der Nenndrehzahl. Wird dieses Signal nicht innerhalb von 10 Sekunden nach Ansteuerung des Motors gemeldet, muss der Motor abgeschaltet werden.
- Die Anzahl der Starts des Rührwerkmotors müssen gezählt werden (Wartungsintervall).
- Im Mischbehälter sind drei Geber zu installieren:
 - Behälter voll: Öffner. Ist der maximale Füllstand erreicht, wird der Kontakt geöffnet.
 - Füllstand im Behälter über Minimum: Schließer. Ist der minimale Füllstand erreicht, wird der Kontakt geschlossen.
 - Behälter nicht leer: Schließer. Ist der Behälter nicht leer, ist der Kontakt geschlossen.

Bereich Abfluss:

- Der Abfluss soll über ein Magnetventil gesteuert werden.
- Das Magnetventil wird durch den Bediener gesteuert, jedoch spätestens beim Signal "Behälter leer" wieder geschlossen.
- Das Öffnen des Abflussventils ist verriegelt, wenn
 - der Rührwerkmotor läuft
 - der "Behälter leer" ist

Bedienpult

Um einem Bediener das Starten und Stoppen sowie das Beobachten des Prozesses zu ermöglichen, soll außerdem ein Bedienpult angelegt werden. Auf dem Bedienpult befinden sich:

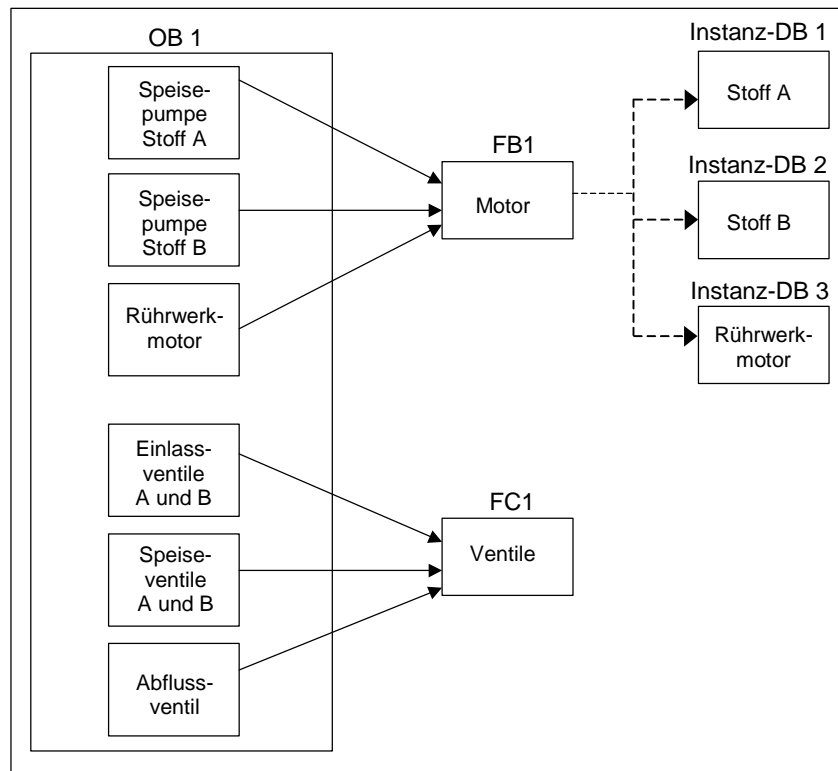
- Taster zum Steuern der wichtigsten Vorgänge. Mit dem Taster "Wartungsanzeige rücksetzen" schalten Sie die Wartungsanzeigelampen derjenigen Motoren aus, bei denen eine Wartung fällig ist, und setzen die zugehörigen Zählerstände für das Wartungsintervall auf 0.
- Anzeigelampen zum Ablesen des Betriebszustands.
- der NOT-AUS-Schalter.

A.4.5.2 Definieren von Codebausteinen

Mit dem Verteilen des Anwenderprogramms auf verschiedene Bausteine und der Hierarchie der Bausteinaufrufe legen Sie die Struktur des Programms fest.

Hierarchie der Bausteinaufrufe

Nachfolgendes Bild zeigt die Hierarchie der Bausteine, die im strukturierten Programm aufgerufen werden sollen.



- OB1: Er ist die Schnittstelle zum Betriebssystem der CPU und enthält das Hauptprogramm. Im OB 1 werden die Bausteine FB 1 und FC 1 aufgerufen und die spezifischen Parameter übergeben, die für die Steuerung des Prozesses erforderlich sind.
- FB 1: Die Speisepumpe für Stoff A, die Speisepumpe für Stoff B und der Rührwerk-motor können durch einen einzigen Funktionsbaustein gesteuert werden, da die Anforderungen (Ein-, Ausschalten, Einsätze zählen, etc.) identisch sind.
- Instanz-DB 1-3: Die Aktualparameter und die statischen Daten für die Steuerung der Speisepumpen für Stoff A, Stoff B und für den Rührwerk-motor sind unterschiedlich und werden deshalb in drei dem FB 1 zugeordneten Instanz-DBs abgelegt.
- FC 1: Die Einlass- und Speiseventile für die Stoffe A und B sowie das Abflussventil verwenden ebenfalls einen gemeinsamen Codebaustein. Da nur die Funktion Öffnen und Schließen programmiert werden muss, genügt ein einziger FC.

A.4.5.3 Zuordnen von symbolischen Namen

Definieren symbolischer Namen

In dem Beispielprogramm werden Symbole verwendet, die Sie mit STEP 7 Lite in der Symboltabelle definieren müssen. Nachfolgende Tabellen zeigen die symbolischen Namen und die absoluten Adressen der verwendeten Programmelemente.

Symbolische Adressen von Speisepumpe, Rührwerkmotor und Einlassventile			
Symbolischer Name	Adresse	Datentyp	Beschreibung
Feed_pump_A_start	E 0.0	BOOL	Starttaster Speisepumpe für Stoff A
Feed_pump_A_stop	E 0.1	BOOL	Stopptaster Speisepumpe für Stoff A
Flow_A	E 0.2	BOOL	Stoff A fließt
Inlet_valve_A	A 4.0	BOOL	Ansteuerung des Einlassventils für Stoff A
Feed_valve_A	A 4.1	BOOL	Ansteuerung des Speiseventils für Stoff A
Feed_pump_A_on	A 4.2	BOOL	Anzeigelampe "Speisepumpe Stoff A läuft"
Feed_pump_A_off	A 4.3	BOOL	Anzeigelampe "Speisepumpe Stoff A läuft nicht"
Feed_pump_A	A 4.4	BOOL	Ansteuerung der Speisepumpe für Stoff A
Feed_pump_A_fault	A 4.5	BOOL	Anzeigelampe "Fehler der Speisepumpe A"
Feed_pump_A_maint	A 4.6	BOOL	Anzeigelampe "Wartung der Speisepumpe A erforderlich"
Feed_pump_B_start	E 0.3	BOOL	Starttaster Speisepumpe für Stoff B
Feed_pump_B_stop	E 0.4	BOOL	Stopptaster Speisepumpe für Stoff B
Flow_B	E 0.5	BOOL	Stoff B fließt
Inlet_valve_B	A 5.0	BOOL	Ansteuerung des Einlassventils für Stoff B
Feed_valve_B	A 5.1	BOOL	Ansteuerung des Speiseventils für Stoff B
Feed_pump_B_on	A 5.2	BOOL	Anzeigelampe "Speisepumpe Stoff B läuft"
Feed_pump_B_off	A 5.3	BOOL	Anzeigelampe "Speisepumpe Stoff B läuft nicht"
Feed_pump_B	A 5.4	BOOL	Ansteuerung der Speisepumpe für Stoff B
Feed_pump_B_fault	A 5.5	BOOL	Anzeigelampe "Fehler der Speisepumpe B"
Feed_pump_B_maint	A 5.6	BOOL	Anzeigelampe "Wartung der Speisepumpe B erforderlich"
Agitator_running	E 1.0	BOOL	Rückmeldesignal des Rührwerkmotors
Agitator_start	E 1.1	BOOL	Starttaster Rührwerk
Agitator_stop	E 1.2	BOOL	Stopptaster Rührwerk
Agitator	A 8.0	BOOL	Ansteuerung des Rührwerks
Agitator_on	A 8.1	BOOL	Anzeigelampe "Rührwerk läuft"
Agitator_off	A 8.2	BOOL	Anzeigelampe "Rührwerk läuft nicht"
Agitator_fault	A 8.3	BOOL	Anzeigelampe "Fehler des Rührwerkmotors"
Agitator_maint	A 8.4	BOOL	Anzeigelampe "Wartung des Rührwerkmotors erforderlich"

Symbolische Adressen von Sensoren und Behälterfüllstandsanzeige			
Symbolischer Name	Adresse	Datentyp	Beschreibung
Tank_below_max	E 1.3	BOOL	Sensor "Mischbehälter nicht voll"
Tank_above_min	E 1.4	BOOL	Sensor "Mischbehälter über Minimum"
Tank_not_empty	E 1.5	BOOL	Sensor "Mischbehälter nicht leer"
Tank_max_disp	A 9.0	BOOL	Anzeigelampe "Mischbehälter voll"
Tank_min_disp	A 9.1	BOOL	Anzeigelampe "Mischbehälter unter Minimum"
Tank_empty_disp	A 9.2	BOOL	Anzeigelampe "Mischbehälter leer"

Symbolische Adressen des Abflussventils			
Symbolischer Name	Adresse	Datentyp	Beschreibung
Drain_open	E 0.6	BOOL	Taster zum Öffnen des Abflussventils
Drain_closed	E 0.7	BOOL	Taster zum Schließen des Abflussventils
Drain	A 9.5	BOOL	Ansteuerung des Abflussventils
Drain_open_disp	A 9.6	BOOL	Anzeigelampe "Abflussventil offen"
Drain_closed_disp	A 9.7	BOOL	Anzeigelampe "Abflussventil geschlossen"

Symbolische Adressen der übrigen Programmelemente			
Symbolischer Name	Adresse	Datentyp	Beschreibung
EMER_STOP_off	E 1.6	BOOL	NOT-AUS-Schalter
Reset_maint	E 1.7	BOOL	Resettaster für die Wartungsanzeigelampen aller Motoren
Motor_block	FB 1	FB 1	FB zum Steuern der Pumpen und des Rührwerk-motors
Valve_block	FC 1	FC 1	FC zum Steuern der Ventile
DB_feed_pump_A	DB 1	FB 1	Instanz-DB für die Steuerung der Speisepumpe A
DB_feed_pump_B	DB 2	FB 1	Instanz-DB für die Steuerung der Speisepumpe B
DB_agitator	DB 3	FB 1	Instanz-DB für die Steuerung des Rührwerk-motors

A.4.5.4 Erstellen des FB für den Motor

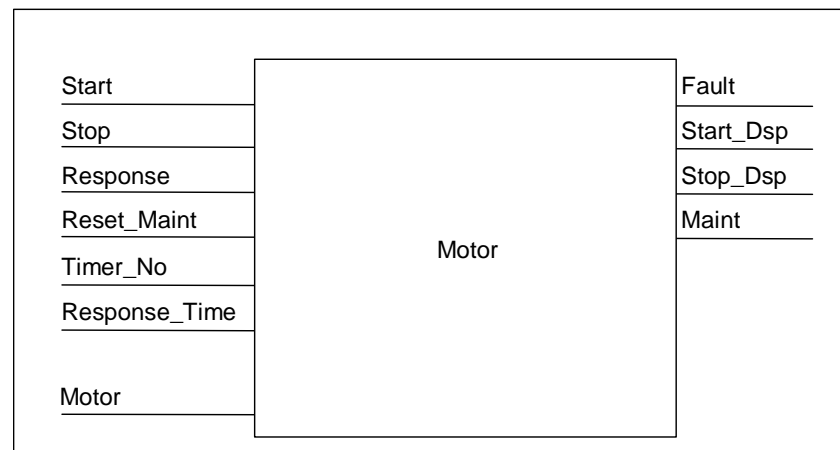
Anforderungen an den FB

Der FB für den Motor enthält die folgenden logischen Funktionen:

- Es gibt einen Start- und einen Stoppeingang.
- Verriegelungen ermöglichen den Betrieb der Geräte (Pumpen und Rührwerkmotor). Der Zustand der Verriegelungen wird in den temporären Lokaldaten (L-Stack) von OB 1 gespeichert (z. B. "Enable_Motor") und mit den Start- und Stoppeingängen verknüpft, wenn der FB für den Motor bearbeitet wird.
- Eine Rückmeldung von den Geräten muss innerhalb einer bestimmten Zeit erscheinen. Andernfalls wird angenommen, dass ein Fehler aufgetreten ist. Der FB stoppt daraufhin den Motor.
- Der Timer und die Zeitdauer für den Rückmelde- bzw. Fehlerzyklus müssen festgelegt werden.
- Ist der Starttaster gedrückt und die Freigabe erteilt, dann schaltet sich das Gerät ein und läuft solange, bis der Stopptaster gedrückt wird.
- Wird das Gerät eingeschaltet, wird ein Timer gestartet. Erscheint das Rückmeldesignal des Geräts nicht vor Ablauf des Timers, dann stoppt das Gerät.

Festlegen der Ein- und Ausgänge

Nachfolgendes Bild zeigt die Ein- und Ausgänge des allgemeinen FB für den Motor.



Definieren der Parameter für den FB

Wenn Sie einen wiederverwendbaren FB für den Motor (zum Steuern der beiden Pumpen und des Rührwerkmotors) erstellen möchten, müssen Sie allgemeine Parameternamen für die Eingänge und die Ausgänge definieren.

Der FB für den Motor im Beispielprozess hat die folgenden Anforderungen:

- Er benötigt Signale vom Bedienpult zum Stoppen bzw. Starten des Motors und der Pumpen.
- Er benötigt ein Rückmeldesignal von Motor und Pumpen, dass der Motor läuft.
- Er muss die Zeit zwischen dem Senden des Signals, das den Motor einschaltet, und dem Empfangen des Rückmeldesignals ermitteln. Erscheint nach einer bestimmten Zeit kein Rückmeldesignal, muss der Motor ausgeschaltet werden.
- Er muss die entsprechenden Anzeigen auf dem Bedienpult ein- bzw. ausschalten.
- Er liefert ein Signal zur Ansteuerung des Motors.

Diese Anforderungen können als Eingänge und Ausgänge des FB festgelegt werden. Nachfolgende Tabelle zeigt die Parameter des FB für den Motor in unserem Beispielprozess.

Parametername	Eingang	Ausgang	Durchgang
Start	n		
Stop	n		
Response	n		
Reset_Maint	n		
Timer_No	n		
Response_Time	n		
Fault		n	
Start_Dsp		n	
Stop_Dsp		n	
Maint		n	
Motor			n

Deklarieren der Variablen des FB für den Motor

Sie müssen die Eingangs-, Ausgangs- und Durchgangparameter des FB für den Motor deklarieren.

Adresse	Deklaration	Name	Typ	Anfangswert
0.0	in	Start	BOOL	FALSE
0.1	in	Stop	BOOL	FALSE
0.2	in	Response	BOOL	FALSE
0.3	in	Reset_Maint	BOOL	FALSE
2.0	in	Timer_No	TIMER	

Adresse	Deklaration	Name	Typ	Anfangswert
4.0	in	Response_Time	S5TIME	S5T#0MS
6.0	out	Fault	BOOL	FALSE
6.1	out	Start_Dsp	BOOL	FALSE
6.2	out	Stop_Dsp	BOOL	FALSE
6.3	out	Maint	BOOL	FALSE
8.0	lin_out	Motor	BOOL	FALSE
10.0	stat	Timer_bin	WORD	W#16#0
12.0	stat	Timer_BCD	WORD	W#16#0
14.0	stat	Starts	INT	0
16.0	stat	Start_Edge	BOOL	FALSE

Bei FBs werden die Eingangs-, Ausgangs-, Durchgangsparameter und die statischen Variablen in dem Instanz-DB gespeichert, der in der Aufrufanweisung angegeben wird. Die temporären Variablen werden im L-Stack gespeichert.

Programmieren des FB für den Motor

In STEP 7 Lite muss jeder Baustein, der von einem anderen Baustein aufgerufen wird, vor dem Baustein erstellt werden, der den Aufruf enthält. Im Beispielprogramm müssen Sie also den FB für den Motor vor dem OB 1 erstellen.

Der Anweisungsteil des FB 1 sieht in der Programmiersprache AWL folgendermaßen aus:

Netzwerk 1 Start/Stop und Selbsthaltung

```

U(
O   #Start
O   #Motor
)
UN  #Stop
=   #Motor

```

Netzwerk 2 Anlaufüberwachung

```

U   #Motor
L   #Response_Time
SE  #Timer_No
UN  #Motor
R   #Timer_No
L   #Timer_No
T   #Timer_bin
LC  #Timer_No
T   #Timer_BCD
U   #Timer_No
UN  #Response
S   #Fault
R   #Motor

```

Netzwerk 3 Startlampe und Fehlerrücksetzung

```
U    #Response
=    #Start_Dsp
R    #Fault
```

Netzwerk 4 Stoplampe

```
UN   #Response
=    #Stop_Dsp
```

Netzwerk 5 Zählung der Starts

```
U    #Motor
FP   #Start_Edge
SPBN lab1
L    #Starts
+    1
T    #Starts

lab1: NOP    0
```

Netzwerk 6 Wartungsanzeigelampe

```
L    #Starts
L    50
>=I
=    #Maint
```

Netzwerk 7 Rücksetzen des Zählers für die Anzahl der Starts

```
U    #Reset_Maint
U    #Maint
SPBN END
L    0
T    #Starts

END: NOP    0
```

Erstellen der Instanz-Datenbausteine

Zum Anlegen der drei Datenbausteine wählen Sie jeweils den Menübefehl **Einfügen > Baustein**. Klicken Sie die Option "Datenbaustein", tragen die entsprechende Datenbausteinnummer ein und wählen Sie in der nebenliegenden Klappliste den Funktionsbaustein FB 1 aus. Damit sind die Datenbausteine als Instanz-Datenbausteine mit fester Zuordnung zum FB 1 festgelegt.

A.4.5.5 Erstellen der FC für die Ventile

Anforderungen an die FC

Die Funktion für die Einlass- und Speiseventile sowie für das Abflussventil enthält die folgenden logischen Funktionen:

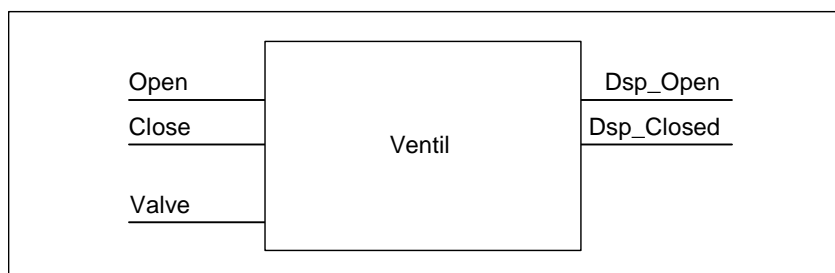
- Es gibt je einen Eingang zum Öffnen und zum Schließen der Ventile.
- Verriegelungen ermöglichen das Öffnen und Schließen der Ventile. Der Zustand der Verriegelungen wird in den temporären Lokaldaten (L-Stack) von OB 1 gespeichert (z. B. "Enable_Valve") und mit den Eingängen zum Öffnen und Schließen verknüpft, wenn die FC für die Ventile bearbeitet wird.

Nachfolgende Tabelle zeigt die Parameter, die an die FC übergeben werden müssen.

Parameter für die Ventile	Eingang	Ausgang	Durchgang
Open	✓		
Close	✓		
Dsp_Open		✓	
Dsp_Closed		✓	
Valve			✓

Festlegen der Ein- und Ausgänge

Nachfolgendes Bild zeigt die Ein- und Ausgänge der allgemeinen FC für die Ventile. Die Geräte, die den FB für den Motor aufrufen, übergeben Eingangsparameter. Die FC für die Ventile gibt Ausgangsparameter zurück.



Deklarieren der Variablen der FC für die Ventile

Wie bei dem FB für den Motor müssen Sie auch bei der FC für die Ventile die Eingangs-, Ausgangs- und Durchgangsparameter deklarieren (siehe nachfolgende Variablen Deklarationstabelle).

Adresse	Deklaration	Name	Typ	Anfangswert
0.0	in	Open	BOOL	FALSE
0.1	in	Close	BOOL	FALSE
2.0	out	Dsp_Open	BOOL	FALSE
2.1	out	Dsp_Closed	BOOL	FALSE
4.0	In_out	Valve	BOOL	FALSE

Bei FCs werden die temporären Variablen im L-Stack gespeichert. Die Eingangs-, Ausgangs- und Durchgangsvariablen werden als Pointer auf den Codebaustein gespeichert, der die FC aufgerufen hat. Zusätzlicher Speicherplatz im L-Stack (nach den temporären Variablen) wird für diese Variablen verwendet.

Programmieren der FC für die Ventile

Auch die Funktion FC 1 für die Ventile müssen Sie vor dem OB 1 erstellen, da aufgerufene Bausteine vor aufrufenden Bausteinen erstellt werden müssen.

Der Anweisungsteil des FC 1 sieht in der Programmiersprache AWL folgendermaßen aus:

Netzwerk 1 Öffnen/Schließen und Selbsthaltung

```

U(
O   #Open
O   #Valve
)
UN  #Close
=   #Valve

```

Netzwerk 2 Anzeige, OB Ventil geöffnet

```

U   #Valve
=   #Dsp_Open

```

Netzwerk 3 Anzeige, OB Ventil geschlossen

```

UN  #Valve
=   #Dsp_Closed

```

A.4.5.6 Erstellen des OB 1

OB 1 bestimmt die Struktur des Beispielprogramms. Außerdem enthält OB 1 die Parameter, die an die verschiedenen Bausteine übergeben werden, z. B.:

- Die AWL-Netzwerke für die Speisepumpen und den Rührwerkmotor liefern dem FB für den Motor die Eingangsparameter zum Starten ("Start"), Stoppen ("Stop"), für die Rückmeldung ("Response") und für das Rücksetzen der Wartungsanzeige ("Reset_Maint"). Der FB für den Motor wird in jedem Zyklus des AS bearbeitet.
- Wird der FB für den Motor bearbeitet, legen die Eingänge "Timer_No" und "Response_Time" fest, welcher Timer verwendet wird und innerhalb welcher Zeitdauer eine Rückmeldung erfolgen muss.
- Die FC für die Ventile und der FB für die Motoren werden in jedem Programmzyklus des Automatisierungssystems bearbeitet, da sie im OB 1 aufgerufen werden.

Das Programm verwendet den FB für den Motor mit verschiedenen Instanz-DBs, um die Aufgaben für die Steuerung der Speisepumpen und des Rührwerkmotors zu erfüllen.

Deklariieren von Variablen für OB 1

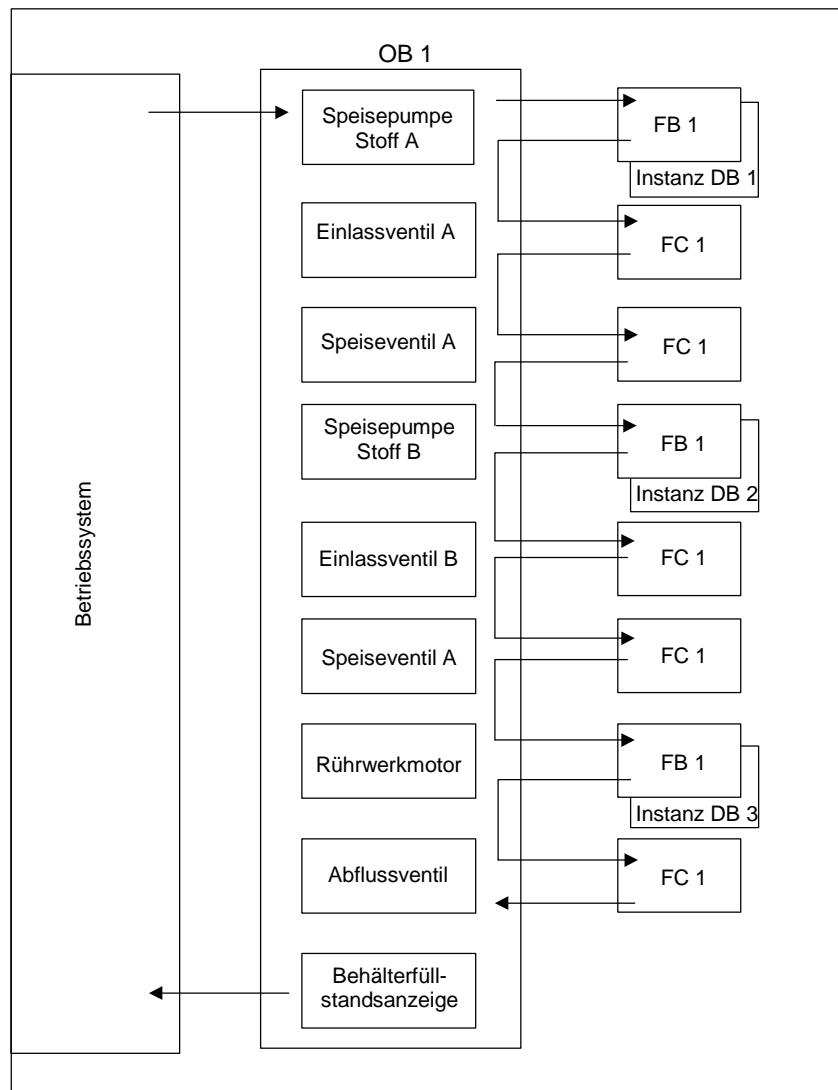
Nachfolgend wird die Variablendeklarationstabelle für den OB 1 abgebildet. Die ersten 20 Bytes enthalten die Startinformationen des OB 1 und dürfen nicht geändert werden.

Adresse	Deklaration	Name	Typ
0.0	temp	OB1_EV_CLASS	BYTE
1.0	temp	OB1_SCAN1	BYTE
2.0	temp	OB1_PRIORITY	BYTE
3.0	temp	OB1_OB_NUMBR	BYTE
4.0	temp	OB1_RESERVED_1	BYTE
5.0	temp	OB1_RESERVED_2	BYTE
6.0	temp	OB1_PREV_CYCLE	INT
8.0	temp	OB1_MIN_CYCLE	INT
10.0	temp	OB1_MAX_CYCLE	INT
12.0	temp	OB1_DATE_TIME	DATE_AND_TIME
20.0	temp	Enable_Motor	BOOL
20.1	temp	Enable_Valve	BOOL
20.2	temp	Start_Fulfilled	BOOL
20.3	temp	Stop_Fulfilled	BOOL
20.4	temp	Inlet_Valve_A_Open	BOOL
20.5	temp	Inlet_Valve_A_Closed	BOOL
20.6	temp	Feed_Valve_A_Open	BOOL
20.7	temp	Feed_Valve_A_Closed	BOOL
21.0	temp	Inlet_Valve_B_Open	BOOL
21.1	temp	Inlet_Valve_B_Closed	BOOL
21.2	temp	Feed_Valve_B_Open	BOOL
21.3	temp	Feed_Valve_B_Closed	BOOL
21.4	temp	Open_Drain	BOOL
21.5	temp	Close_Drain	BOOL
21.6	temp	Close_Valve_Fulfilled	BOOL

Erstellen des Programms für OB 1

In STEP 7 Lite muss jeder Baustein, der von einem anderen Baustein aufgerufen wird, vor dem Baustein erstellt werden, der den Aufruf enthält. In dem Beispielprogramm müssen Sie sowohl den FB für den Motor als auch die FC für die Ventile vor dem Programm in OB 1 erstellen.

Die Bausteine FB 1 und FC 1 werden im OB 1 mehrfach aufgerufen, der FB 1 mit verschiedenen Instanz-DBs:



Der Anweisungsteil des OB 1 sieht in der Programmiersprache AWL folgendermaßen aus:

Netzwerk 1 Verriegelungen für Speisepumpe A

```

U    "EMER_STOP_off"
U    "Tank_below_max"
UN   "Drain"
=    #Enable_Motor

```

Netzwerk 2 Aufruf FB Motor für Stoff A

```

U            "Feed_pump_A_start"
U            #Enable_Motor
=            #Start_Fulfilled
U(
O            "Feed_pump_A_stop"
ON   #Enable_Motor
)
=            #Stop_Fulfilled
CALL "Motor_block", "DB_feed_pump_A"
Start   :=#Start_Fulfilled
Stop    :=#Stop_Fulfilled
Response :="Flow_A"
Reset_Maint :="Reset_maint"
Timer_No :=T12
Reponse_Time:=S5T#7S
Fault   :="Feed_pump_A_fault"
Start_Dsp :="Feed_pump_A_on"
Stop_Dsp  :="Feed_pump_A_off"
Maint    :="Feed_pump_A_maint"
Motor    :="Feed_pump_A"

```

Netzwerk 3 Verzögerung der Ventilfeigabe Stoff A

```

U    "Feed_pump_A"
L    S5T#1S
SE   T    13
UN   "Feed_pump_A"
R    T    13
U    T    13
=    #Enable_Valve

```

Netzwerk 4 Einlassventilsteuerung für Stoff A

```

UN   "Flow_A"
UN   "Feed_pump_A"
=    #Close_Valve_Fulfilled
CALL "Valve_block"
Open  :=#Enable_Valve
Close :=#Close_Valve_Fulfilled
Dsp_Open :=#Inlet_Valve_A_Open
Dsp_Closed:=#Inlet_Valve_A_Closed
Valve :="Inlet_Valve_A"

```

Netzwerk 5 Speiseventilsteuerung für Stoff A

```

UN   "Flow_A"
UN   "Feed_pump_A"
=     #Close_Valve_Fulfilled
CALL "Valve_block"
      Open  :=#Enable_Valve
      Close :=#Close_Valve_Fulfilled
      Dsp_Open   :=#Feed_Valve_A_Open
      Dsp_Closed:=#Feed_Valve_A_Closed
      Valve  :="Feed_Valve_A"

```

Netzwerk 6 Verriegelungen für Speisepumpe B

```

U     "EMER_STOP_off"
U     "Tank_below_max"
UN    "Drain"
=     #Enable_Motor

```

Netzwerk 7 Aufruf FB Motor für Stoff B

```

U           "Feed_pump_B_start"
U           #Enable_Motor
=           #Start_Fulfilled
U(
O           "Feed_pump_B_stop"
ON          #Enable_Motor
)
=           #Stop_Fulfilled
CALL "Motor_block", "DB_feed_pump_B"
      Start :=#Start_Fulfilled
      Stop  :=#Stop_Fulfilled
      Response :="Flow_B"
      Reset_Maint :="Reset_maint"
      Timer_No :=T14
      Reponse_Time:=S5T#7S
      Fault :="Feed_pump_B_fault"
      Start_Dsp :="Feed_pump_B_on"
      Stop_Dsp :="Feed_pump_B_off"
      Maint :="Feed_pump_B_maint"
      Motor :="Feed_pump_B"

```

Netzwerk 8 Verzögerung der Ventilfeigabe Stoff B

```

U     "Feed_pump_B"
L     S5T#1S
SE    T     15
UN    "Feed_pump_B"
R     T     15
U     T     15
=     #Enable_Valve

```

Netzwerk 9 Einlassventilsteuerung für Stoff B

```

UN   "Flow_B"
UN   "Feed_pump_B"
=     #Close_Valve_Fulfilled
CALL "Valve_block"
      Open   :=#Enable_Valve
      Close  :=#Close_Valve_Fulfilled
      Dsp_Open   :=#Inlet_Valve_B_Open
      Dsp_Closed:=#Inlet_Valve_B_Closed
      Valve    :="Inlet_Valve_B"

```

Netzwerk 10 Speiseventilsteuerung für Stoff B

```

UN   "Flow_B"
UN   "Feed_pump_B"
=     #Close_Valve_Fulfilled
CALL "Valve_block"
      Open   :=#Enable_Valve
      Close  :=#Close_Valve_Fulfilled
      Dsp_Open   :=#Feed_Valve_B_Open
      Dsp_Closed:=#Feed_Valve_B_Closed
      Valve    :="Feed_Valve_B"

```

Netzwerk 11 Verriegelungen für Rührwerk

```

U     "EMER_STOP_off"
U     "Tank_above_min"
UN    "Drain"
=     #Enable_Motor

```

Netzwerk 12 Aufruf FB Motor für Rührwerk

```

U     "Agitator_start"
U     #Enable_Motor
=     #Start_Fulfilled
U(
O     "Agitator_stop"
ON    #Enable_Motor
)
=     #Stop_Fulfilled
CALL "Motor_block", "DB_agitator"
      Start   :=#Start_Fulfilled
      Stop    :=#Stop_Fulfilled
      Response :="Agitator_running"
      Reset_Maint :="Reset_maint"
      Timer_No   :=T16
      Reponse_Time:=S5T#10S
      Fault     :="Agitator_fault"
      Start_Dsp :="Agitator_on"
      Stop_Dsp  :="Agitator_off"
      Maint    :="Agitator_maint"
      Motor    :="Agitator"

```

Netzwerk 13 Verriegelungen für Abflussventil

```
U    "EMER_STOP_off"
U    "Tank_not_empty"
UN   "Agitator"
=    "Enable_Valve"
```

Netzwerk 14 Abflussventilsteuerung

```
U          "Drain_open"
U          #Enable_Valve
=          #Open_Drain
U(
O          "Drain_closed"
ON   #Enable_Valve
)
=          #Close_Drain
CALL "Valve_block"
      Open  :=#Open_Drain
      Close :=#Close_Drain
      Dsp_Open   :="Drain_open_disp"
      Dsp_Closed :="Drain_closed_disp"
      Valve    :="Drain"
```

Netzwerk 15 Behälterfüllstandsanzeige

```
UN   "Tank_below_max"
=    "Tank_max_disp"
UN   "Tank_above_min"
=    "Tank_min_disp"
UN   "Tank_not_empty"
=    "Tank_empty_disp"
```

A.4.6 Beispiel zur Hantierung mit Uhrzeitalarmen

Struktur des Anwenderprogramms "Uhrzeitalarme"

FC 12

OB 10

OB 1 und OB 80

A.4.6.1 Struktur des Anwenderprogramms Uhrzeitalarme

Aufgabenstellung

Ausgang A 4.0 soll in der Zeit von Montag, 5.00 Uhr bis Freitag, 20.00 Uhr gesetzt sein. In der Zeit von Freitag, 20.00 Uhr bis Montag, 5.00 Uhr soll der Ausgang A 4.0 zurückgesetzt sein.

Umsetzung im Anwenderprogramm

Nachfolgende Tabelle zeigt die Teilaufgaben der verwendeten Bausteine.

Baustein	Teilaufgabe
OB 1	Aufruf der Funktion FC 12
FC 12	In Abhängigkeit vom Zustand des Ausgangs A 4.0, des Uhrzeitalarm-Status und der Eingänge E 0.0 und E 0.1 <ul style="list-style-type: none"> • Startzeitpunkt vorgeben • Uhrzeitalarm setzen • Uhrzeitalarm aktivieren • CAN_TINT
OB 10	Abhängig vom aktuellen Wochentag <ul style="list-style-type: none"> • Startzeitpunkt vorgeben • Ausgang A 4.0 setzen bzw. rücksetzen • nächsten Uhrzeitalarm setzen • nächsten Uhrzeitalarm aktivieren
OB 80	Setzen des Ausgangs A 4.1 Startereignisinformation des OB 80 speichern im Merkerbereich

Verwendete Operanden

Nachfolgende Tabelle zeigt die verwendeten globalen Operanden. Die bausteintemporären Variablen sind im Deklarationsteil des jeweiligen Bausteins deklariert.

Operand	Bedeutung
E 0.0	Eingang für die Freigabe von "Uhrzeitalarm setzen" und Uhrzeitalarm aktivieren"
E 0.1	Eingang für die Stornierung eines Uhrzeitalarms
A 4.0	Ausgang, der vom Uhrzeitalarm-OB (OB 10) gesetzt/rückgesetzt wird
A 4.1	Ausgang, der bei einem Zeitfehler (OB 80) gesetzt wird
MW 16	STATUS des Uhrzeitalarms (SFC 31 "QRY_TINT")
MB 100 bis MB 107	Speicher für Starterereignisinformation des OB 10 (nur Uhrzeit)
MB 110 bis MB 129	Speicher für Starterereignisinformation des OB 80 (Zeitfehler)
MW 200	RET_VAL des SFC 28 "SET_TINT"
MB 202	Binäres Ergebnis- (Statusbit BIE)-Zwischenspeicher für SFCs
MW 204	RET_VAL des SFC 30 "ACT_TINT"
MW 208	RET_VAL des SFC 31 "QRY_TINT"

Verwendete SFCs und FCs

Im Programmbeispiel werden die folgenden Systemfunktionen und Funktionen verwendet:

- SFC 28 "SET_TINT" : Uhrzeitalarm stellen
- SFC 29 "CAN_TINT" : Uhrzeitalarm stornieren
- SFC 30 "ACT_TINT" : Uhrzeitalarm aktivieren
- SFC 31 "QRY_TINT" : Uhrzeitalarm abfragen
- FC 3 "D_TOD_DT" : DATE und TIME_OF_DAY zu DT zusammenfassen

A.4.6.2 FC 12

Deklarationsteil

Im Deklarationsteil des FC 12 werden folgende bausteintemporäre Variablen deklariert:

Variablenname	Datentyp	Deklaration	Kommentar
IN_UHRZEIT	TIME_OF_DAY	temp	Startzeit Vorgabe
IN_DATUM	DATE	temp	Startdatum Vorgabe
OUT_UHRZEIT_DATUM	DATE_AND_TIME	temp	Startdatum/-zeit gewandelt
OK_MERKER	BOOL	temp	Freigabe für Uhrzeitalarm-Stellen

Anweisungsteil AWL

Im Anweisungsteil des FC 12 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (FC 12)	Erläuterung
<pre> Network 1: CALL SFC 31 OB_NR := 10 RET_VAL:= MW 208 STATUS := MW 16 </pre>	<pre> //SFC QRY_TINT //STATUS Uhrzeitalarme abfragen. </pre>
<pre> Network 2: UN A 4.0 SPB mont L D#1995-1-27 T #IN_DATUM L TOD#20:0:0.0 T #IN_UHRZEIT SPA wndl mont: L D#1995-1-23 T #IN_DATUM L TOD#5:0:0.0 T #IN_UHRZEIT wndl: NOP 0 </pre>	<pre> //Startzeitpunkt vorgeben in //Abhaengigkeit von A 4.0 (in Variable //#IN_DATUM und #IN_UHRZEIT) //Startdatum ist ein Freitag. //Startdatum ist ein Montag. </pre>

AWL (FC 12)	Erläuterung
Network 3:	
CALL FC 3	//Format wandeln von DATE und
IN1 := #IN_DATUM	//TIME_OF_DAY nach DATE_AND_TIME (für
IN2 := #IN_UHRZEIT	//Uhrzeitalarm stellen)
RET_VAL := #OUT_UHRZEIT_DATUM	
Network 4:	
U E 0.0	//Alle Voraussetzungen für Uhrzeitalarm
UN M 17.2	//stellen erfüllt? (Eingang für
U M 17.4	//Freigabe gesetzt und Uhrzeitalarm
= #OK_MERKER	//nicht aktiv und Uhrzeitalarm-OB ist
	//geladen)
Network 5:	//Wenn ja, dann Uhrzeitalarm stellen...
U #OK_MERKER	
SPBNB m001	
CALL SFC 28	
OB_NR := 10	
SDT := #OUT_UHRZEIT_DATUM	
PERIOD := W#16#1201	
RET_VAL := MW 200	
m001 U BIE	
= M 202.3	
Network 6:	//...und Uhrzeitalarm aktivieren.
U #OK_MERKER	
SPBNB m002	
CALL SFC 30	
OB_NR := 10	
RET_VAL := MW 204	
m002 U BIE	
= M 202.4	
Network 7:	//Wenn Eingang für die Stornierung von
U E 0.1	//Uhrzeitalarmen gesetzt ist, dann
SPBNB m003	//Uhrzeitalarm stornieren.
CALL SFC 29	
OB_NR := 10	
RET_VAL := MW 210	
m003 U BIE	
= M 202.5	

A.4.6.3 OB 10

Deklarationsteil

Abweichend vom voreingestellten Deklarationsteil des OB 10 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)
- Weitere bausteintemporäre Variablen WTAG, IN_DATUM, IN_UHRZEIT und OUT_UHRZEIT_DATUM

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	temp	Gesamte Startereignisinformation des OB 10 als Struktur deklariert
E_ID	WORD	temp	Ereignis-ID
PR_KLASSE	BYTE	temp	Prioritätsklasse
OB_NR	BYTE	temp	OB-Nummer
RESERVED_1	BYTE	temp	Reserviert
RESERVED_2	BYTE	temp	Reserviert
PERIODE	WORD	temp	Periodizität des Uhrzeitalarms
RESERVED_3	DWORD	temp	Reserviert
T_STMP	STRUCT	temp	Struktur für Uhrzeitangaben
JAHR	BYTE	temp	
MONAT	BYTE	temp	
TAG	BYTE	temp	
STUNDE	BYTE	temp	
MINUTEN	BYTE	temp	
SEKUNDEN	BYTE	temp	
MSEK_WTAG	WORD	temp	
	END_STRUCT	temp	
	END_STRUCT	temp	
WTAG	INT	temp	Wochentag
IN_DATUM	DATE	temp	Eingangsvariable für FC 3 (Wandlung des Zeitformates)
IN_UHRZEIT	TIME_OF_DAY	temp	Eingangsvariable für FC 3 (Wandlung des Zeitformates)
OUT_UHRZEIT_DATUM	DATE_AND_TIME	temp	Ausgangsvariable für FC 3 und Eingangsvariable für SFC 28

Anweisungsteil AWL

Im Anweisungsteil des OB 10 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 10)	Erläuterung
Network 1:	
L #STARTINFO.T_STMP.MSEK_WTAG	//Wochentag selektieren
L W#16#F	
UW	
T #WTAG	//und merken.
Network 2:	
L #WTAG	//Wenn Wochentag nicht Montag ist,
L 2	//dann Montag, 5.00 Uhr als nächsten
<>I	//Startzeitpunkt vorgeben und
SPB mont	//Ausgang A 4.0 rücksetzen.
Network 3:	
L D#1995-1-27	
T #IN_DATUM	//Sonst, d. h. wenn Wochentag = Montag
L TOD#20:0:0.0	//ist, dann Freitag, 20.00 Uhr als
T #IN_UHRZEIT	//nächsten Startzeitpunkt vorgeben und
SET	//Ausgang A 4.0 setzen.
= A 4.0	
SPA wnd1	
mont:	
L D#1995-1-23	
T #IN_DATUM	
L TOD#5:0:0.0	
T #IN_UHRZEIT	
CLR	
= A 4.0	
wnd1:	
NOP 0	//Startzeitpunkt vorgeben abgeschlossen.
Network 4:	//Vorgegebenen Startzeitpunkt in Format
CALL FC 3	//DATE_AND_TIME wandeln (für SFC 28).
IN1 := #IN_DATUM	
IN2 := #IN_UHRZEIT	
RET_VAL := #OUT_UHRZEIT_DATUM	//Uhrzeitalarm stellen.
Network 5:	
CALL SFC 28	
OB_NR := 10	
SDT := #OUT_UHRZEIT_DATUM	
PERIOD := W#16#1201	
RET_VAL := MW 200	
U BIE	
= M 202.1	
Network 6:	
CALL SFC 30	//Uhrzeitalarm aktivieren.
OB_NR := 10	
RET_VAL := MW 204	
U BIE	
= M 202.2	
Network 7:	
CALL SFC 20	//Blocktransfer: Uhrzeitangabe aus
SRCBLK := #STARTINFO.T_STMP	//Startereignisinformation des OB 10 in
RET_VAL := MW 206	//den Merkerbereich MB 100 bis MB 107
DSTBLK := P#M 100.0 BYTE 8	//retten.

A.4.6.4 OB 1 und OB 80

Da die Startereignisinformation des OB 1 (OB für zyklisches Programm) in diesem Beispiel nicht ausgewertet wird, ist nur die Startereignisinformation des OB 80 dargestellt.

Anweisungsteil OB 1

Im Anweisungsteil des OB 1 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 1)	Erläuterung
CALL FC 12	//Aufruf der Funktion FC 12

Deklarationsteil OB 80

Abweichend vom voreingestellten Deklarationsteil des OB 80 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	temp	Gesamte Startereignisinformation des OB 80 als Struktur deklariert
E_ID	WORD	temp	Ereignis-ID
PR_KLASSE	BYTE	temp	Prioritätsklasse
OB_NR	BYTE	temp	OB-Nummer
RESERVED_1	BYTE	temp	Reserviert
RESERVED_2	BYTE	temp	Reserviert
Z1_INFO	WORD	temp	Zusatzinformation über das Ereignis, das den Fehler verursacht hat.
Z2_INFO	DWORD	temp	Zusatzinformation über Ereignis-ID, Prioritätsklasse und OB-Nr. des Fehlerereignisses
T_STMP	STRUCT	temp	Struktur für Uhrzeitangaben
Jahr	BYTE	temp	
MONAT	BYTE	temp	
TAG	BYTE	temp	
STUNDE	BYTE	temp	
MINUTEN	BYTE	temp	
SEKUNDEN	BYTE	temp	
MSEK_WTAG	WORD	temp	
	END_STRUCT	temp	
	END_STRUCT	temp	

Anweisungsteil OB 80

Im Anweisungsteil des OB 80, der vom Betriebssystem bei einem Zeitfehler aufgerufen wird, geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 80)	Erläuterung
Network 1:	
UN A 4.1	//Ausgang A 4.1 setzen, wenn Zeitfehler
S A 4.1	//aufgetreten ist.
CALL SFC 20	//Blocktransfer: Gesamte Start-
SRCBLK := #STARTINFO	//ereignisinformation in den Merker-
RET_VAL := MW 210	//bereich MB 110 bis MB 129 retten.
DSTBLK := P#M 110.0 Byte 20	

A.4.7 Beispiel zur Hantierung mit Verzögerungsalarmen

A.4.7.1 Struktur des Anwenderprogramms Verzögerungsalarme

Aufgabenstellung

Wenn Eingang E 0.0 gesetzt wird, soll Ausgang A 4.0 10 Sekunden später gesetzt werden. Jedes Setzen von Eingang E 0.0 soll die Verzögerungszeit neu starten.

Als anwendungsspezifisches Kennzeichen soll der Zeitpunkt (Sekunden und Millisekunden) des Verzögerungsalarm-Starts in der Startereignisinformation des Verzögerungsalarm-OB (OB 20) erscheinen.

Falls in diesen 10 Sekunden E 0.1 gesetzt wird, soll der Organisationsbaustein OB 20 nicht aufgerufen werden; d. h. der Ausgang A 4.0 soll nicht gesetzt werden.

Wenn Eingang E 0.2 gesetzt wird, soll Ausgang A 4.0 wieder zurückgesetzt werden.

Umsetzung im Anwenderprogramm

Nachfolgende Tabelle zeigt die Teilaufgaben der verwendeten Bausteine.

Baustein	Teilaufgabe
OB 1	Aktuelle Uhrzeit lesen und aufbereiten für Start des Verzögerungsalarms Abhängig vom Flankenwechsel am Eingang E 0.0 Verzögerungsalarm starten Abhängig vom Status des Verzögerungsalarms und vom Flankenwechsel am Eingang E 0.1 Verzögerungsalarm stornieren Abhängig vom Zustand des Eingangs E 0.2 Ausgang A 4.0 zurücksetzen
OB 20	Setzen des Ausganges A 4.0 Aktuelle Uhrzeit lesen und aufbereiten Startereignisinformation in den Merkerbereich retten

Verwendete Operanden

Nachfolgende Tabelle zeigt die verwendeten Globaldaten. Die bausteintemporären Variablen sind im Deklarationsteil des jeweiligen Bausteins deklariert.

Operand	Bedeutung
E 0.0	Eingang für die Freigabe von "Verzögerungsalarm starten"
E 0.1	Eingang für die Stornierung eines Verzögerungsalarms
E 0.2	Eingang für das Rücksetzen des Ausgangs A 4.0
A 4.0	Ausgang, der vom Verzögerungsalarm-OB (OB 20) gesetzt wird
MB 1	Genutzt für Flankenmerker und Binärergebnis- (Statusbit BIE)-Zwischenspeicher für SFCs
MW 4	STATUS des Verzögerungsalarms (SFC 34 "QRY_TINT")
MD 10	Sekunden und Millisekunden BCD-codiert aus der Starterereignisinformation des OB 1
MW 100	RET_VAL des SFC 32 "SRT_DINT"
MW 102	RET_VAL des SFC 34 "QRY_DINT"
MW 104	RET_VAL des SFC 33 "CAN_DINT"
MW 106	RET_VAL des SFC 20 "BLKMOV"
MB 120 bis MB 139	Speicher für Starterereignisinformation des OB 20
MD 140	Sekunden und Millisekunden BCD-codiert aus der Starterereignisinformation des OB 20
MW 144	Sekunden und Millisekunden BCD-codiert aus der Starterereignisinformation des OB 1; gewonnen aus Starterereignisinformation des OB 20 (anwenderspezifisches Kennzeichen SIGN)

Verwendete SFCs

Im Anwenderprogramm "Verzögerungsalarme" werden die folgenden Systemfunktionen verwendet:

- SFC 32 "SRT_DINT" : Verzögerungsalarm starten
- SFC 33 "CAN_DINT" : Verzögerungsalarm stornieren
- SFC 34 "QRY_DINT" : Zustand eines Verzögerungsalarms abfragen

A.4.7.2 OB 20

Deklarationsteil

Abweichend vom voreingestellten Deklarationsteil des OB 20 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	temp	Start-Info zu OB 20
E_ID	WORD	temp	Ereignis-ID
AE_NR	BYTE	temp	Ablaufebene
OB_NR	BYTE	temp	OB-Nr
DK1	BYTE	temp	Datenkennung 1
DK2	BYTE	temp	Datenkennung 2
SIGN	WORD	temp	anwenderspezifisches Kennzeichen
DTIME	TIME	temp	Zeit, mit der Verzögerungsalarm gestartet wird
T_STMP	STRUCT	temp	Struktur für Uhrzeitangaben (Zeitstempel)
JAHR	BYTE	temp	
MONAT	BYTE	temp	
TAG	BYTE	temp	
STUNDE	BYTE	temp	
MINUTEN	BYTE	temp	
SEKUNDEN	BYTE	temp	
MSEC_WTAG	WORD	temp	
	END_STRUCT	temp	
	END_STRUCT	v	

Anweisungsteil

Im Anweisungsteil des OB 20 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 20)	Erläuterung
Network 1:	
SET	//Ausgang A 4.0 unbedingt setzen
= A 4.0	
Network 2:	
L AW 4	//Ausgangswort sofort aktualisieren
T PAW 4	
Network 3:	
L #STARTINFO.T_STMP.SEKUNDEN	//Sekunden aus Startereignisinformation
T MW 140	//lesen
L #STARTINFO.T_STMP.MSEC_WTAG	//Millisekunden und Wochentag aus
T MW 142	//Startereignisinformation lesen
L MD 140	
SRD 4	//Wochentag eliminieren und
T MD 140	//Millisekunden zurückschreiben
Network 4:	//(befinden sich nun BCD-codiert in
L #STARTINFO.SIGN	//MW 142).
T MW 144	//Startzeitpunkt des Verzögerungsalarms
Network 5:	//(=Aufruf des SFC 32) aus
CALL SFC 20	//Startereignisinformation lesen
SRCBLK := STARTINFO	//Startereignisinformation in Merker-
RET_VAL := MW 106	//bereich kopieren (MB 120 bis MB 139)
DSTBLK := P#M 120.0 BYTE 20	

A.4.7.3 OB 1

Deklarationsteil

Abweichend vom voreingestellten Deklarationsteil des OB 1 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	temp	Start-Info zu OB 1
E_ID	WORD	temp	Ereignis-ID
AE_NR	BYTE	temp	Ablaufebene
OB_NR	BYTE	temp	OB-Nr
DK 1	BYTE	temp	Datenkennung 1
DK 2	BYTE	temp	Datenkennung 2
AKT_ZYK	INT	temp	aktuelle Zykluszeit
MIN_ZYK	INT	temp	minimale Zykluszeit
MAX_ZYK	INT	temp	maximale Zykluszeit
T_STMP	STRUCT	temp	Struktur für Uhrzeitangaben (Zeitstempel)

Variablenname	Datentyp	Deklaration	Kommentar
JAHR	BYTE	temp	
MONAT	BYTE	temp	
TAG	BYTE	temp	
STUNDE	BYTE	temp	
MINUTEN	BYTE	temp	
SEKUNDEN	BYTE	temp	
MSEC_WTAG	WORD	temp	
	END_STRUCT	temp	
	END_STRUCT	temp	

Anweisungsteil

Im Anweisungsteil des OB 1 geben Sie folgendes AWL-Anwenderprogramm ein:

AWL (OB 1)	Erläuterung
Network 1:	
L #STARTINFO.T_STMP.SEKUNDEN	//Sekunden aus Startereignisinformation
T MW 10	//lesen
L #STARTINFO.T_STMP.MSEC_WTAG	//Millisekunden und Wochentag aus
T MW 12	//Startereignisinformation lesen
L MD 10	//Wochentag eliminieren und
SRD 4	//Millisekunden zurückschreiben
T MD 10	//(befinden sich nun BCD-codiert in
	//MW 12)
Network 2:	//Positive Flanke an Eingang E 0.0?
U E 0.0	
FP M 1.0	
= M 1.1	
Network 3:	//Wenn ja, Verzögerungsalarm starten
U M 1.1	//(Startzeitpunkt des Verzögerungsalarms
SPBNB m001	//dem Parameter SIGN zugewiesen)
CALL SFC 32	
OB_NR := 20	
DTIME := T#10S	
SIGN := MW 12	
RET_VAL:= MW 100	
m001: NOP 0	
Network 4:	//Status des Verzögerungsalarms abfragen
CALL SFC 34	//(SFC QRY_DINT)
OB_NR := 20	
RET_VAL:= MW 102	
STATUS := MW 4	
Network 5:	//Positive Flanke am Eingang E 0.1?
U E 0.1	
FP M 1.3	
= M 1.4	
Network 6:	//... und Verzögerungsalarm ist
U M 1.4	//aktiviert (Bit 2 des Verzögerungs-
U M 5.2	//alarm-STATUS)?
SPBNB m002	//Dann Verzögerungsalarm stornieren
CALL SFC 33	
OB_NR := 20	
RET_VAL:= MW 104	
m002: NOP 0	//Ausgang A 4.0 mit Eingang E 0.2
U E 0.2	//zurücksetzen
R A 4.0	

A.5 Zugriff auf Prozess- und Peripheriedatenbereiche

Dieses Kapitel beschreibt die Adressierung von Peripheriedatenbereichen (Nutzdaten, Diagnose- und Parameterdaten).

Weitere Informationen zu den Systemfunktionen, die in diesem Kapitel erwähnt werden, finden Sie im Referenzhandbuch *Systemsoftware für S7-300/400, System- und Standardfunktionen*.

A.5.1 Zugriff auf den Prozessdatenbereich

Die CPU kann entweder indirekt über das Prozessabbild oder über den Rückwand-/P-Bus direkt auf Ein- und Ausgänge zentraler und dezentraler Digitalein-/ausgabebaugruppen zugreifen.

Auf Ein- und Ausgänge zentraler und dezentraler Analogein-/ausgabebaugruppen greift die CPU direkt über den Rückwand-/P-Bus zu.

Baugruppenadressierung

Die Zuordnung zwischen Adressen, die im Anwenderprogramm verwendet werden, und den Baugruppen erfolgt durch die Konfigurierung der Baugruppen mit STEP 7 Lite.

- Bei zentraler Peripherie: Anordnung des Baugruppenträgers und Zuordnung der Baugruppen auf Steckplätzen in der Konfigurationstabelle.
- Bei dezentraler Peripherie (PROFIBUS-DP): Anordnung der DP-Slaves in der Konfigurationstabelle "Mastersystem" mit Vergabe der PROFIBUS-Adresse und Zuordnung der Baugruppen auf Steckplätze.

Das Konfigurieren der Baugruppen löst die Adresseinstellung der einzelnen Baugruppen über Schalter ab. Die CPU erhält vom PG als Ergebnis der Konfigurierung Daten, anhand derer sie die zugeordneten Baugruppen erkennt.

Peripherieadressierung

Für Ein- und Ausgänge existiert jeweils ein eigener Adressbereich. Deshalb muss die Adresse eines Peripheriebereichs außer der Byte- oder Wortangabe zusätzlich die Kennung E für die Eingänge und A für die Ausgänge beinhalten.

Nachfolgende Tabelle zeigt die zur Verfügung stehenden Peripherie-Adressbereiche.

Operandenbereich	Zugriff über Einheiten der folgenden Größe:	S7-Notation
Peripheriebereich: Eingänge	Peripherieeingangsbyte Peripherieeingangswort Peripherieeingangs-Doppelwort	PEB PEW PED
Peripheriebereich: Ausgänge	Peripherieausgangsbyte Peripherieausgangswort Peripherieausgangs-Doppelwort	PAB PAW PAD

Welche Adressbereiche bei den einzelnen Baugruppen möglich sind, entnehmen Sie den folgenden Handbüchern.

- Handbuch Automatisierungssystem S7-300, Aufbauen, CPU-Daten
- Referenzhandbuch Automatisierungssysteme S7-300, M7-300, Baugruppendaten

Baugruppen-Anfangsadresse

Die Baugruppen-Anfangsadresse ist die niedrigste Byte-Adresse einer Baugruppe. Sie stellt die Anfangsadresse des Nutzdatenbereichs der Baugruppe dar und wird in vielen Fällen stellvertretend für die ganze Baugruppe verwendet.

Sie wird z. B. bei Prozessalarmen, Diagnosealarmen, Ziehen/Stecken-Alarmen, und Stromversorgungsfehlern in die Startinformation des zugehörigen Organisationsbausteins eingetragen und identifiziert damit die alarmgebende Baugruppe.

A.5.2 Zugriff auf den Peripheriedatenbereich

Der Peripheriedatenbereich lässt sich unterteilen in:

- Nutzdaten und
- Diagnose- und Parameterdaten.

Beide Bereiche haben einen Eingangsbereich (nur lesender Zugriff möglich) und einen Ausgangsbereich (nur schreibender Zugriff möglich).

Nutzdaten

Adressiert werden die Nutzdaten über die Byteadresse (bei digitalen Signalbaugruppen) oder die Wortadresse (bei analogen Signalbaugruppen) des Ein- bzw. Ausgangsbereichs. Auf Nutzdaten zugreifen können Sie über Lade- und Transferbefehle, Kommunikationsfunktionen bzw. über den Prozessabbildtransfer. Nutzdaten können z. B. digitale und analoge Ein-, Ausgangssignale von Signalbaugruppen sein.

Bei der Übertragung von Nutzdaten kann eine Datenkonsistenz von maximal 4 Byte erreicht werden. Verwenden Sie die Anweisung "Transferiere Doppelwort", werden 4 Byte zusammenhängend und unverändert (konsistent) übertragen. Verwenden Sie vier einzelne Anweisungen "Transferiere Eingangsbyte", könnte an einer Befehlsgränze ein Prozessalarm-OB gestartet werden, der Daten an die gleiche Adresse überträgt und damit den Inhalt der ursprünglichen vier Byte verändert.

Diagnose- und Parameterdaten

Die Diagnose- und Parameterdaten einer Baugruppe können nicht einzeln adressiert werden, sondern nur zusammengefasst zu ganzen Datensätzen. Diagnose- und Parameterdaten werden grundsätzlich konsistent übertragen.

Adressiert werden die Diagnose- und Parameterdaten über die Anfangsadresse der betreffenden Baugruppe und die Datensatznummer. Datensätze werden unterteilt in Eingangs- und Ausgangsdatsätze. Eingangsdatsätze können nur gelesen, Ausgangsdatsätze können nur beschrieben werden. Auf Datensätze zugreifen können Sie mit Hilfe von Systemfunktionen oder Kommunikationsfunktionen (Bedienen und Beobachten, BuB). Nachfolgende Tabelle zeigt die Zuordnung der Datensätze zu Diagnose- und Parameterdaten.

Daten	Beschreibung
Diagnosedaten	Bei diagnosefähigen Baugruppen erhalten Sie beim Lesen der Datensätze 0 und 1 die Diagnosedaten dieser Baugruppe.
Parameterdaten	Bei parametrierbaren Baugruppen übertragen Sie beim Schreiben der Datensätze 0 und 1 die Parameter dieser Baugruppe.

Zugreifen auf Datensätze

Die Informationen in den Datensätzen einer Baugruppe können Sie nutzen, um parametrierbare Baugruppen nachzuparametrieren und Diagnoseinformationen diagnosefähiger Baugruppen auszulesen.

Nachfolgende Tabelle zeigt, mit welchen Systemfunktionen Sie auf Datensätze zugreifen können.

SFC	Anwendung
Baugruppen parametrieren	
SFC 55 WR_PARM	Übertragen der änderbaren Parameter (Datensatz 1) zur adressierten Signalbaugruppe.
SFC 56 WR_DPARM	Übertragen der Parameter aus den SDBs 100 bis 129 zur adressierten Signalbaugruppe.
SFC 57 PARM_MOD	Übertragen aller Parameter aus den SDBs 100 bis 129 zur adressierten Signalbaugruppe.
SFC 58 WR_REC	Übertragen eines beliebigen Datensatzes zur adressierten Signalbaugruppe
Diagnoseinformation auslesen	
SFC 59 RD_REC	Auslesen der Diagnosedaten

A.6 Einstellen des Betriebsverhaltens

A.6.1 Einstellen des Betriebsverhaltens

Dieses Kapitel erläutert, wie Sie die Eigenschaften von S7-Automatisierungssystemen, die nicht fest vorgegeben sind, durch Einstellen von Systemparametern oder Einsatz von Systemfunktionen SFC beeinflussen können.

Detaillierte Informationen zu den Baugruppenparametern finden Sie in den Handbüchern zu den jeweiligen Automatisierungssystem-Familien, z. B.

- Handbuch Automatisierungssystem S7-300, Aufbauen, CPU-Daten
- Referenzhandbuch Automatisierungssysteme S7-300, M7-300, Baugruppendaten

Alles über SFCs finden Sie im Referenzhandbuch *Systemsoftware für S7-300/400, System- und Standardfunktionen* oder in der Online-Hilfe zur Bibliothek *System Function Blocks*.

A.6.2 Ändern des Verhaltens und der Eigenschaften von Baugruppen

Defaulteinstellungen

Alle parametrierbaren Baugruppen des S7-Automatisierungssystems sind bei Lieferung auf Defaultwerte eingestellt, die für Standardanwendungen geeignet sind. Mit diesen Defaultwerten können Sie die Baugruppen direkt, ohne weitere Einstellungen, einsetzen. Die Defaultwerte entnehmen Sie den Baugruppen-Beschreibungen in den jeweiligen Handbüchern. Die Defaultwerte sind auch in den Parametrierdialogen von STEP 7 Lite berücksichtigt.

Welche Baugruppen können parametriert werden?

Sie können jedoch auch das Verhalten und die Eigenschaften der Baugruppen parametrieren und damit auf Ihre Erfordernisse und die Gegebenheiten Ihrer Anlage einstellen. Parametrierbare Baugruppen sind CPUs, FMs, CPs sowie einige Analogein-/ausgabebaugruppen und Digitaleingabebaugruppen.

Es gibt parametrierbare Baugruppen mit und ohne Pufferung.

Baugruppen ohne Pufferung müssen nach jedem Spannungsausfall erneut mit den entsprechenden Daten versorgt werden. Die Parameter dieser Baugruppen werden im remanenten Speicherbereich der CPU gespeichert (indirekte Parametrierung durch die CPU).

Einstellen und Laden der Parameter

Baugruppenparameter stellen Sie mit STEP 7 Lite ein. Beim Speichern der Hardwarekonfiguration, die auch die Parameter enthält, erzeugt STEP 7 Lite "Systemdatenbausteine", die mit dem Anwenderprogramm in die CPU geladen und von dort beim Anlauf in die zugehörigen Baugruppen übertragen werden. Da die Systemdatenbausteine die Hardwarekonfiguration repräsentieren, werden sie z. B. in der Projektübersicht als "Hardware" visualisiert.

Was kann parametrierbar werden?

Beispiele für parametrierbare CPU-Eigenschaften:

- Anlaufverhalten
- Zyklus
- MPI
- Diagnose
- Remanenz
- Taktmerker
- Alarmbehandlung
- On-Board-Peripherie
- Schutzstufe
- Echtzeituhr
- Asynchronfehler

Parametrieren mit SFCs

Außer der Parametrierung mit STEP 7 Lite gibt es noch die Möglichkeit, mit Hilfe von Systemfunktionen Baugruppenparameter vom Programm aus zu ändern. Nachfolgende Tabelle zeigt, welche SFCs im Anwenderprogramm genutzt werden können.

SFC	Anwendung
SFC 55 WR_PARM	Übertragen der änderbaren Parameter (Datensatz 1) zur adressierten Signalbaugruppe.
SFC 56 WR_DPARM	Übertragen der Parameter aus den zugehörigen SDBs zur adressierten Signalbaugruppe.
SFC 57 PARM_MOD	Übertragen aller Parameter aus den zugehörigen SDBs zur adressierten Signalbaugruppe.
SFC 58 WR_REC	Übertragen eines beliebigen Datensatzes zur adressierten Signalbaugruppe

Die Systemfunktionen sind ausführlich beschrieben im Referenzhandbuch *Systemsoftware für S7-300/400, System- und Standardfunktionen*.

Welche Baugruppenparameter dynamisch änderbar sind, entnehmen Sie bitte den Handbüchern zu den jeweiligen Baugruppen, z. B.

- Handbuch Automatisierungssystem S7-300, Aufbauen, CPU-Daten
- Referenzhandbuch Automatisierungssysteme S7-300, M7-300, Baugruppendaten

A.6.3 Nutzen der Uhrzeitfunktionen

Alle CPUs sind mit einer Uhr (Echtzeituhr oder Software-Uhr) ausgestattet. Die Uhr kann im Automatisierungssystem sowohl als Uhrzeitmaster als auch als Uhrzeitslave mit externer Synchronisation fungieren. Sie ermöglicht die Verwendung von Uhrzeitalarmen und Betriebsstundenzählern.

Uhrzeitformat

Die Uhr zeigt immer Uhrzeit (Mindestauflösung 1 s) und Datum mit Wochentag an. Bei einigen CPUs ist auch die Anzeige von Millisekunden möglich (siehe entsprechende Dokumentation zu den technischen Daten der CPUs).

Stellen und Lesen der Uhrzeit der CPU-Uhr

Sie stellen die Uhrzeit und das Datum

- aus dem Anwenderprogramm mit dem Aufruf der SFC 0 SET_CLK oder
- über das erweiterte CPU-Bedienpanel vom PG/PC aus und starten damit die Uhr.

Sie lesen die aktuelle Uhrzeit und das Datum

- aus dem Anwenderprogramm mit der SFC 1 READ_CLK oder
- über das erweiterte CPU-Bedienpanel vom PG/PC aus

Hinweis: Um Anzeigedifferenzen in HMI-Systemen in der Uhrzeit zu vermeiden, sollten Sie auf der CPU **Winterzeit** einstellen!

Parametrieren der Uhr

Ist in einem Netz mehr als eine Baugruppe mit Uhr vorhanden, müssen Sie durch Parametrieren mit STEP 7 Lite einstellen, welche CPU bei der Synchronisation der Uhrzeit als Master und welche als Slave fungieren soll. Durch Parametrierung stellen Sie außerdem ein, OB die Synchronisierung über den K-Bus oder über die MPI-Schnittstelle erfolgen soll und in welchen Intervallen die automatische Synchronisierung erfolgen soll.

Synchronisieren der Uhrzeit

Um sicherzustellen, dass die Uhrzeit aller Baugruppen im Netz übereinstimmt, werden die Slave-Uhren vom Systemprogramm in regelmäßigen (parametrierbaren) Abständen synchronisiert. Mit der Systemfunktion SFC 48 SNC_RTCB können Sie Datum und Uhrzeit von der Master-Uhr an die Slave-Uhren übertragen.

Einsetzen eines Betriebsstundenzählers

Ein Betriebsstundenzähler zählt die Einschaltzeiten eines angeschlossenen Betriebsmittels oder die Betriebsdauer der CPU als Summe der Betriebsstunden.

Im Betriebszustand STOP wird der Betriebsstundenzähler angehalten. Sein Wert bleibt auch nach Urlöschen erhalten. Bei Neustart (Warmstart) muss der Betriebsstundenzähler vom Anwenderprogramm wieder gestartet werden, bei Wiederanlauf läuft er automatisch weiter, wenn er vorher gestartet war.

Mit der SFC 2 SET_RTM können Sie den Betriebsstundenzähler auf einen Anfangswert setzen. Mit der SFC 3 CTRL_RTM können Sie den Betriebsstundenzähler starten oder stoppen. Mit der SFC 4 READ_RTM können Sie die aktuelle Anzahl der Betriebsstunden und den Zustand des Zählers ("gestoppt" oder "zählt") auslesen.

Eine CPU kann bis zu 8 Betriebsstundenzähler haben. Die Nummerierung beginnt bei 0.

A.6.4 Verwenden von Taktmerkern und Zeiten

Taktmerker

Ein Taktmerker ist ein Merker, der seinen Binärzustand periodisch im Puls-Pausen-Verhältnis 1:1 ändert. Welches Merkerbyte der CPU zum Taktmerkerbyte wird, bestimmen Sie bei der Parametrierung des Taktmerkers mit STEP 7 Lite.

Nutzen

Taktmerker können Sie im Anwenderprogramm verwenden, um z. B. Leuchtmelder mit Blinklicht anzusteuern oder periodisch wiederkehrende Vorgänge (etwa das Erfassen eines Istwertes) anzustoßen.

Mögliche Frequenzen

Jedem Bit des Taktmerkerbytes ist eine Frequenz zugeordnet. Nachfolgende Tabelle zeigt die Zuordnung.

Bit des Taktmerkerbytes	7	6	5	4	3	2	1	0
Periodendauer (s)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Frequenz (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

Achtung

Taktmerker laufen asynchron zum CPU-Zyklus, d. h. in langen Zyklen kann sich der Zustand des Taktmerkers mehrfach ändern.

Zeiten

Zeiten sind ein Speicherbereich des Systemspeichers. Die Funktion einer Zeit wird durch das Anwenderprogramm festgelegt (z. B. Einschaltverzögerung). Die Anzahl der verfügbaren Zeiten ist CPU-abhängig.

Achtung

- Wenn Sie in Ihrem Anwenderprogramm mehr Zeiten einsetzen, als die CPU zulässt, wird ein synchroner Fehler gemeldet und OB 121 gestartet.
 - Bei S7-300 können Zeiten nur im OB 1 und im OB 100 gleichzeitig gestartet und aktualisiert werden, in allen anderen OBs können Zeiten nur gestartet werden.
-

Index

*	
*.awl -Datei	8-4, 8-6, 8-7
*.k7e -Datei	8-2
*.k7p -Datei	8-2
*.sdf -Datei	8-2, 8-6, 8-7

A

Abfragen des Uhrzeitalarms	2-16
Abschaltung Endlage-Rechtslauf / Abschaltung Endlage-Linkslauf	5-20
Abschaltung mit Wiederanlauf (Autoreset)...	5-20
Abschaltung ohne Wiederanlauf	5-20
Absolute und symbolische Adressierung	6-1
Abzweige in FUP-Netzwerken erzeugen	6-67
Abzweige in KOP-Netzwerken erzeugen	6-62
ACCESS	8-8
ACT_TINT.....	2-15
Adressbelegung prüfen (PG/PC-Schnittstelle).....	1-19
Adressen.....	5-15, 6-8
in FUP-Elementen eingeben.....	6-65
in KOP-Elemente eingeben und editieren	6-56
in KOP-Elementen überschreiben.....	6-57
zuweisen	5-15
Adressierung	
absolut	6-1, 6-2
bereichsintern	A-53
bereichsübergreifend	A-53
speicherindirekt.....	A-53
symbolisch	6-1
Adressregister.....	6-103
Adressübersicht	5-15, 6-82
Aktivieren	6-12
der Anzeige von Symbolen im Baustein ..	6-12
Aktualisieren	
Bausteinanfrage.....	6-72
Prozessabbild	A-17
Aktualoperand siehe Aktualparameter	2-26
Aktualparameter....	2-26, 6-72, A-50, A-65, A-66,A-67, A-68, A-69
Aktualwert	6-75
Alarmarten	2-3
Alarmermetrieren	5-7
Alarmergebnisse	
sperrern und freigeben.....	A-75
verzögert bearbeiten	A-76
Alarmgesteuerte Programmbearbeitung	2-3
Alarm-OBs	

abwählen	2-5
Einsatz.....	2-14
parametrieren	2-3, 2-16

Ä

Ändern	
Betriebszustand.....	7-3
Datenwerte in der Datensicht von Datenbausteinen	6-79
Uhrzeit für Uhrzeitalarm.....	2-15
Ändern der Fensteranordnung	3-9
Ändern des Verhaltens und der Eigenschaften von Baugruppen	A-110
Änderungen übernehmen und speichern	4-7

A

Anfangsadresse	A-108
Anfangswert	6-39, 6-75, 6-77, 6-80
ANLAUF	A-4
abbrechen	A-4
Betriebszustand der CPU	A-1
CPU-Tätigkeiten	A-4
Anlauf-OB.....	2-21, A-4, A-7, A-9
Anlaufprogramm	2-22
Anlaufverhalten	5-7
Anlegen	4-4, 6-30
Bausteine	6-30
Datenbausteine (DB).....	6-30
Projekt	4-4
Anordnen.....	5-12, 5-13
Baugruppen im Baugruppenträger	5-12
Boxen	6-63, 6-64
C7-Komplettsystemen	5-13
Ansicht	
vergrößern.....	6-35
verkleinern.....	6-35
zoomen.....	6-36
Anweisungen.....	6-34
aus den Befehlsbibliotheken.....	6-34
eingeben	
Vorgehensweise.....	6-48
Anweisungsliste.....	6-23
Anweisungsteil	6-33, 6-38
Aufbau	6-47
bearbeiten	6-47
in AWL.....	6-37
Suchfunktion für Fehler	6-51
Anwenderdefinierte Datentypen (UDT)	6-24
Beschreibung	A-48
erstellen.....	A-48, A-49

- Struktur eingeben..... 6-78
 - Anwenderprogramm 9-10
 - Aufgaben 2-1
 - Elemente..... 2-2
 - im CPU-Speicher A-13
 - laden A-13, A-14
 - laden in CPU..... 9-3
 - laden in PG 9-10
 - Anwenderspeicher
 - komprimieren 9-15
 - ANY A-51, A-57, A-59
 - Parameter
 - Beschreibung und Verwendung A-60
 - Anzeige..... 5-13, 5-15, 6-82, 10-25
 - Adressübersicht 5-15, 6-82
 - Baugruppenzustand..... 11-2
 - Baustein-Längen..... 6-29
 - Betriebszustand 7-3
 - CPU-Betriebssystem-Version in der
 - Baugruppenliste 5-13
 - Datenstruktur von Datenbausteinen mit
 - zugeordnetem FB (Instanz-DB) 6-76
 - geforderter Werte 10-25
 - Querverweise zu Operanden mit
 - überlappenden Adressbereichen 6-95
 - Struktur von Datenbausteinen mit
 - zugeordnetem UDT..... 6-78
 - Symbole im Baustein aktivieren 6-12
 - Anzeigeformat festlegen 10-16
 - Anzeigegröße eines Fensters einstellen..... 6-19
 - Anzeigen
 - für den Programmstatus festlegen 10-30
 - im Programmstatus..... 10-28
 - Arbeiten 13-2
 - ohne Original-Projekt auf dem PG/PC..... 13-2
 - Arbeitsbereich..... 3-4, 5-2
 - für das Konfigurieren..... 5-2
 - Arbeitsfenster
 - Ansicht
 - verkleinern 6-35
 - zoomen 6-36
 - Arbeitsschritte für eine Automatisierungs-
 - lösung 1-1
 - Arbeitsspeicher 9-3, 9-4, A-12, A-13, A-14
 - ARRAY A-37, A-40
 - asynchrone Ereignisse..... 2-10
 - Asynchrone Fehler..... 2-24, 2-25, 11-23
 - Asynchronereignisse
 - sperrern und freigeben..... A-75
 - verzögert bearbeiten A-76
 - Aufbau 3-3, 5-4, 8-10, 11-4, 11-6
 - Anweisungsteil 6-47
 - der Benutzungsoberfläche 3-3
 - der Export-Datei..... 8-11
 - der Sicht 'HW-Diagnose' 11-6
 - der Sicht 'HW-Konfiguration' 5-4
 - der Sicht 'HW-Vergleich' 11-4
 - Online-Verbindungen 7-1
 - Querverweisliste 6-82, 6-83
 - UDT 6-24
 - Variablendeklarationstabelle 6-39, 6-40
 - Aufruf.....3-2, 11-7
 - Baugruppenzustand 11-7
 - Hilfefunktionen.....3-2
 - Aufrufhierarchie im Anwenderprogramm.....2-8
 - Aufrufumgebung von Bausteinen
 - festlegen..... 10-31
 - Aufspalten
 - einer Kreuzung in KOP-Netzwerken.....6-61
 - Aufteilung der Speicherbereiche A-12
 - Auftrennen
 - geschlossener Parallelzweige in
 - KOP-Netzwerken.....6-60
 - Aus den aktuellen Einstellungen eine
 - Dokumentationsvorlage erstellen 12-10
 - Ausgänge6-85
 - Prozessabbild..... A-17
 - Ausgangsadresse.....5-15
 - Ausgangsparameter
 - RET_VAL auswerten 11-22
 - Auskunft über Baugruppen..... 11-7
 - Auskunftsfunktionen 11-11
 - Auskunftsfunktionen des Baugruppen-
 - zustands 11-9
 - Ausschalten
 - des Tests mit Programmstatus..... 10-33
 - Statusleiste.....6-18
 - Symbolleiste6-18
 - Ausschneiden
 - markierte Bereiche in die
 - Zwischenablage 10-16
 - Auswählen
 - Programmiersprache6-21
 - Auswählen eines Stationstyps.....5-11
 - Auswerten
 - Ausgangsparameter RET_VAL 11-22
 - Auswertung des Diagnosepuffers..... A-21
 - Automation License Manager.....1-11
 - Autorisierung 1-16
 - Autorisierung installieren 1-13
 - Autorisierungsdiskette 1-13
 - Autorisierungsprogramm 1-13
 - AWL6-21, 6-23
 - Bausteine eingeben.....6-48
 - Einstellungen6-69
 - AWL (Anweisungsliste)..... 1-4
 - AWL-Anweisungen
 - eingeben.....6-70
 - Kommentare eingeben6-71
 - Regeln für die Eingabe6-69
 - Textbereiche markieren.....6-70
 - awl-Datei8-6
- ## B
- Baugruppen..... 5-7, 5-12, 5-16, 5-17, 9-11, 13-1
 - anordnen im Baugruppenträger.....5-12
 - Eigenschaften festlegen5-7
 - parametrieren A-110, A-111
 - spezifizieren9-11
 - tauschen.....5-16, 13-1
 - verschieben5-16

- Baugruppenadressierung.....A-107
 Baugruppen-AnfangsadresseA-108
 BaugruppenparameterA-111
 übertragen mit SFCs.....A-110
 übertragen mit STEP7Lite.....A-110
 Baugruppenstatus..... 11-6
 Baugruppentausch..... 5-12
 Baugruppenträgerausfall (OB 86) 11-33
 Baugruppentyp-abhängiger Umfang
 der Auskunftsfunktionen im
 Baugruppenzustand..... 11-11
 Baugruppenzustand..... 11-2, 11-7, 11-11
 aktualisieren..... 11-10
 anzeigen 11-2
 Auskunftsfunktionen..... 11-9
 Baustein 6-30
 anlegen 6-30
 Aufrufumgebung festlegen..... 10-31
 aus der B-Stack-Liste öffnen..... 11-15
 aus der U-Stack-Liste öffnen 11-15
 zum Verändern des PointersA-54
 Baustein zu einem Diagnosepuffereintrag
 öffnen 11-14
 Bausteinabhängigkeiten..... 6-88, 6-89, 6-90
 Bausteinaufrufe..... 2-8, 2-9
 aktualisieren..... 6-72
 Bausteinbibliotheken..... 6-32
 Bausteine1-6, 1-7, 1-8, 1-9, 2-2
 eingeben in AWL..... 6-48
 im Anwenderprogramm..... 2-2
 umverdrahten..... 6-51
 vergleichen 6-30
 Bausteine (geladene)
 auf integriertem EPROM speichern 9-6
 im PG/PC bearbeiten 9-10
 Bausteine in das PG laden 9-10
 Bausteine in die CPU laden 9-2
 Bausteineditor..... 1-8, 6-20
 Bausteineigenschaften..... 1-8, 6-25, 6-27, 6-33
 Baustein-Längen anzeigen 6-29
 einstellen..... 6-31
 Baustein-Instanzen 10-31
 Bausteincommentar 6-49, 6-50
 Baustein-Längen
 anzeigen 6-29
 Bausteinschutz 6-28
 Bausteinschutz setzen 6-27
 Baustein-Stack..... A-12, A-20
 Bausteinsymbole..... 6-10
 Bausteintitel 6-49
 Bausteinvergleich..... 6-30
 BCDA-34
 BCD-Format.....A-35
 Bearbeiten
 geladener Bausteine im PG/PC 9-10
 Projekt..... 4-6
 Bearbeiten einer geladenen Hardware-
 Konfiguration im PG/PC..... 9-11
 Bedienung über Tastatureingabe..... 3-14
 Bedienung über TeleService..... 3-20
 Befehle
 einfügen.....6-34
 Befehlsbibliotheken6-34
 Beispiel
 Datentyp STRING..... A-40
 Eingabe eines zusammenhängenden
 Operandenbereichs10-13
 Eingabe von Steuer-/Forcewerten10-13
 Gleitpunktzahlenformat..... A-31
 Maskieren und Demaskieren von
 Synchronfehlerereignissen A-71
 Sperrungen und Freigeben von Alarm- und
 Asynchronereignissen
 (SFC 39 und 40)..... A-75
 verzögerte Bearbeitung von Alarm-
 und Asynchronereignissen
 (SFC 41 und 42)..... A-76
 zur Hantierung mit Uhrzeitalarmen A-95
 Beispiel für die Eingabe von Operanden
 in Forcetabellen10-12
 Beispiel für die Eingabe von Operanden
 in Variablen Tabellen10-12
 Beispiel zum Arbeiten mit
 Verwendungsstellen6-92
 Beispielprogramm
 industrieller Mischprozess A-77
 BeispielprogrammeA-70
 BeispielprojekteA-71
 Beispieltext12-8
 Belegung des Prozessabbildes5-21
 Benutzungsoberfläche.....3-3
 Beobachten10-2
 prinzipielle Vorgehensweise10-2
 Beobachten von Variablen
 Einführung10-17
 einmalig und sofort10-19
 mit definiertem Trigger.....10-18
 Beobachten/Steuern.....1-7
 Beobachtmodus10-17
 Bereiche aus der Zwischenablage in die
 Variablen Tabelle einfügen10-16
 Besondere Regeln zur Platzhalter-
 baugruppe (DM 370 Dummy)5-9
 Betriebsart
 für den Test festlegen10-32
 Betriebsart Hand-Vor-Ort.....5-20
 Betriebsartenschalter A-10
 Betriebsstundenzähler..... A-113
 Betriebssystem1-16
 Aufgaben2-1
 Betriebssystem der CPU2-10
 Betriebsverhalten einstellen A-110
 Betriebszustand..... A-9, A-10, A-11
 ANLAUFA-1, A-3, A-6, A-7, A-8, A-9
 anzeigen und ändern.....7-3
 HALT A-11
 RUN..... A-1, A-2, A-3
 STOP.....A-1, A-2, A-3, A-4
 Stack-Inhalte11-12
 Betriebszustand
 RUN..... A-10

- Betriebszustand der CPU
 beim Laden ändern 9-7
- Betriebszustände
 der CPU A-1
 Priorität A-3
- Betriebszustände und Übergänge A-1
- Betriebszustandsübergänge A-2
- Bibliotheken 3-3, 3-4, 4-4, 6-34
- Binär-codierte Dezimalzahlen (BCD) A-34
- BLKMOV A-13
- BLOCK A-51
 Parametertyp A-50
- BLOCK_DB A-50
- BLOCK_FB A-51
- BLOCK_FC A-50
- BLOCK_SDB A-50
- Blockierschutz 5-18
- Blockierstrom 5-18
- Blockierzeit 5-18
- BM 147 5-10
- BOOL
 als Datentyp A-26
 Bereich A-27
- Boxen
 ändern 6-64
 entfernen 6-64
 platzieren 6-53, 6-63
- B-Stack
 im B-Stack gespeicherte Daten A-20
 verschachtelte Aufrufe A-20
- Byte
 als Datentyp A-26
 Bereich A-27
- C**
- C7-Komplettsysteme konfigurieren 5-13
- CAN_TINT 2-15
- Certificate of License 1-11, 1-13
- Codebaustein öffnen in 12-7
- Codebausteine
 Aufbau 6-33
 definieren
 Beispiel A-80
 im Bausteineditor 6-33
 Zeitstempel 6-98
- COUNTER A-50, A-51
 Parametertyp A-50, A-51
- CPU (Central Processing Unit)
 Betriebszustände A-1, A-2, A-3
- CPU 31x A-23
- CPU 31xC 5-14, 8-2, 8-5, A-25
- CPU laden mit einer Konfiguration 9-7
- CPU online 1-5
- CPU urlöschen 9-12
- CPU-Bedienpanel 3-3, 3-4
- CPU-Hardwarefehler (OB 84) 11-31
- CPU-Parameter "Zyklusbelastung durch
 Kommunikation" 2-10
- CPU-Schnittstellen parametrieren 5-7
- CREAT_DB A-13
- CRST/WRST A-4
- CTRL_RTM A-113
- D**
- Darstellung
 FUP-Elemente 6-62
 globale oder lokale Symbole 6-4
 KOP-Elemente 6-52
- Darstellung von Schnittstellen und
 Schnittstellenmodulen 5-12
- DATE_AND_TIME A-37
- DATE_AND_TIME (Datum und Uhrzeit)
 Bereich A-39
 Format A-38
- Dateiformat für den Import/Export einer
 Symboltabelle 8-8
- Datenaustausch
 in verschiedenen Betriebszuständen A-10
- Datenbaustein (DB) 2-2
 global 2-31
 Instanz-Datenbausteine 2-26, 2-29
 remanent A-23, A-24, A-25
 Struktur 2-31
- Datenbausteine
 Datensicht 6-75
 Datenstruktur mit zugeordnetem FB
 (Instanz-DB) eingeben/anzeigen 6-76
 Datenwerte auf die Anfangswerte
 rücksetzen 6-80
 Datenwerte in der Datensicht ändern 6-79
 Deklarationssicht 6-74
 Grundlagen 6-73
- Datenbausteine (DB)
 anlegen 6-30
- Datenbausteinregister A-20
- Datenelemente des Datentyps STRUCT
 in die Variablendeklarationstabelle
 eingeben 6-41
- Datensatz
 lesen A-109
 schreiben A-109
 zugreifen auf A-109, A-110
- Datensicht 6-73, 6-75, 6-79
 Datenbausteinen 6-75
- Datenträger 8-5
- Datentyp
 anwenderdefiniert 6-24
- ARRAY 6-42, A-40
- DATE AND TIME
 Datum und Uhrzeit A-38
- DWORD A-34
- REAL
 Gleitpunktzahl A-29, A-30
- S5TIME A-35
- STRING A-39
- STRUCT 6-41, A-41
- UDT 6-24
- WORD A-34
- Datentyp ARRAY in die
 Variablendeklarationstabelle eingeben 6-42

- Datentypen 6-8
 anwenderdefiniert A-37
 ANY A-57
 ARRAY A-37
 Beschreibung A-27
 BOOL A-26, A-27
 Byte A-26
 BYTE A-27
 DATE_AND_TIME A-37
 Datum A-27
 DINT
 Ganzzahl (32 Bit) A-28
 Doppelwort A-26
 Doppelwort (DWORD) A-27
 Einführung A-26
 elementar A-27
 FB
 SFB 2-26, A-37
 Ganzzahl (16 Bit) (INT) A-27
 Ganzzahl (32 Bit) (DINT) A-27
 INT
 Ganzzahl (16 Bit) A-28
 Parametertypen
 ANY
 Parameter A-60, A-61, A-62
 Realzahl (REAL) A-27
 S5TIME A-27
 STRING A-37
 STRUCT A-37
 UDT A-37
 Uhrzeit (TIME_OF_DAY) A-27
 Wort A-26
 Wort (WORD) A-27
 Zeichen (CHAR) A-27
 Zeit (TIME) A-27
 zusammengesetzt A-37
Datentypen (einfache)
 in die Variablendeklarationstabelle
 eingeben 6-41
Datentypen zu Lokaldaten von
 Codebausteinen zuordnen A-63
Datenwerte
 auf die Anfangswerte rücksetzen 6-80
 in der Datensicht von Datenbausteinen
 ändern 6-79
Datum 7-3
 einstellen 7-3
DB 2-2, 2-31, 2-32
DB-Register 6-103
Deaktivieren
 Uhrzeitalarm 2-15
Deckblatt 12-3
Defekt
 Betriebszustand der CPU A-1
Definieren
 Codebausteine A-80
 Symbole bei der Programmeingabe 6-12
Deinstallieren 1-21
 der Nutzungsberechtigung 1-15
Deklarationssicht 6-73, 6-74
 Datenbausteinen 6-74
Deklarationstabelle
 Datentyp ARRAY eingeben 6-42
 einfache Datentypen eingeben 6-41
 löschen von Variablen 6-44
 Multiinstanz eingeben 6-46
 Spaltenbreite variieren 6-44
 Spaltenbreiten einstellen 6-36
 Variablen kopieren 6-43
Deklarationstabelle (siehe
 Variablendeklarationstabelle) 6-33
Deklarationstyp
 ändern 6-40
Deklarieren von lokalen Variablen
 FB für das Beispiel eines industriellen
 Mischprozesses A-83
 OB für das Beispiel eines industriellen
 Mischprozesses A-89
Deklarieren von Parametern
 FC für das Beispiel eines industriellen
 Mischprozesses A-87
Demaskieren
 Startereignisse 2-25
Demaskieren von Synchronfehler-
 ereignissen
 Beispiel A-71, A-72
Diagnose 11-2, 11-5, 11-6, 11-7
 Hardware 11-2, 11-6
 von Baugruppen 11-7
Diagnosealarm 11-9, 11-11, 11-30, 11-31
Diagnosealarm (OB 82) 11-30
Diagnosedaten 11-1, 11-19
 auf Baugruppen 11-19
Diagnoseereignis 11-1, A-21
Diagnosefunktionen 11-1
Diagnosemeldung
 an Teilnehmer senden 11-20
 eigene schreiben 11-20
Diagnosepuffer 11-9, A-21, A-22, A-23
 auslesen 11-17, A-23
 auswerten A-21
 Definition A-21
 Inhalt 11-1, A-21, A-22, A-23
Diagnosepuffereintrag 11-14
Diagnosezustandsdaten 11-19
Dialogfeld 'Baugruppenzustand' 11-7
Dialogfelder 3-7
Digital-Simulationsbaugruppe
 SIM 374 IN/OUT 16 5-9
DIN EN 6.1131-3 1-4
DINT
 Datentyp A-28
Direkthilfe 3-2
DIS_AIRT 2-25
DIS_IRT 2-24
DM 370 Dummy 5-9
DMSK_FLT 2-24
Dokumentation 1-7, 4-10, 12-14
 eines ganzen Projekts drucken 12-14
 von Projektbestandteilen drucken 12-14
Dokumentation zum Softwareprodukt
 STEP 7 Lite 1-10

- Dokumentationsvorlage 12-1
Dokumentationsvorlage verwenden..... 12-11
Dokumentationsvorlagen 12-10
Doppelwort (DWORD)
 Bereich.....A-27
 DatentypA-26
Download einer Konfiguration.....9-7
DP-Schnittstelle der CPU..... 5-7
Drucken
 Bausteine 12-14
 Diagnosepufferinhalt 12-14
 Globaldatentabelle 12-14
 Konfigurationstabelle 12-14
 Projektbestandteile 12-14
 Projektdokumentation 12-14
 Referenzdaten 12-14
 Symboltabelle 12-14
 Variablentabelle 12-14
Druckliste 12-3, 12-7
Druckobjekt..... 12-1, 12-7, 12-8, 12-9, 12-10
Druckobjekte 12-3, 12-5, 12-6
Druckobjekte umsortieren..... 12-3
Druckobjekte verschieben..... 12-3
Dummy-Baugruppe (DM 370 Dummy)..... 5-9
DWORD
 DatentypA-34
- ## E
- Editieren
 Adressen oder Parameter in
 KOP-Elementen 6-56
 in der Symboltabelle 6-12
Editor 6-20
Eigene Diagnosemeldungen senden 11-20
Eigenschaften 6-25, 6-28, 6-31
Ein-/Ausgangsadresse..... 5-15
 zuweisen 5-15
Einfügemenü..... 12-9, 12-10
Einfügen..... 6-11
 Bereich zusammenhängender Operanden
 in eine Variablentabelle..... 10-9
 Bereiche aus der Zwischenablage in die
 Variablentabelle 10-16
 Ersatzwerte bei Fehlererkennung 11-26
 Leerzeilen in Variablendeklarations-
 tabellen 6-41
 Programm 4-4
 Symbole 6-11
 Symbolzeilen..... 6-15
 weitere AWL-Netzwerke 6-71
 weitere FUP-Netzwerke 6-67
 weitere KOP-Netzwerke..... 6-58
Einfügen eines zusammenhängenden
 Operandenbereichs in eine Forcetable . 10-9
Einfügen von Bereichen aus der
 Zwischenablage in die Variablentabelle
 bzw. Forcetable 10-16
Einführung zu Daten- und Parametertypen..A-26
Einführung zum Forcen von Variablen..... 10-24
Einführung zum Konfigurieren der Hardware. 5-1
Einführung zum Testen mit Variablentabellen
 und Forcetablellen..... 10-1
Eingabe3-6, 10-8
 in Dialogfelder3-6
 Operanden oder Symbole in eine
 Variablentabelle..... 10-6
 Operanden oder Symbolen in eine
 Forcetablelle..... 10-7
Eingabemöglichkeiten von
 globalen Symbolen6-10
Eingänge6-85
 Prozessabbild A-17
Eingangsadresse.....5-15
Eingeben
 Adressen oder Parameter6-65
 Adressen oder Parameter in
 KOP-Elementen6-56
 AWL-Anweisungen.....6-70
 Baustein-/Netzwerkcommentare6-50
 Datenelemente des Datentyps
 STRUCT in die
 Variablendeklarationstabelle6-41
 Datenstruktur von Datenbausteinen
 mit zugeordnetem FB (Instanz-DB)6-76
 Datenstruktur von globalen
 Datenbausteinen6-76
 Datentyp ARRAY in die
 Variablendeklarationstabelle6-42
 einfache Datentypen in die
 Variablendeklarationstabelle6-41
 einzelner globaler Symbole im Dialog6-12
 FUP-Elemente.....6-65
 globale Symbole in ein Programm.....6-49
 Kommentare in AWL-Anweisungen.....6-71
 KOP-Elemente6-56
 Multiinstanz in der
 Variablendeklarationstabelle6-46
 Struktur von anwenderdefinierten
 Datentypen (UDT)6-78
 Struktur von Datenbausteinen mit
 zugeordnetem UDT6-78
 von Symbolen.....6-13
Eingeben mehrerer globaler Symbole
 in der Symboltabelle6-13
Eingrenzen von Störungen..... 11-2
Einschalten
 Statusleiste.....6-18
 Symbolleiste6-18
 Test mit Programmstatus 10-33
Einstellen.....6-4
 Anzeigegröße eines Fensters.....6-19
 Aufrufumgebung des Bausteins 10-31
 Bausteineigenschaften6-31
 Betriebsverhalten A-110
 Fensterteilung eines Arbeitsfensters6-36
 Operandenvorrang (absolut/symbolisch)....6-4
 PG/PC-Schnittstelle.....1-19
 Sortiermodus in der Symboltabelle.....6-17
 Spaltenbreiten bei einer
 Deklarationstabelle6-36
 Uhrzeit und Datum.....7-3

Einstellungen	6-52
Bausteineditor	6-34
für die Programmiersprache AWL	6-69
für die Programmiersprache FUP	6-62
für die Programmiersprache KOP	6-52
Einzelne Objekte drucken	12-14
Einzelne Symbole definieren	6-14
Elementare Datentypen	A-27
Elemente in Dialogfeldern	3-6
EM 300	5-10
EN/ENO	
Beschaltung	6-64
EN_AIRT	2-24
EN_IRT	2-24
Entstehung von Lücken im Anwender- speicher (RAM)	9-15
Entwurfsmethoden	
entwerfen eines strukturierten Programms	A-80
EPROM	9-6, A-23, A-24
EPROM-Bereich	A-13
Ereignisse	
asynchron	2-10
Ermitteln gestörter Baugruppen	11-5
Erreichbare Teilnehmer	7-1
Ersatzwert	
Verwenden von SFC 44 (RPL_VAL)	11-26
Erstellen	
eines anwenderdefinierten Datentyps	A-48
FB für den Motor	A-83, A-84, A-85
FC für die Ventile	A-87, A-88
Felder	A-43, A-44, A-45
Forceauftrag	10-26
OB 1 für das Beispiel eines industriellen Mischprozesses	A-89
parallele Verzweigungen in KOP-Netzwerken	6-59
Software im Projekt (prinzipiell)	4-10
Struktur	A-46, A-47
Variablentabelle	10-3
von Anwenderprogrammen	6-33
Erstellen und Öffnen einer Forcetable	10-3
Erstellen und Öffnen einer Variablentabelle	10-3
Erzeugen	
Abzweige in FUP-Netzwerken	6-67
Abzweige in KOP-Netzwerken	6-62
einer geschlossenen Verzweigung in KOP-Netzwerken	6-59
neue Zweige in KOP-Netzwerken	6-59
Verbindungen in FUP-Netzwerke	6-67
ET 200S	5-10
ET 200X	5-10
Export-Datei	8-10
Exportieren	8-1, 8-12
mehrsprachig verwaltete Texte	8-12
Symboltabelle	6-14
Exportieren von Texten für die Übersetzung	8-8
F	
F1	3-2
FB	2-2, 2-26, 2-27, 2-28, A-37
FC	2-2, 2-26
FC 12	A-97
Fehler	
während der Installation	1-17
Fehler vermeiden beim Aufruf von Bausteinen	6-101
Fehlerbehebung	
Programmbeispiele	11-23
Fehlererkennung	
OB-Typen	
OB 81	11-24, 11-25
Programmbeispiele	
Ersatzwerte	11-26
verwenden von Fehler-OBs um auf Fehler zu reagieren	2-24
Fehler-OB	11-23, 11-24, 11-25
OB-Typen	
OB121 und OB122	2-24
OB70 und OB72	2-24
OB80 bis OB87	2-24
verwenden von Fehler-OBs zur Reaktion auf Ereignisse	2-24
Fehler-OBs als Reaktion auf die Erkennung eines Fehlers	11-23
Fehlersuche	11-2
in Bausteinen	6-51
Feld (Datentyp ARRAY)	
Anzahl verschachtelter Ebenen	A-42
Beschreibung	A-43
Felder	
erstellen	A-43, A-44, A-45
für den Datenzugriff verwenden	A-43
Fenster	
Anzeigegröße einstellen	6-19
Fenster "Hardware Katalog" hantieren	5-16
Fensteranordnung	3-9, 3-10
speichern	3-10
wiederherstellen	3-10
Fensteranordnung ändern	3-9
Fensterinhalt	3-9
Fensterteilung	
eines Arbeitsfensters einstellen	6-36
Fenstertypen umschalten	3-19
FEPROM	A-23
Festlegen	
Anzeige für den Programmstatus	10-30
Anzeigeformat	10-16
Beobachtmodus	10-20
Betriebsart für den Test	10-32
Festlegen der Eigenschaften von Baugruppen/Schnittstellen	5-14
Festlegen der Eigenschaften von Komponenten	5-7
Feststellbare Fehler	11-23

- Filtern
 der Symboltabelle 6-16
 von Symbolen 6-13
 Firmware-Version..... 5-13
 Flash-File-System 1-17
 Forceauftrag 10-24
 erstellen 10-25
 Forcen..... 1-7, 10-2, 10-26
 Forcen von Variablen..... 10-24
 Einführung 10-24
 Sicherheitsvorkehrungen 10-25
 Forcetable 10-2, 10-7, 10-8, 10-9
 Forcewerte 10-24
 Beispiele für die Eingabe 10-13
 Formaloperand siehe Formalparameter..... 2-26
 Formalparameter 2-26, 2-27, 6-72,
 A-50, A-66, A-67, A-68, A-69
 Format
 BLOCK.....A-51
 COUNTERA-51
 Datentyp ARRAYA-40
 Datentyp DATE_AND_TIME
 (Datum und Uhrzeit).....A-38
 Datentyp DINT (32-Bit-Ganzzahlen)A-28
 Datentyp INT (16-Bit-Ganzzahlen).....A-28
 Datentyp REAL (Gleitpunktzahlen)A-29
 Datentyp S5TIME (Zeitdauer)A-35
 Datentyp STRINGA-39
 Datentyp STRUCTA-41
 Datentypen WORD und DWORD
 bei binär-codierten Dezimalzahlen.....A-34
 Parametertyp ANYA-57
 Parametertyp POINTERA-52
 Parametertypen BLOCK
 COUNTER und TIMERA-51
 TIMER.....A-51
 Format des Datentyps DWORD.....A-33
 Format des Datentyps TIMEA-36
 Format des Datentyps WORDA-33
 Fragezeichen 6-90
 Freigeben von Alarm- und
 Asynchronereignissen
 BeispielA-75
 Funktion (FC)..... 2-2, 2-26
 Anwendungsbereich 2-26
 Erstellen
 Beispiel-FC für das Beispiel eines
 industriellen MischprozessesA-87
 Funktionen (FC)..... 2-26
 Funktionsbausteine (FB)..... 2-26
 Funktionsbausteine (FBs) 2-2, 2-26
 Aktualparameter..... 2-27, 2-28
 Anwendungsbereich 2-27
 Erstellen
 Beispiel-FB für das Beispiel eines
 industriellen MischprozessesA-83
 Funktionsplan 6-23
 FUP..... 6-21, 6-23
 FUP (Funktionsplan)..... 1-4
 FUP-Anweisungen
 Regeln für die Eingabe 6-63
 FUP-Elemente
 Adressen oder Parameter eingeben.....6-65
 Darstellung6-62
 eingeben.....6-65
 überschreiben.....6-66
 FUP-Layout6-62
 FUP-Netzwerke
 Abzweige erzeugen.....6-67
 markieren in.....6-66
 Verbindungen erzeugen6-67
 Verbindungen trennen und
 neu verbinden.....6-68
 Für STEP7.....8-6
 Fußzeile.....12-1
 Fußzeilen.....12-9
- ## G
- Ganzzahl (16 Bit)
 Format..... A-28
 Ganzzahl (16 Bit) (INT)
 Bereich A-27
 Ganzzahl (32 Bit)
 Format..... A-28
 Ganzzahl (32 Bit) (DINT)
 Bereich A-27
 Geladene Bausteine auf integriertem
 EPROM speichern.....9-6
 Geladene Bausteine im PG/PC bearbeiten ..9-10
 Geschlossene Parallelzweige in
 KOP-Netzwerken auftrennen.....6-60
 Geschlossene Verzweigung in
 KOP-Netzwerk erzeugen.....6-59
 Gestörte Baugruppen ermitteln 11-5
 Gleitpunktzahl
 Beispiel..... A-31
 Grundelemente..... A-29
 Komponentenfelder A-29
 Parameter..... A-29
 Gleitpunktzahlen
 Format..... A-30
 Globale Datenbausteine
 Datenstruktur eingeben6-76
 Zeitstempel.....6-99
 Globale Datenbausteine (DB).....2-31
 Globale Symbole
 einzeln eingeben im Dialog6-12
 in ein Programm eingeben6-49
 mehrere eingeben in der Symboltabelle ..6-13
 Globale und lokale Symbole.....6-3
 Grundlagen
 zu Datenbausteinen.....6-73
 Grundsätzliche Bedienung beim
 Hardware konfigurieren5-2
 Grundsätzliche Schritte beim Konfigurieren
 einer Station5-3

- H**
- HALT..... A-1, A-2, A-3, A-11
 - Hardware 1-6, 11-4, 11-6, 11-7
 - diagnostizieren..... 11-2, 11-6, 11-7
 - konfigurieren 4-10, 5-1, 5-4
 - vergleichen 11-4
 - Hardware Katalog 5-2
 - Hardware-Konfiguration 9-2, 9-10
 - in das PG laden 9-10
 - in die CPU laden 9-2
 - Hintergrund-OB
 - Priorität 2-23
 - programmieren..... 2-23
 - Hinweise zur Änderung von
 - Register-Inhalten..... 6-103
 - Hochformat 12-1
 - HW-Diagnose 1-6, 11-6, 11-7
 - HW-Konfiguration..... 1-6, 5-4
 - HW-Vergleich..... 1-6, 11-2, 11-4
- I**
- Icons 3-5
 - Icons (siehe Symbole) 1-1, 1-5, 11-2
 - Ident-Nummer eintragen 1-17
 - Importieren..... 6-14, 8-1, 8-6, 8-7, 8-13
 - externe Quelle..... 4-4
 - mehrsprachig verwaltete Texte 8-13
 - Symboltabelle 6-14
 - Importieren von Texten für die Übersetzung.. 8-8
 - In_out-Parameter A-70
 - Indirekte Parametrierung A-110
 - Inhaltsverzeichnis 12-3
 - Installationsvoraussetzungen..... 1-16
 - Installieren des
 - Automation License Managers..... 1-13
 - Installieren von STEP 7 Lite..... 1-16
 - Instanz 2-29
 - Instanz-Datenbaustein 2-29, 2-30
 - mehrere Instanzen für einen
 - FB erstellen..... 2-26
 - remanent..... A-23
 - Instanz-Datenbausteine 2-29
 - Zeitstempel 6-99
 - INT
 - Datentyp A-28
 - Interruptbelegung
 - prüfen..... 1-19, 1-20
- K**
- k7e 8-2
 - k7p 8-2
 - Kaltstart..... 2-21, 2-22, A-1
 - Kennzeichnung
 - von Symbolen 6-4
 - Kombinationsfeld
 - Definition 3-6
 - Kommentare
 - zu Bausteinen..... 6-49
 - zu Netzwerken..... 6-49
 - Kommentare anzeigen 12-7
 - Kommentare in AWL-Anweisungen
 - eingeben..... 6-71
 - Kommentarzeichen 10-6
 - Kommentarzeile 10-6
 - Kommunikation..... 11-10
 - Kommunikationsfehler (OB 87) 11-34
 - Kommunikationsfehler-OB..... 11-34
 - Kommunikationslast 2-10, 2-13, 2-14
 - Kommunikationsprozesse 2-10
 - Kompakt-CPU A-23
 - Kompakt-CPUs..... 5-14
 - Komprimieren
 - des Anwenderspeichers 9-15
 - Komprimieren des Anwenderspeichers (RAM) nach mehrfachen Löschen
 - oder Nachladen 9-15
 - Konfiguration 5-22
 - speichern 5-22
 - Konfigurationstabelle 5-6
 - Konfigurieren 5-1, 5-4, 5-13, 9-8
 - C7-Komplettsysteme 5-13
 - der Hardware..... 5-4
 - Hardware 4-10
 - wann erforderlich? 5-1
 - Konfigurieren der Hardware (prinzipiell) 4-10
 - Konfigurieren von ET 200S 5-10
 - Konsistenz einer Stationskonfiguration
 - prüfen 9-7
 - Konsistenzprüfung..... 5-22
 - Kontaktplan 6-22
 - Kontextabhängige Hilfe 3-2
 - Kontrolle der Zykluszeiten zur Vermeidung
 - von Zeitfehlern..... 11-16
 - KOP 6-21, 6-22
 - unzulässige Verschaltungen..... 6-55
 - KOP (Kontaktplan) 1-4
 - KOP-Anweisungen
 - Regeln für die Eingabe 6-53
 - KOP-Elemente 6-53, 6-54
 - Adressen eingeben und editieren 6-56
 - Adressen überschreiben..... 6-57
 - Darstellung 6-52
 - eingeben..... 6-56
 - Parameter eingeben und editieren 6-56
 - Parameter überschreiben 6-57
 - überschreiben..... 6-57
 - Kopfzeile..... 12-1, 12-9
 - Kopieren 4-6
 - markierte Bereiche in die
 - Zwischenablage 10-16
 - Projekt 4-6, 4-9
 - Projektteil..... 4-9
 - Symbolzeilen in die Zwischenablage..... 6-17
 - Variablen in Deklarationstabellen 6-43
 - Kopieren / Duplizieren von Forcetabellen..... 10-5
 - Kopieren / Duplizieren
 - von Variablen tabellen 10-4

KOP-Layout	6-52
KOP-Netzwerke	
Abzweige erzeugen	6-62
geschlossene Parallelzweige auftrennen	6-60
geschlossene Verzweigung erzeugen	6-59
Kreuzung aufspalten	6-61
markieren in	6-58
neue Zweige erzeugen	6-59
parallele Verzweigungen erstellen	6-59
Kreuzung in KOP-Netzwerken aufspalten....	6-61
Kurzinfo.....	3-2
Kurzschluss	6-55

L

Lade- und Arbeitsspeicher	A-13
Lade- und Arbeitsspeicher in der CPU.....	9-3
Lade-/Arbeitsspeicher löschen.....	9-12
Lademöglichkeiten	A-13
Lademöglichkeiten abhängig vom Ladespeicher.....	9-4
Laden	
Anwenderprogramm	A-13
Anwenderprogramme in CPU	9-3
Voraussetzungen	9-1
Laden aus der CPU in das PG.....	9-8, 9-10
Laden der Hardware-Konfiguration in die CPU.....	9-2
Laden einer Konfiguration in ein Zielsystem ..	9-7
Laden in die CPU.....	9-2
Laden in PG	9-8, 9-11
Laden von Bausteinen in die CPU	9-2
Ladespeicher	9-3, 9-4, A-12, A-13, A-14
nicht ablaufrelevante DBs speichern.....	A-13
Layout.....	6-52, 6-62
Leerzeilen	
in Variablendeklarationstabellen einfügen	6-41
Leistungsdaten	11-9
Leitfaden durch STEP 7 Lite	1-1
License Key	1-11, 1-12, 1-15
License Manager	1-11, 1-12
Linear programmieren.....	2-3
Listenfeld	3-6
Lizenz	1-12, 1-13
Lizenz-Typen	1-13
Enterprise License	1-11
Floating License.....	1-13
Rental License	1-11
Trial License.....	1-13
Upgrade License.....	1-13
Lokaldatenbedarf	6-86
Lokaldatenbereiche.....	5-7
Lokaldaten-Stack	A-12, A-18
Löschen	
Lade-/Arbeitsspeicher	9-12
Objekte	3-13
Symbolzeilen.....	6-15
Variablen in Deklarationstabellen.....	6-44
Löschen der Memory Card in der CPU.....	9-14
Löschen eine Projekts.....	4-10

Löschen einer Sprache.....	8-13
Löschen einzelner Bausteine in der CPU.....	9-13
Löschen und Umbenennen eines Projekts...4-10	
L-Stack	
Bearbeitung von Daten in einem verschachtelten Aufruf.....	A-20
speichern von temporären Variablen.....	2-26
überschreiben des L-Stack.....	A-18
zuordnen von Speicher zu lokalen Variablen	A-18

M

Markieren	
in FUP-Netzwerken	6-66
in KOP-Netzwerken.....	6-58
Symbolzeilen	6-17
Textbereiche in AWL-Anweisungen	6-70
Markieren von Zeilen in der Konfigurationstabelle (HW konfigurieren) ..	5-16
Markierte Bereiche in die Zwischenablage ausschneiden	10-16
Markierte Bereiche in die Zwischenablage kopieren.....	10-16
Maskieren	
Startereignisse	2-24
Maskieren von Synchronfehlerereignissen Beispiel.....	A-71
Maßnahmen im Programm zur Störungsbehandlung	11-21
Maximalzykluszeit	2-10
Mehrdeutige Symbole	6-9
Mehrdeutigkeit.....	6-9
Memory Card.....	9-6, A-14
parametrieren	1-17
Menüleiste	3-3, 3-4
Merker	6-84, 6-85
Speicherbereich	
remanent	A-23
Micro Memory Card (MMC)	8-2, 8-5, A-25
Micro Memory Card als Datenträger verwenden.....	8-5
Microsoft Windows	1-16
Mindestzykluszeit	2-11, 2-12
Mischprozess	A-77
MMC.....	8-2, 8-4, 8-5
Mnemonic.....	3-9
einstellen	6-69
Motor-RECHTS / Motor-LINKS.....	5-20
MPI-ISA-Card (Auto)	1-19
MPI-Karte im PG/PC	1-19
MPI-Schnittstelle	1-16
MPI-Schnittstelle der CPU.....	5-7
MSK_FLT	2-24
Multiinstanz	2-26, 2-29
in der Variablendeklarationstabelle eingeben.....	6-46
Multiinstanzen	
Regeln	6-46
Verwendung	6-45

N

Nachladen von Bausteinen in die CPU	9-5
Netzausfall	A-5
Netzwerk	6-58
einfügen	6-58
Netzwerke	6-23
Abschluss in KOP	6-53
Netzwerkcommentar	6-49
Netzwerkcommentare	
eingeben	6-50
Netzwerktitle	6-49, 6-50
Neue Schriftart Formatvorlage erstellen	12-12
Neue Seitenlayout Formatvorlage	
erstellen	12-13
Neue Zweige in KOP-Netzwerk erzeugen ...	6-59
Neues Projekt	1-5
Neustart . 2-21, 2-22, 9-1, A-1, A-4, A-5, A-6, A-7	
abbrechen	A-7
automatisch	A-4
ungepuffert	A-4
manuell	A-4
nicht proportionale Schrift	12-7
Nicht verwendete Symbole	6-16
Nichtflüchtiger RAM	A-23
Non-Retain	A-23
Notstart	5-20
Nullpointer	A-57
Nullstromerkennung	5-18
Nutzdaten	A-108
Nutzen der Uhrzeitfunktionen	A-112
Nutzungsberechtigung durch den	
Automation License Manager	1-11
NVRAM	A-23, A-24

O

OB	2-2, 2-3, 2-4, 2-5, 2-6, 2-7
OB 1	A-105, A-106
OB 1 und OB 80	A-101
OB 10	A-99, A-100
OB 100	A-4
OB 101	A-4
OB 102	A-4
OB 122	11-36
OB 1-Zykluszeit	2-10
OB 20	A-104, A-105
OB 80	11-29
OB 81	11-29, 11-30
OB 82	11-30, 11-31
OB 84	11-31
OB 85	11-32, A-18
OB 86	
Baugruppenträgerausfall	11-33
OB 87	11-34
Obergrenzen für die Eingabe von Zählern .	10-11
Obergrenzen für die Eingabe von Zeiten ...	10-10
Objekt	
löschen	3-13
umbenennen	3-13
verschieben	3-13

Offline-Konfiguration	11-3
Öffnen	4-6
Baustein aus der U-Stack-Liste	11-15
des Bausteins aus der B-Stack-Liste	11-15
des Bausteins zu einem	
Diagnosepuffereintrag	11-14
Projekt	4-6
Symboltabelle	6-14
Variablentabelle	10-3
von Symboltabellen	6-13
Online-Hilfe	
aufrufen	3-2
Themen	3-2
Online-Konfiguration	11-3
Online-Sicht auf die CPU	1-5
Online-Verbindungen	
aufbauen	7-1
Operand	10-2, 10-8
Operanden	
in eine Variablentabelle eingeben	10-6
umverdrahten	6-51, 6-52
Operanden ohne Symbol	6-17
Operandenbereiche	
Beschreibung	A-15
Operandenfeldbreite	6-52, 6-62
Operandenvorrang	6-4, 6-5
Optimieren der Vorlage für die	
Übersetzung	8-14
Optimierung des Übersetzungsvorgangs	8-15
Optionen	12-7
Optionen Schriftart und Seitenlayout	
festlegen	12-7
Optionspaket	3-20, 10-1
Organisationsbaustein (OB)	
Hintergrund-OB (OB 90)	2-3, 2-23
Organisationsbaustein für die	
Programmbearbeitung im Hintergrund	
(OB 90)	2-23
Organisationsbaustein für zyklische	
Programmbearbeitung (OB1)	2-10
Organisationsbausteine	2-2
Definition	2-3
Erstellen eines OB für das Beispiel eines	
industriellen Mischprozesses	A-89
Fehlererkennung	
OB 122	
Ersatzwerte	11-26
Prioritätsklassen	2-3, 2-4, 2-5
reagieren auf Fehler	2-24
Startinformation	2-4
Organisationsbausteine für alarmgesteuerte	
Programmbearbeitung	2-14
Organisationsbausteine für den Anlauf	
(OB100/OB102)	2-21
Organisationsbausteine für den Prozessalarm	
(OB 40 bis OB 47)	2-20
Organisationsbausteine für den Uhrzeitalarm	
(OB 10 bis OB 17)	2-15
Organisationsbausteine für den	
Verzögerungsalarm (OB 20 bis OB 23)	2-18

- Organisationsbausteine für den Weckalarm
(OB 30 bis OB 38)..... 2-19
- Organisationsbausteine für die
Fehlerbearbeitung (OB80bisOB87 /
OB 121 bis OB 122)..... 2-24
- Organisationsbausteine und
Programmstruktur 2-3
- P**
- Packen (ET 200S)..... 5-10
- Parallele Verzweigungen in
KOP-Netzwerken erstellen..... 6-59
- Parallelzweige (geschlossene) in
KOP-Netzwerken auftrennen 6-60
- Parameter
in KOP-Elemente eingeben
und editieren 6-56
- in KOP-Elementen überschreiben..... 6-57
- Parameter in FUP-Elementen eingeben 6-65
- Parametertyp
ANY A-50
- BLOCK_DB A-51
- BLOCK_FB A-50
- BLOCK_FC A-51
- BLOCK_SDB A-51
- COUNTER A-51
- POINTER A-51
- TIMER A-50
- Parametertyp POINTER verwenden A-53
- Parametertypen A-50
- ANY A-60, A-61, A-62, A-63
- Einführung A-26
- Parametervergleich (Hardware)..... 11-2
- Parametrierbare Baugruppen A-110
- Parametrieren 5-1
- indirekt A-110
- mit SFCs A-111
- mit STEP7Lite A-110
- prozessalarmfähige Signalbaugruppen.... 2-20
- Uhr A-112
- Parametrieren der PG/PC-Schnittstelle 1-19
- Parametrierung im Anwenderprogramm 5-7
- Parametrierung von Vergleichsstellen 5-10
- PARAM_MOD A-109, A-111
- Passwort 7-2
- Passwort vergeben 5-7
- Passwortschutz für Zugriff auf Zielsysteme ... 7-2
- Peripherie
Adressbereiche A-107, A-108
- Peripherieausgänge im STOP-Zustand
der CPU steuern 10-23
- Peripheriedaten A-108
- Peripheriezugriffsfehler (PZF)..... 11-36, A-18
- bei Prozessabbild-Aktualisierung
(OB 85) A-17
- bei Zugriff auf nicht vorhandene
Peripherieadresse (OB 122) 11-36
- Personenschaden verhüten 10-24
- PG/PC-Schnittstelle 1-19
- parametrieren..... 1-19
- Phasenverschiebung.....2-19
- Physik..... 11-2, 11-3
- Piktogramme (siehe Symbole) 1-1, 11-2
- Pinnadel 3-8
- Platzhalterbaugruppe DM 370 Dummy.....5-9
- Pneumatic-Interface-Modul5-10
- POINTER A-50, A-52, A-53, A-54
- Parametertyp A-50
- Pointerformat.....A-50, A-52
- Powermodul5-10
- Prinzipielle Vorgehensweise..... 10-2
- Beobachten und Forcen mit
Forcetabellen..... 10-2
- Beobachten und Steuern..... 10-2
- Ermitteln einer STOP-Ursache 11-12
- Erstellen von Codebausteinen.....6-33
- Priorität5-7
- ändern2-3
- für Alarme einstellen.....5-7
- Hintergrund-OB2-23
- Prozessalarm2-21
- Uhrzeitalarm2-16
- Verzögerungsalarm2-18
- Prioritätsklassen5-7
- einstellen5-7
- PROFIBUS-Adresse.....5-15
- PROFIBUS-DP-Schnittstelle der CPU.....5-7
- Programm.....1-3, 1-9
- einfügen.....4-4
- Referenzdaten1-5
- Programm in das PG laden9-10
- Programmablauffehler (OB 85) 11-32
- Programmablauffehler-OB..... 11-32
- Programmbearbeitung
alarmgesteuert2-3, 2-14
- zyklisch.....2-6
- Programmbeispiele
einfügen von Ersatzwerten 11-26
- Ersatzwerte 11-26, 11-27
- FB für das Beispiel eines industriellen
Mischprozesses..... A-83
- FC für das Beispiel eines industriellen
Mischprozesses..... A-87
- OB für das Beispiel eines industriellen
Mischprozesses..... A-89
- reagieren auf Batteriefehler 11-23
- Programme in einer CPU2-1
- Programmieren
einer FC
Beispiel..... A-87
- eines OB 1
Beispiel..... A-89
- FB
Beispiel..... A-83
- Hintergrund-OB2-23
- Programmierfehler (OB 121) 11-35
- Programmierfehler-OB 11-35
- Programmiersprache
auswählen6-21
- festlegen6-21
- FUP (Funktionsplan).....6-23

KOP (Kontaktplan)	6-22
umschalten	6-37
Programmiersprache AWL (Anweisungsliste)	6-23
Programmiersprachen	1-3, 1-4
Programmierung entwerfen eines strukturierten Programms	A-80
Übergeben von Parametern	2-26
Verwendung von Datenbausteinen	2-26
Programmkonsistenz herstellen	6-96
Programmstatus	10-27
anzeigen	10-28
Anzeigen festlegen	10-30
Steuern von Variablen	10-33
Test ein- und ausschalten	10-33
Programmstatus von Datenbausteinen	10-29
Programmstruktur	1-7, 6-81, 6-86, 6-87
Projekt	1-6, 1-8, 1-9
anlegen	4-4
kopieren	4-6, 4-9
löschen	4-6
öffnen	4-6
Software erstellen (prinzipiell)	4-10
Projektbestandteile drucken	12-14
Projektdatei auf Micro Memory Card (MMC) ablegen	8-2
Projektdatei zwischen STEP 7 Lite und STEP 7 austauschen	8-6
Projektdokumentation	1-7, 4-10
Projektdokumentation drucken	12-14
Projektdokumentation zusammenstellen	12-3
Projekte Bearbeitungsreihenfolge	4-4
Projekte auf Datenträger sichern	8-2
Projektfenster	3-3, 3-4, 4-1, 4-3
Projektfenster und Sichten in STEP7Lite	1-5
Projektsicht	4-1
Projektteil kopieren	4-9
proportionale Schrift	12-7
Prozess untergliedern	A-77
Prozessabbild	2-10, A-17
aktualisieren	2-10
Ein-/Ausgänge	A-17
löschen	2-21
Prozessalarm	2-14
Priorität	2-21
Regeln	2-20
starten	2-20
Prozessalarmfähige Signalbaugruppen parametrieren	2-20
Prozessbetrieb	2-10
Prozessüberwachung	10-2
Prüfen der Konsistenz einer Stationskonfiguration	9-7
PZF (Peripheriezugriffsfehler)	A-17

Q

QRY_TINT	2-16
Quellen	8-7
externe	4-4
Querformat	12-1
Querverweise	6-81, 6-95
zu Operanden mit überlappenden Adressbereichen	6-95
Querverweisliste	1-7, 6-81, 6-82

R

RAM	A-12, A-24
RAM-Bereich	A-13
RDSYSST	11-17, 11-18, A-23
READ_CLK	A-112
READ_RTM	A-112
REAL Datentyp	A-29
Realzahl Bereich	A-27
Datentyp	A-27
Referenzdaten	6-16, 6-17
Referenzdaten (siehe Referenzen)	6-81
Regeln für den Umgang mit mit License Keys	1-15
für die Bildung von Multiinstanzen	6-46
für die Eingabe von AWL-Anweisungen	6-69
für die Eingabe von FUP-Anweisungen	6-63
für die Eingabe von KOP-Anweisungen	6-53
für FUP	6-63
für KOP	6-53
Prozessalarm	2-20
Uhrzeitalarm	2-15
Verzögerungsalarm	2-18
Weckalarm	2-19
Regeln für den Umgang mit License Keys	1-15
Regeln für die Anordnung von	5-8, 5-9, 5-10
Baugruppen (SIMATIC 300)	5-8
Modulen (ET 200S)	5-10
Modulen (ET 200X)	5-10
SIM 374 IN/OUT 16	5-9
Register	3-7, 6-31
Eigenschaften	6-31
Registerdialog	3-6
Registerdialog 'Baugruppenzustand'	11-7
Register-Inhalt	6-103
Rekursionen in Bausteinabhängigkeiten (Darstellung)	6-86
Remanenter Speicher bei S7-300-CPU's	A-23
Remanenz nach Netzausfall	A-4
Remanenzverhalten einstellen	5-7
Restzyklus	A-7
RET_VAL auswerten	11-22
Retaineigenschaft von DBs	A-25
Ringpuffer (Diagnosepuffer)	A-21
RPL_VAL	11-26, 11-27
RTD-Modul	5-10

- Rücksetzen
 Datenwerte auf die Anfangswerte 6-80
- RUN A-9
 Betriebszustand der CPU A-1
 CPU-Tätigkeiten A-4
- RUN-P A-10
- S**
- S5TIME
 Bereich A-27
 Datentyp A-35
 Format A-35
 Zeitbasis A-35
- S7-31xC 8-1
 S7Lite-Exportdatei 8-2
 S7-Memory Card 9-6
 S7-PLCSIM 10-1
 Sachschaden verhüten 10-24
 Sammelwarnung 5-20
 Schachtelungstiefe 2-8
 Schlüsselschalter A-9
 Schnelles Konfigurieren 9-8
 Schnelles Positionieren auf
 Verwendungsstellen im Programm 6-91
 Schnittstellen der CPU parametrieren 5-7
 Schnittstellenmodule und Schnittstellen
 (Darstellung in HWKonfig) 5-12
 Schriftart 12-1, 12-8
 Schriftart Formatvorlage 12-1, 12-11, 12-12
 Schriftart Formatvorlagen verwenden 12-12
 Schriftgröße 12-1, 12-8
 Schriftstil 12-1, 12-8
- Schritte
 zum Anzeigen und Ändern des
 Betriebszustands 7-3
 zum Anzeigen und Einstellen von
 Uhrzeit und Datum 7-3
 zum Öffnen des Bausteins zu einem
 Diagnosepuffereintrag 11-14
- Schritte zum Laden von Objekten
 aus der CPU in das PG/PC 9-10
- Schutzstufe einstellen 5-7
- SDB (Systemdaten) A-108
- sdf 8-4, 8-6, 8-7
- SDF 8-8
- sdf-Datei 8-6
- Seitenlayout 12-9, 12-10
 seitenlayout Formatvorlage 12-1
 Seitenlayout Formatvorlage 12-13, 12-14
 Seitenlayout Formatvorlage verwenden 12-10
 Seitennummerierung 12-3
- Senden
 eigener Diagnosemeldungen 11-20
- Service-Fall 13-2
- SET_CLK 2-16, A-112
- SET_RTM A-112
- SET_TINT 2-16
- Setup
 Flash-File-System 1-17
 Ident-Nummer eintragen 1-17
- Memory Card-Parametrierung 1-17
- SFB 2-32, A-37
- SFB 20 STOP 2-10
- SFC 2-33
- SFC 0 SET_CLK 2-15, A-112
- SFC 1 READ_CLK A-112
- SFC 2 SET_RTM A-113
- SFC 20 BLKMOV A-14
- SFC 22 CREAT_DB A-13
- SFC 26 UPDAT_PI 2-10, A-17
- SFC 27 UPDAT_PO 2-10, A-17
- SFC 28 SET_TINT 2-15
 Beispiel in AWL A-95
- SFC 29 CAN_TINT 2-16
 Beispiel in AWL A-95
- SFC 3 CTRL_RTM A-112
- SFC 30 ACT_TINT 2-15, 2-16
 Beispiel in AWL A-95
- SFC 31 QRY_TINT 2-15
 Beispiel in AWL A-95
- SFC 32 SRT_DINT 2-18
- SFC 36 MSK_FLT 2-25
 Beispiel in AWL A-71
 Beispiel in KOP A-71
- SFC 37 DMSK_FLT 2-25
 Beispiel in AWL A-71
 Beispiel in KOP A-71
- SFC 38 READ_ERR
 Beispiel in AWL A-71
 Beispiel in KOP A-71
- SFC 39 DIS_IRT 2-25
 Beispiel in AWL A-75
- SFC 4 READ_RTM A-113
- SFC 40 EN_IRT 2-25
 Beispiel in AWL A-75
- SFC 41 DIS_AIRT 2-24
 Beispiel in AWL A-76
- SFC 42 EN_AIRT 2-25
 Beispiel in AWL A-76
- SFC 44 RPL_VAL 11-26
- SFC 46 STP 2-10
- SFC 48 SNC_RTCB A-112
- SFC 51 RDSYSST 11-17, 11-18, A-21
- SFC 52 WR_USMSG 11-20
- SFC 55 WR_PARM A-108, A-110
- SFC 56 WR_DPARM A-108, A-110
- SFC 57 PARM_MOD A-108, A-110
- Shift+F1 3-2
- Sicherheitshinweise
 überschreiten des L-Stack A-18
- Sicherheitsvorkehrungen zum Forcen
 von Variablen 10-25
- Sichern von Informationen aus dem
 Zielsystem 9-8
- Sichten 1-5, 1-6, 1-7, 1-8
- SIM 374 IN/OUT 16 5-9
- Simulationsbaugruppe SIM 374 IN/OUT 16 5-9
- Sitzungsgedächtnis 3-9
- SNC_RTCB A-112
- Soll-Ist-Baugruppenüberwachung
 Anlauf-OBS 2-21

Soll-Ist-Vergleich.....	A-4	Startereignisse für Anlauf-OB.....	2-21
Sortieren		Station.....	5-11
in der Querverweisliste	6-83	konfigurieren.....	5-11
Symbole.....	6-13	parametrieren.....	5-11
Symboltabelle	6-17	Stationskonfiguration	
Spaltenbreite von Deklarationstabellen		(siehe HW-Konfiguration).....	5-4
variieren.....	6-44	Stationskonfiguration in ein	
Spaltenbreiten bei einer		Zielsystem laden	9-7
Deklarationstabelle einstellen	6-36	Stationstyp.....	5-11
Spannungslos		auswählen	5-11
Betriebszustand der CPU	A-1	Statusleiste.....	3-3, 3-4
Speicherbereiche.....	A-12	Statusleiste ein-/ausschalten.....	6-18
Adressbereiche.....	A-16	Steckplatz- und sonstigen Regeln	
Arbeitsspeicher	A-12	(Hardware konfigurieren).....	5-7
Ladespeicher	A-12	Steckplatzregeln (S7-300).....	5-8
remanenter Speicher	A-23	STEP 7	8-1
Systemspeicher	A-12	STEP 7 Lite	1-1, 1-3, 1-4
Speichern.....	8-2	Fehler während der Installation	1-18
Fensteranordnung.....	3-10	STEP 7 Lite (Projektfenster).....	1-5
geladener Bausteine auf integriertem		STEP7	8-6
EPROM.....	9-6	STEP7Lite	
geladener Bausteine auf		Fehler-OBS	
S7-Memory Card in der CPU	9-6	reagieren auf Fehler	2-24
Symboltabelle	6-18	Steuermodus.....	10-20, 10-21
Variablentabelle	10-5	Steuern.....	10-20, 10-26
Speichern einer Forcetable.....	10-6	CPU im STOP mit vorgegebenen	
Speichern einer Konfiguration.....	5-22	Werten initialisieren	10-23
Speichern und übernehmen.....	4-7	Peripherieausgänge im STOP-Zustand	
Speichern unter	8-1, 8-2	der CPU.....	10-23
Sperrern von Alarm- und Asynchron-		prinzipielle Vorgehensweise.....	10-2
ereignissen		Steuern und Beobachten.....	1-5
Beispiel	A-75	Steuern von Variablen	
Spezifizieren von Baugruppen	9-11	Einführung.....	10-20
Sprache	6-21	einmalig und sofort	10-22
Sprachelemente.....	6-34	im Programmstatus	10-33
Sprachen übersetzen.....	8-12	mit definiertem Trigger.....	10-22
Sprachwechsel	8-13	Steuerwerte	
bei mehrsprachig verwalteten Texten	8-13	Beispiele für die Eingabe.....	10-13
Springen		STOP.....	11-12, A-4
aus der Programmstruktur		Betriebszustand der CPU.....	A-1
Programmteil.....	6-94	STOP-Ursache ermitteln	11-12
aus der Querverweisliste in den		Störungen eingrenzen	11-2
Programmteil.....	6-94	Störungsbehandlung	11-21
Spulen		STRING	A-37, A-39
platzieren	6-53	Stromfluss	6-55
SRT_DINT	2-18	Stromgrenzwerte	5-18
Stack-Inhalte im Betriebszustand STOP....	11-12	Stromversorgungsfehler (OB 81).....	11-29
Stacks	11-10	STRUCT.....	A-37, A-41
Standardbibliothek	4-4	Struktur	
Standardbibliotheken		des Anwenderprogramms	
Übersicht.....	6-32	"Uhrzeitalarme".....	A-95
Starten		erstellen.....	A-46
Prozessalarm	2-20	Ladespeicher	A-13
STEP 7 Lite-Installation	1-17	Struktur des Anwenderprogramms	
Uhrzeitalarm.....	2-15	"Verzögerungsalarne".....	A-102
Verzögerungsalarm.....	2-18	Struktur und Bestandteile	
Weckalarm.....	2-19	der Symboltabelle.....	6-6
Starten von STEP7Lite	3-1	Struktur-Datentyp (STRUCT)	
Startereignisse		Anzahl verschachtelter Ebenen.....	A-42
maskieren	2-25	Beschreibung	A-46
verzögern.....	2-25	Strukturiert programmieren.....	2-7

- Strukturierte Datentypen A-37, A-42
 Feld A-43, A-44, A-45
 Verschachtelungsstrukturen
 und Felder A-42
 Struktur
 Verschachtelungsstrukturen
 und Felder A-42
 Strukturierter Datentypen
 Struktur A-46
 Strukturiertes Programm
 entwerfen A-80
 Vorteile 2-2
 Suchen bzw. Ersetzen von Begriffen 3-11
 Suchfunktion für Fehler im Anweisungsteil .. 6-51
 Symbol 10-8
 Symbole 1-3, 1-5, 1-9, 3-5, 6-9,
 6-14, 6-49, 11-4
 bearbeiten 6-14
 bei der Programmeingabe definieren 6-12
 einfügen 6-49
 einzeln definieren 6-14
 filtern 6-13
 gleich 11-4
 global 6-3
 im Projektfenster 1-5, 1-6, 1-7, 1-8
 im Projektfenster 1-1, 3-5
 in der Programmstruktur 6-86, 6-87,
 6-88, 6-89, 6-90
 in der Symboltabelle 6-6
 in der Vergleichstabelle (HW-Vergleich) .. 11-2
 in eine Variablen-tabelle eingeben 10-6
 lokal 6-3
 mehrdeutig 6-9
 sortieren 6-13
 ungleich 11-4
 unvollständig 6-9
 Symbole (Icons) 3-5, 11-2
 Symbole anzeigen 12-7
 Symbole eingeben 6-13
 Symbolik 6-4
 Symbolinformation 6-2
 Symbolinformationen anzeigen 12-7
 Symbolische Adressierung 6-4
 im Beispielprogramm A-81
 Symbolische Namen zuordnen A-81
 Symbolkonflikte in Bausteinabhängigkeiten
 (Darstellung) 6-86
 Symbolleiste 3-3, 3-4, 6-18
 Symbolleiste ein-/ausschalten 6-18
 Symboltabelle 1-6, 1-7, 6-4, 6-8, 8-6, 8-7, 8-8
 Dateiformat für den Import/Export 8-8
 exportieren 6-14
 filtern 6-16
 für globale Symbole 6-6
 importieren 6-14
 öffnen 6-13, 6-14
 sortieren 6-17
 speichern 6-18
 Struktur und Bestandteile 6-6
 Symboltabelle exportieren 8-8
 Symbolzeilen
 einfügen 6-15
 löschen 6-15
 markieren 6-17
 Symbolzeilen in die Zwischenablage
 kopieren 6-17
 Synchroner Fehler 2-24, 2-25
 Synchronfehlerereignisse
 maskieren und demaskieren A-71
 Synchronisationszustand 11-4
 Synchronisieren
 Uhr A-112
 Systemarchitektur
 Betriebszustände der CPU A-1
 Zyklus 2-10
 Systemdaten 11-19
 Systemdiagnose
 erweitern 11-20
 Systemfehler 11-1
 Systemfunktionen 2-2, 2-32, 2-33
 Arten 2-32
 Systemfunktionsbausteine 2-2, 2-32
 Arten 2-32
 Systemfunktionsbausteine (SFB) und
 Systemfunktionen (SFC) 2-32
 Systemspeicher A-12
 Systemverhalten einstellen 5-7
 Systemzustandsliste 11-11, 11-18, 11-19
 Inhalt 11-18
 lesen 11-18
 Systemzustandsliste SZL 11-18
 SZL 11-11, 11-18, 11-19
- ## T
- Taktmerker A-113
 Taktmerker konfigurieren 5-7
 Tastenkombinationen
 für das Bewegen des Mauszeigers 3-17
 für das Markieren von Texten 3-18
 für das Umschalten zwischen
 Fenstertypen 3-19
 für den Zugriff auf die Online-Hilfe 3-18
 für Menübefehle 3-15
 Tauschen von Baugruppen 5-16, 13-1
 TC-Modul 5-10
 Teilnehmeradresse 5-15
 Teilprozessabbild A-17
 aktualisieren mit SFCs A-17
 systemseitig aktualisieren A-17
 TeleService 3-20
 Test
 Betriebsart festlegen 10-32
 mit Programmstatus ein- und
 ausschalten 10-33
 Testen
 Einleitung 10-1
 mit der Variablen-tabelle 10-1
 mit Programmstatus 10-27
 Überblick 10-1
 Testen mit der Variablen-tabelle 13-1
 Texte mehrsprachig verwalten 8-8

- Texttypen mehrsprachig verwalteter Texte .. 8-10
 Thermisches Motormodell..... 5-19
 Thermoelement..... 5-10
 TIMER..... A-50, A-51
 ParametertypA-50
 Tipps und Tricks..... 13-1
 Tipps zum Bearbeiten von
 Stationskonfigurationen 5-16
 Titel
 zu Bausteinen 6-49
 zu Netzwerken 6-49
 Titelleiste..... 3-3, 3-4
 Tooltipp (siehe Kurzinfo)..... 3-2
 Trennen 6-68
 Triggerbedingung..... 10-17
 Triggerbedingungen für Programmstatus .. 10-31
 Triggerpunkt..... 10-17, 10-18, 10-20, 10-21
 Triggerpunkte
 einstellen..... 10-17
- ## U
- Überblick
 Testarten..... 10-1
 Übergabe an In_out-Parameter eines FBA-70
 Übergänge
 von Betriebszuständen A-1
 Übergeben von Parametern
 Beispiel-FB für das Beispiel
 eines industriellen Mischprozesses..... A-83
 Entwerfen von Parametern für ein
 strukturiertes ProgrammA-83
 ParametertypenA-50
 speichern der übergebenen Werte..... 2-26
 Überlappende Adressbereiche 6-95
 Übermittlung von Diagnoseinformationen .. 11-17
 Übernehmen 8-2
 Übernehmen und speichern..... 4-7
 Überschreiben
 Adressen oder Parameter in
 KOP-Elementen..... 6-57
 FUP-Elemente 6-66
 KOP-Elementen 6-57
 Überschreibmodus..... 6-51, 6-57, 6-66
 Überschreiten des L-Stack.....A-18
 Übersetzen 8-12
 mehrsprachig verwaltete Texte 8-12
 Übersetzen von Texten 8-8
 Übersicht
 Bibliotheken 6-32
 Mögliche Aktionen des Motorstarters..... 5-20
 Standardbibliotheken 6-32
 Übersicht der möglichen Referenzen..... 6-81
 Übersicht zur Projektdokumentation 12-1
 Überwachungszeiten für den Anlauf 2-21
 UDT 6-24, A-37, A-48, A-49, A-50
 Struktur eingeben..... 6-78
 Uhr
 parametrieren.....A-112
 synchronisieren.....A-112
 Uhrzeit 7-3
 ändern2-16, 2-17
 einstellen7-3
 lesen A-112
 stellen A-112
 Uhrzeit (TIME_OF_DAY)
 Bereich A-27
 Uhrzeitalarm
 abfragen2-16
 deaktivieren2-16
 Priorität2-16
 Regeln2-15
 starten2-16
 Uhrzeit ändern.....2-15
 Uhrzeitalarmliste
 Handierung..... A-95
 Struktur..... A-95
 Uhrzeitformat..... A-112
 Uhrzeitfunktionen A-112
 Umbenennen
 Objekte3-13
 Umbenennen eines Projekts4-10
 Umschalten
 Programmiersprache6-37
 Umschalten auf "CPU online"1-5
 Umschalten zwischen Fenstertypen3-19
 Umverdrahten
 Bausteine6-51, 6-52
 Operanden6-51
 Unsymmetrie5-19
 Unterbrechungs-Stack.....A-12, A-19
 Unterbrechungszeit A-4
 Unterschied zwischen Speichern und
 Laden von Bausteinen.....9-3
 Unterschiede zwischen Forcen und
 Steuern von Variablen10-26
 Unterstützte Hardware.....1-1
 Unvollständige Symbole.....6-9
 Unvollständige und mehrdeutige
 Symbole in der Symboltabelle6-9
 Unzulässige Verschaltungen in KOP.....6-55
 UPDAT_PI..... 2-10, A-17
 UPDAT_PO 2-10, A-17
 Urlöschen A-4
 der CPU.....9-12
 U-Stack
 Beschreibung A-19
 Verwendung durch den Systempeicher . A-19
- ## V
- Variable10-8
 Variablen 10-18, 10-19, 10-22, 10-25, 10-33
 beobachten..... 10-17, 10-18, 10-19
 forcen 10-25
 in Variablendeklarationstabellen
 kopieren.....6-43
 steuern 10-20, 10-22, 10-33
 Variablendeklarationstabelle6-33, 6-37,
 6-38, 6-41
 Aufgabe6-37
 Datentyp ARRAY eingeben6-42

- einfache Datentypen eingeben 6-41
 FB für das Beispiel eines industriellen
 Mischprozesses A-83
 FC für das Beispiel eines industriellen
 Mischprozesses A-87
 für OB 81 11-26
 Leerzeilen einfügen 6-41
 Multiinstanz eingeben 6-46
 OB für das Beispiel eines industriellen
 Mischprozesses A-89
 Spaltenbreite variieren 6-44
 Struktur 6-39
 Variablen kopieren 6-43
 Variablen löschen 6-44
 Variablen-tabelle 10-2, 10-4
 bearbeiten 10-6
 Beispiel 10-6
 Bereich zusammenhängender
 Operanden einfügen 10-9
 Bereiche aus der Zwischenablage
 einfügen 10-16
 erstellen und öffnen 10-3
 kopieren 10-4
 maximale Größe 10-7
 Nutzen 10-1
 Operanden oder Symbole eingeben 10-6
 speichern 10-1, 10-5
 Syntaxprüfung 10-7
 Variieren
 Spaltenbreite von
 Variablen-deklarationstabellen 6-44
 Verbindung trennen und neu verbinden
 in FUP-Netzwerken 6-68
 Verbindungen in FUP-Netzwerken
 erzeugen 6-67
 Verbraucherabzweig 5-10
 Vergleich der Konfiguration
 (Online/Offline/Physik) 11-4
 Vergleich der Konfiguration
 'Online/Offline/Physik' 11-2
 Vergleichen 6-30
 Bausteine 6-30
 Vergleichsstelle 5-10
 Vergrößern 6-35
 der Ansicht 6-35
 Verhältnis zwischen
 Betriebszuständen der CPU A-1
 Verkleinern
 der Ansicht 6-35
 Vermeiden von Fehlern beim Aufrufen von
 Bausteinen 6-101
 Verschachtelte Aufrufe von Codebausteinen
 Auswirkungen auf den B-Stack und den
 L-Stack A-20
 Verschaltungen 6-55
 Verschieben
 Objekte 3-13
 Verschieben von Baugruppen 5-16
 Verwalten von Texten (für Übersetzung) 8-8
 Verwenden A-113
 anwenderdefinierte Datentypen für den
 Datenzugriff A-48
 Felder für den Datenzugriff A-43
 Parametertyp ANY A-60
 Parametertyp POINTER A-53
 Strukturen für den Datenzugriff A-46
 Systemspeicherbereiche A-15
 von Taktmerkern und Zeiten A-113
 zusammengesetzte Datentypen A-42
 Verwendete
 Bits und Bytes 6-84
 Operanden 6-84, 6-85
 Zähler 6-84
 Zeiten 6-85
 Verwendete Operanden 1-7, 6-81
 Verwendung der Variablen-deklaration in
 Codebausteinen 6-37
 Verwendung von Multiinstanzen 6-45
 Verzögern
 Startereignisse 2-24
 Verzögerte Bearbeitung von Alarm- und
 Asynchronereignisse
 Beispiel A-76
 Verzögerungsalarm
 Priorität 2-18
 Regeln 2-18
 starten 2-18
 Struktur A-102
 Verzweigung (geschlossene) in
 KOP-Netzwerk erzeugen 6-59
 Verzweigung auf 6-62
 Verzweigungen 6-54
 Verzweigungen (parallele) in
 KOP-Netzwerken erstellen 6-59
 Voraussetzungen
 für das Laden 9-1
 Voreinstellungen für Bausteineditor
 KOP/FUP/AWL 6-34
 Vorgefertigte Bausteine 2-32
 Vorgehen beim Installieren 1-17
 Vorgehensweise
 beim Eingeben von Anweisungen 6-48
 Vorgehensweise beim Konfigurieren und
 Parametrieren einer Station 5-11
 Vorlage 12-10
 Vorlagen definieren und verwenden 12-10
- ## W
- Wählen der Sprache 8-13
 Warmstart A-4
 Warnen 5-20
 Warnung
 überschreiten des L-Stack A-18
 Was ist ein STEP 7 Lite Projekt 4-1
 Was wird wann geladen? 9-2
 Was wird wann zurückgeladen? 9-8
 Wechsel der Betriebszustände A-1
 Weckalarm 2-19
 Regeln 2-19
 starten 2-19

Werte entforcen	10-26	Zeitfehler-OB	11-29
Werte forcen	10-2	Zeitstempel	
Wiederanlauf	A-5	als Bausteineigenschaft.....	6-97
abbrechen	A-4	bei globalen Datenbausteinen	6-99
automatisch.....	A-5	bei Instanz-Datenbausteinen.....	6-99
manuell	A-4	bei UDTs und von UDTs	
Wiederbereitschaftszeit.....	5-19	abgeleiteten DBs	6-100
Wiederherstellen		in Codebausteinen	6-98
Fensteranordnung.....	3-10	Zeitstempel und Zeitstempelkonflikte	6-97
Windows	1-16	Zeitstempelkonflikt. 6-96, 6-97, 6-98, 6-99, 6-100	
WORD		Zeitstempelkonflikte in Baustein-	
Datentyp	A-34	abhängigkeiten (Darstellung).....	6-86
Wort (WORD)		Zeitsystem	11-9
Bereich.....	A-27	Zoomen	
Datentyp	A-26	Ansicht.....	6-36
WR_DPARM.....	A-109, A-111	Zugangsberechtigung.....	7-2
WR_PARM	A-109, A-111	Zugriff auf den Peripheriedatenbereich	A-108
WR_USMSG.....	11-20	Zugriff auf den Prozessdatenbereich.....	A-107
		Zugriff auf Prozess- und	
Z		Peripheriedatenbereiche	A-107
Zahlendarstellung		Zulässige Adressen und Datentypen	
binär-codierte Dezimalzahlen (BCD).....	A-34	in der Symboltabelle.....	6-8
Bit.....	A-26	Zulässige Bausteineigenschaften	
Byte.....	A-26	je Bausteinart	6-28
Datum und Uhrzeit (DATE_AND_TIME) ..	A-38	Zulässige Datentypen beim Übergeben von	
Doppelwort.....	A-26	Parametern.....	A-65
Ganzzahl (16 Bit)	A-28	Zuordnen	
Ganzzahl (32 Bit)	A-28	symbolische Namen	A-81
Gleitpunktzahlen	A-29, A-30, A-31	Zuordnen von Datentypen zu Lokaldaten von	
S5TIME	A-35	Codebausteinen	A-63
Wort	A-26	Zuordnen von Speicher	
Zähler		im L-Stack	A-19
Obergrenzen für die Eingabe.....	10-11	Zusammengesetzte Datentypen.....	A-37, A-42
Speicherbereich		Zusammenspiel zwischen	
remanent.....	A-23	Variablendeklarationstabelle und	
Zeichen (CHAR)		Anweisungsteil	6-38
Bereich.....	A-27	Zweige (neu) in KOP-Netzwerk erzeugen	6-59
Zeitbasis für S5TIME	A-35	Zyklische Programmbearbeitung.....	2-3, 2-5, 2-6
Zeiten.....	A-114	Zyklus.....	2-3, 2-4, 2-10, 2-11, 2-12
Obergrenzen für die Eingabe.....	10-10	Zyklusbelastung durch	
Zeiten (T)		Kommunikation.....	2-10, 2-13
Speicherbereich		Zyklusüberwachungszeit	2-11, 2-12
remanent.....	A-23	Zykluszeit	2-11
Zeitfehler (OB 80)	11-29	Zykluszeit zur Vermeidung von Zeitfehlern	
		kontrollieren.....	11-16

