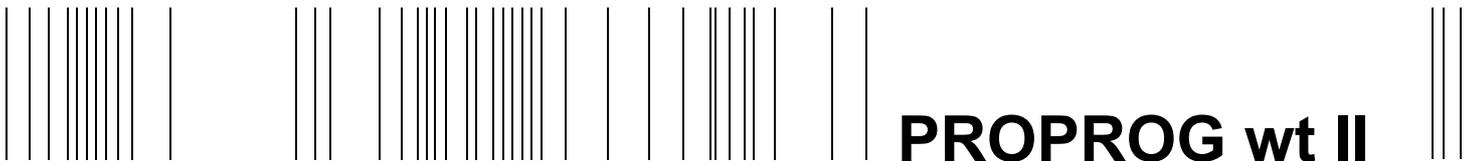


be in motion be in motion



PROPROG wt II

Handbuch

D

5.99003.06



Titel	Handbuch
Produkt	PROPROG wt II
Stand	09. Mai 2005
Artikelnummer	340355
Copyright	<p>Dieses Handbuch darf vom Eigentümer ausschließlich für den internen Gebrauch in beliebiger Anzahl kopiert werden. Für andere Zwecke darf dieses Handbuch auch auszugsweise weder kopiert noch vervielfältigt werden. Verwertung und Mitteilung von Inhalten dieses Handbuches sind nicht gestattet.</p> <p>Bezeichnungen bzw. Unternehmenskennzeichen in diesem Handbuch können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.</p>
Verbindlichkeit	<p>Dieses Handbuch ist Teil des Gerätes/der Maschine. Dieses Handbuch muss jederzeit für den Bediener zugänglich und in einem leserlichen Zustand sein. Bei Verkauf/Verlagerung des Gerätes/der Maschine muss dieses Handbuch vom Besitzer zusammen mit dem Gerät/der Maschine weitergegeben werden.</p> <p>Nach Verkauf des Gerätes/der Maschine sind dieses Original und sämtliche Kopien an den Käufer zu übergeben. Nach Entsorgung oder anderem Nutzungsende sind dieses Original und sämtliche Kopien zu vernichten.</p> <p>Mit der Übergabe des vorliegenden Handbuches werden entsprechende Handbücher mit einem früheren Stand außer Kraft gesetzt.</p> <p>Bitte beachten Sie, dass Angaben/Zahlen/Informationen aktuelle Werte zum Druckdatum sind. Zur Ausmessung, Berechnung und Kalkulationen sind diese Angaben nicht rechtlich verbindlich.</p> <p>Die Firma Baumüller Nürnberg GmbH behält sich vor, im Rahmen der eigenen Weiterentwicklung der Produkte die technischen Daten und die Handhabung von Baumüller-Produkten zu ändern.</p> <p>Es kann jedoch keine Gewährleistung bezüglich der Fehlerfreiheit dieses Handbuches, soweit nicht in den Allgemeinen Verkaufs- und Lieferbedingungen anders beschrieben, übernommen werden.</p>
Hersteller	<p>Baumüller Nürnberg GmbH Ostendstr. 80 - 90 90482 Nürnberg Deutschland Tel. +49 9 11 54 32 - 0 Fax: +49 9 11 54 32 - 1 30 www.baumueller.de</p>



Inhaltsverzeichnis

1	Einführung	7
1.1	PROPROG wt II - ein effizientes, leistungsfähiges und umfassendes Programmierwerkzeug	7
1.2	Welche Dokumentationen erhalten Sie?	8
1.3	Symbole und Textvereinbarungen	8
2	PROPROG wt II und die IEC 61131-3	11
2.1	Was ist die IEC 61131-3?	11
2.2	Konfigurationselemente	12
2.2.1	Konfigurationen	12
2.2.2	Ressourcen	12
2.2.3	Tasks	12
2.2.4	Konfigurationselemente	13
2.3	POEs, Programme, Funktionsbausteine und Funktionen	13
2.3.1	Programm-Organisationseinheiten - POEs	13
2.3.2	POEs	15
2.3.3	Instanziierung	15
2.3.4	Deklarations- und Anweisungsteil einer POE	16
2.4	Variablen und Datentypen	17
2.5	Projekte	18
2.6	Bibliotheken	19
2.7	Programmiersprachen und AS	20
3	Allgemeines zum Programm	21
3.1	Systemvoraussetzungen	21
3.1.1	Hardwarevoraussetzungen	21
3.1.2	Softwarevoraussetzungen	22
3.2	Programminstallation	22
3.3	Programmstart	22
3.4	Arbeiten mit Maus und Tastatur	23
3.5	Die Benutzeroberfläche	24
3.5.1	Anpassen der Benutzeroberfläche	24
3.5.2	Das Menü	25
3.5.3	Symbolleisten	27
3.5.4	Definieren von Tastatur-Shortcuts mit dem Shortcut-Manager	29
3.5.5	Hauptfenster und Arbeitsbereich	35
3.5.6	Meldungsfenster	37
3.5.7	Querverweisfenster	38
3.5.8	Symbole im Querverweisfenster	39
3.5.9	Statuszeile	43
3.6	Verwenden der Hilfe	43
3.6.1	Welche Hilfe-Dateien sind verfügbar?	43
3.6.2	Verwenden der allgemeine und SPS-spezifischen Hilfe	43
3.6.3	Verwenden der Hilfe zu Funktionen und Funktionsbausteinen	44
3.7	Die Editoren	45
3.7.1	Der Projektbaum - ein leistungsstarkes Werkzeug zur Programmorganisation	45
3.7.2	Projektbaum	45
3.7.3	Instanzbaum	46
3.7.4	Der graphische Editor - einfaches Programmieren in AS, FBS und KOP	46



Inhaltsverzeichnis

3.7.5	Der Texteditor - einfaches Programmieren in AWL und ST.	47
3.7.6	Der Seitenlayout-Editor - Gestaltung von Seitenlayouts zum Drucken	47
3.8	Der Editor-Assistent	48
3.9	Das Übersichtsfenster für graphische Arbeitsblätter	51
3.10	Schließen von Arbeitsblättern	52
3.11	Beenden des Programms.	53
4	Arbeiten mit Projekten	55
4.1	Erstellen eines neuen Projekts.	55
4.1.1	Erzeugen eines neuen Projekts mit Hilfe des Projekt-Assistenten	55
4.1.2	Erzeugen eines neuen Projekts mit Hilfe einer Projektvorlage	57
4.2	Speichern eines bestehenden Projekts als Vorlage.	58
4.3	Ändern der Eigenschaften vorhandener POEs	59
4.4	Einfügen neuer POEs.	61
4.5	Einfügen von Arbeitsblättern	63
4.6	Einbinden von Bibliotheken	65
4.7	Löschen von Arbeitsblättern, POEs und Bibliotheken	67
4.8	Speichern von Änderungen in Arbeitsblättern beim Editieren	68
4.9	Speichern eines Projekts unter einem neuen Namen	69
4.10	Packen der Projektdateien in eine Archivdatei.	70
4.11	Übersetzen der Projektsprache	73
4.11.1	Exportieren einer Projektübersetzungsdatei	73
4.11.2	Importieren einer Projektübersetzungsdatei	75
4.11.3	Umschalten der Projektsprache.	77
4.12	Konvertierung des Quell-Codes für AWL, FBS und KOP.	78
5	Literale, Datentypen und Variablen	81
5.1	Literale	81
5.1.1	Numerische Literale.	81
5.1.2	Zeichenfolge- (String-) Literale	82
5.1.3	Zeitdauer-Literale.	82
5.2	Die IEC-Datentypen	83
5.2.1	Elementare Datentypen	83
5.2.2	Generische Datentypen	84
5.2.3	Anwenderdefinierte Datentypen	85
5.3	Datentypen für Felder.	85
5.3.1	Deklarieren von Feldern	85
5.3.2	Programmierbeispiel	86
5.3.3	Mehrdimensionale Felder	87
5.3.4	Initialisieren von Feldern	87
5.4	Datentypen für Strukturen	88
5.4.1	Deklarieren von Strukturen	88
5.4.2	Programmierbeispiel	89
5.4.3	Felder von Strukturen	89
5.4.4	Strukturen mit Feldern	90
5.4.5	Initialisieren von Strukturen	90
5.5	Datentypen für Zeichenfolgen	91
5.5.1	Deklarieren von Zeichenfolgen	91
5.5.2	Programmierbeispiel	91
5.6	Aufrufen des Texteditors mit einem Datentyp-Arbeitsblatt.	91
5.7	Editieren von Datentyp-Deklarationen mit dem Editor-Assistenten	92
5.8	Symbolische, adressierte und direkt dargestellte Variablen.	93
5.9	Globale und lokale Variablen	94
5.10	Remanente Variablen.	95
5.11	Initialisieren von Variablen	95



5.12	Schlüsselwörter zur Variablen-Deklaration	96
5.13	Deklarieren von Variablen	98
5.14	Instanzieren	100
6	Editieren in ST	103
6.1	Aufrufen des Texteditors mit einem ST-Arbeitsblatt	103
6.2	Einführung in ST	104
6.3	Einfügen und Editieren von Zuweisungen	105
6.4	Einfügen und Editieren weiterer Anweisungen	106
6.5	Einfügen von Anweisungen mit dem Editor-Assistenten	107
6.6	Einfügen von Variablen	109
6.7	Aufruf von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten	112
7	Editieren in AWL	115
7.1	Aufrufen des Texteditors mit einem AWL-Arbeitsblatt	115
7.2	Anweisungen, Operatoren, Modifier und Operanden	116
7.3	Einfügen von Anweisungen mit dem Editor-Assistenten	117
7.4	Einfügen von Variablen	119
7.5	Verwenden von Sprüngen und Marken	123
7.6	Aufrufen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten	123
8	Editieren in FBS	127
8.1	Aufrufen des graphischen Editors mit einem FBS-Arbeitsblatt	127
8.2	Einführung in FBS	128
8.3	Einfügen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten	129
8.4	Ändern der Eigenschaften von Funktionen und Funktionsbausteinen	130
8.5	Ersetzen von Funktionen und Funktionsbausteinen	131
8.6	Einfügen von Variablen	132
8.7	Verbinden von Objekten	134
8.8	Negieren von Eingängen und Ausgängen	138
8.9	Duplizieren von Funktionseingängen	139
9	Editieren in KOP	141
9.1	Aufrufen des graphischen Editors mit einem KOP-Arbeitsblatt	141
9.2	KOP-Netzwerke, Kontakte, Spulen und Stromschienen	142
9.3	Einfügen von Kontakten und Spulen	143
9.4	Einfügen von seriellen Kontakten und Spulen	144
9.5	Einfügen von parallelen Kontakten oder Spulen	145
9.6	Verwenden des Editiermodus für KOP-Zweige	146
9.7	Ändern der Eigenschaften von Kontakten und Spulen	147
9.8	Einfügen von Variablen	150
9.9	Aufrufen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten	151
10	Editieren in AS	155
10.1	Aufrufen des graphischen Editors mit einem AS-Arbeitsblatt	155
10.2	Einführung in AS	156
10.3	Einfügen eines Anfangs-AS-Netzwerkes	157
10.4	Einfügen weiterer Schritte und Transitionen	158
10.5	Umwandeln eines Anfangsschritts in einen normalen Schritt und umgekehrt	160
10.6	Einfügen von alternativen Zweigen	161
10.7	Einfügen von simultanen Zweigen	162
10.8	Verwenden des Editiermodus für AS-Zweige	164
10.9	Einfügen von Variablen als Aktionen	165
10.10	Einfügen von Variablen als Transitionen	168
10.11	Aufrufen von Funktionen	172



10.12	Lupen von Aktionen und Transitionen	173
11	Kompilieren, Senden und Debuggen	175
11.1	Einfügen von Konfigurationen, Ressourcen und Tasks	175
11.1.1	Einfügen der DEFAULT-Task	178
11.1.2	Einfügen einer neuen Task	179
11.1.3	Einfügen einer System-Task	181
11.2	Zuweisen von Programmen zu Tasks	181
11.3	Kompilieren eines Projekts	182
11.4	Kompilieren eines Projekts mit 'Make'	183
11.5	Online-Änderungen von POEs	185
11.6	Senden an das Zielsystem	187
11.7	Aufrufen von Arbeitsblättern im Online-Modus	190
11.8	Einstellen des Layout und der Notation von Online-Werten	193
11.9	Umschalten zwischen Online- und Offline-Modus	194
11.10	Umschalten in Adressstatus mit Durchlaufkontrolle	195
11.11	Forcen und Überschreiben von Variablen	196
11.12	Setzen und Rücksetzen von Breakpoints	199
11.13	Debuggen mit gesetzten Breakpoints	201
11.14	Verwenden des Watch-Fensters	204
11.15	Debuggen von anwenderdefinierten Datentypen mit dem Watch-Fenster	206
11.16	Arbeiten mit der Logikanalyse	208
12	Drucken der Projekte mit eigenem Seitenlayout	215
12.1	Drucken des Projekts	215
12.1.1	Steuern des Druckprozesses mit dem Dialog 'Projekt drucken'	215
12.1.2	Definieren eines Seitenlayouts als Standardseitenlayout	218
12.2	Arbeiten mit dem Seitenlayout-Editor	220
12.2.1	Erstellen eines neuen Seitenlayouts	220
12.2.2	Definieren des Codebereichs	221
12.2.3	Einfügen von Elementen in ein Seitenlayout	223
12.2.4	Editieren von Umgebungselementen	223
12.3	Arbeiten mit der Seitenansicht	224
13	Systemgrenzen für PROPROG wt II	227
13.0.1	Grenzen für Projekte	227
13.0.2	Unterstützte Programmiersprachen	228
13.0.3	Datentypen	229
13.0.4	Implementationsbesonderheit anwenderdefinierter Datentypen	229
13.0.5	Implementationsbesonderheit des Datentyps STRING	230
13.0.6	Implementationsbesonderheit des ST	231
13.0.7	Implementationsbesonderheit des VAR_IN_OUT	231
13.0.8	Folgende Besonderheiten sind in PROPROG wt II zu beachten	231

EINFÜHRUNG

1.1 PROPROG wt II - ein effizientes, leistungsfähiges und umfassendes Programmierwerkzeug

PROPROG wt II ist ein Standard-Programmiersystem, sowohl für eine IEC-konforme SPS als auch für eine traditionelle SPS. Es basiert auf der Norm IEC 61131-3 und schließt sämtliche IEC-Merkmale ein.

Das Programmiersystem verfügt über leistungsstarke Funktionen für die unterschiedlichen Entwicklungsphasen von SPS-Anwendungen:

- * Editieren
- * Kompilieren
- * Debuggen
- * Drucken

Das Programmiersystem basiert auf einer modernen 32-Bit-Windows-Technologie und ermöglicht dem Benutzer eine einfache Bedienung durch Werkzeuge wie Zoom, Scrollen, spezielle Symbolleisten, Drag & Drop, einen Shortcut-Manager und andockbare Fenster.

Insbesondere ermöglicht das System die Bearbeitung mehrerer Konfigurationen und Ressourcen innerhalb eines Projektes sowie das Einbinden von Projektbibliotheken. Desweiteren verfügt es über ein leistungsstarkes System zum Debuggen. Mit Hilfe des sehr einfach zu bedienenden Projektbaum-Editors können Projekte angezeigt und editiert werden. Die komplexe Struktur der Norm IEC 61131-3 wird so einfach und transparent wie möglich dargestellt. Dies ermöglicht es dem Benutzer auf einfache Weise POEs, Datentypen, Bibliotheken und Konfigurationselemente in den Projektbaum einzufügen und zu editieren.

Das Programmiersystem besteht aus einem SPS-unabhängigen Kern zum Programmieren in den verschiedenen IEC Programmiersprachen: dazu gehören die Textsprachen Strukturierter Text (ST) und Anweisungsliste (AWL) sowie die graphischen Sprachen Funktionsbaustein-Sprache (FBS), Kontaktplan (KOP) und Ablaufsprache (AS).

Zum Programmieren in den einzelnen Sprachen steht jeweils ein Editor-Assistent zur Verfügung, der ein schnelles und sehr einfaches Einfügen von vorbereiteten Schlüsselwörtern, Anweisungen, Operatoren, Funktionen und Funktionsbausteinen in den einzelnen Arbeitsblättern ermöglicht. Der Editor-Assistent kann auch zum Deklarieren von Variablen und Datentypen verwendet werden.

1.2 Welche Dokumentationen erhalten Sie?

Der unabhängige Kern des Systems wird durch spezielle Teile vervollständigt, die den verschiedenen SPSen angepasst sind.

Das neue, einfache Online-Handling und die 32-Bit-Simulation bieten Möglichkeiten zum Debuggen des Adresstatus und eine Multitask-Testumgebung in Echtzeit.

Ein einfach zu bedienendes Werkzeug für die Projektdokumentation ermöglicht das Drucken des Projektes in zeitsparender optimierter Form (mit weniger Papier) mit einem Seitenlayout, das vom Benutzer festgelegt werden kann.

1.2 Welche Dokumentationen erhalten Sie?

Die Dokumentation gliedert sich in mehrere Teile. Für das Verständnis der Beschreibungen werden Kenntnisse im Umgang mit MS-Windows vorausgesetzt.

Das Handbuch liefert alle Hintergrundinformationen zum besseren Verständnis des Systemkonzepts und der erforderlichen Bedienschritte. Alle Schritte, vom Starten des Programms, über das Editieren von Arbeitsblättern, bis hin zum Beenden werden anhand von verschiedenen Beispielen und Abbildungen beschrieben. Das Handbuch wendet sich vor allem an diejenigen Anwender, die einen vollständigen Überblick über die Erstellung eines SPS-Programms mit Hilfe des Programmiersystems bekommen möchten.

Die kontextsensitive Hilfe, die durch Drücken von F1 aufgerufen werden kann, liefert detaillierte und themenspezifische Informationen zu allen Teilen des Programms. Die kontextsensitive Hilfe besteht aus mehreren Teilen. Ein allgemeiner Teil beschreibt die generell zutreffenden und nicht auf eine bestimmte SPS ausgelegten Themen. Der SPS-spezifische Teil beschreibt alle Objekte, Dialoge und Funktionen, die von SPS zu SPS unterschiedlich sind. Die Hilfe sollte von erfahrenen Anwendern eingesetzt werden, die ein konkretes Problem haben und nach detaillierten Informationen suchen. Die kontextsensitive Hilfe kann auch für Funktionen und Funktionsbausteine aufgerufen werden, die mit Hilfe des Editor-Assistenten in die Arbeitsblätter eingefügt werden können.



SPS-spezifische Informationen finden Sie auch in Ihrer Hardware-Dokumentation.

1.3 Symbole und Textvereinbarungen

In diesem Handbuch werden die folgenden Symbole verwendet:

- * wird für Aufzählungen verwendet.
- wird für einen Bedienschritt verwendet, den Sie durchführen müssen.
- ◇ wird für einen Bedienschritt verwendet, den Sie bei Bedarf durchführen können.



wird für eine Folge von Bedienschritten verwendet, die Sie mit der Maus durchführen.



wird für eine Folge von Bedienschritten verwendet, die Sie mit der Tastatur durchführen.



HINWEIS

Dieses Symbol wird verwendet, um auf eine wichtige Information hinzuweisen



Dieses Symbol wird verwendet, um auf andere Dokumente oder Kapitel im Handbuch zu verweisen.



Dieses Symbol verweist auf Informationen in der Online-Hilfe zum Programmiersystem oder der SPS-Hilfe.

Im vorliegenden Handbuch gelten die folgenden Textvereinbarungen:

- ' einfache Anführungszeichen werden für Namen von Symbolen, Menüpunkte oder Eigennamen von Objekten verwendet, z. B. Menüpunkt 'Ausschneiden' oder Funktionsbaustein 'Level'.
- <ALT> Spitze Klammern werden für Bezeichnungen von Tasten auf Ihrer Tastatur verwendet sowie für Wörter, die Sie eingeben müssen.
- <ALT> + <F4> wird verwendet, wenn Sie zwei Tasten gleichzeitig drücken müssen.
- Editorname* Kursivbuchstaben werden als Platzhalter für Eigennamen verwendet.

PROPROG WT II UND DIE IEC 61131-3

2.1 Was ist die IEC 61131-3?

Die Norm IEC 61131 wurde entwickelt, um die unterschiedlichen Sprachen, Befehlsätze und Konzepte im Bereich der Automatisierungssysteme zu vereinheitlichen. Die Vielzahl der verfügbaren SPS-Konzepte hat in der Vergangenheit dazu geführt, dass Systeme auf unterschiedlichen SPS-Plattformen von unterschiedlichen Herstellern nicht kompatibel sind. Aus diesem Grund mussten bislang für Schulungen sowie für Hard- und Software hohe Investitionen getätigt werden.

In der IEC 61131 werden die Programmiersprachen, die Schnittstellen zwischen SPS und Programmiersystem, die unterschiedlichen Anweisungen sowie die Bearbeitung und die Strukturierung von Projekten vereinheitlicht. Der Einsatz IEC 61131 konformer SPS- und Programmiersysteme bietet den Vorteil der Portabilität aller Plattformen und der Verwendung einheitlicher Konzepte, wodurch die Kosten für die Automatisierungssysteme reduziert werden.

Die Norm besteht aus mehreren Teilen und technischen Berichten. Der dritte Teil befasst sich mit den Programmiersprachen.

Natürlich hat die Norm einen großen Einfluss auf das Konzept, die Struktur und die Funktionen eines Programmiersystems sowie auf das Programmieren des Systems.

Die wesentlichen Veränderungen durch die IEC 61131-3 sind:

- * Die Deklaration der Variablen ähnelt der Variablen-Deklaration in höheren Programmiersprachen.
- * Deklaration von Datentypen ist möglich.
- * Globale und lokale Daten können voneinander unterschieden werden.
- * Programmieren bedeutet symbolisches Programmieren.

Zum besseren Verständnis und zur Vereinfachung des Programmierens sind in den folgenden Abschnitten einige IEC-Grundlagen und deren Umsetzung in Ihrem Programmiersystem beschrieben.

2.2 Konfigurationselemente

In einem IEC 61131-3 konformen SPS-Programmiersystem wird die Hardwarestruktur durch Konfigurationselemente abgebildet.

Diese Konfigurationselemente sind im wesentlichen:

- * Konfigurationen
- * Ressourcen
- * Tasks

2.2.1 Konfigurationen

Eine Konfiguration ist vergleichbar mit einem speicherprogrammierbaren Steuerungssystem, z. B. einem Rack. In einer Konfiguration können eine oder mehrere Ressourcen definiert werden.

2.2.2 Ressourcen

Eine Ressource ist vergleichbar mit einer CPU, die in das Rack eingesetzt werden kann. In einer Ressource können globale Variablen deklariert werden, die nur in dieser Ressource gültig sind. Es können eine oder mehrere Tasks ausgeführt werden.

2.2.3 Tasks

Tasks bestimmen den Zeitplan der ihnen zugewiesenen Programme. Der Anwender muss dazu einer Task Programme zuweisen, deren zeitlicher Ablauf dann bei der Programmausführung von den Einstellungen der zugeordneten Task gesteuert wird. Die Eigenschaften der Task bestimmen den zeitlichen Ablauf (Zeitplan) der Programmausführung.

Die IEC 61131-3 definiert verschiedene Arten von Zeitverhalten. Daher wird zwischen drei verschiedenen Task-Typen unterschieden:

- * **Zyklische Tasks** werden innerhalb von fest definierten Zeitintervallen aktiviert. Das zugewiesene Programm wird dadurch periodisch ausgeführt.
- * **Ereignis- oder Interrupt-Tasks** werden aktiviert, wenn ein bestimmtes Ereignis eingetreten ist.
- * **Systemtasks** werden aktiviert, wenn während der Ausführung des SPS-Programms ein Fehler auftritt. Systemtasks sind mit Systemprogrammen verknüpft.

Jede Task besitzt eine festgelegte Priorität. In sogenannten Systemen mit vorberechtigtem Aufruf wird eine aktive Task mit niedrigerer Priorität sofort unterbrochen wenn eine höherpriorie Task durch ein bestimmtes Ereignis aktiviert wird. In Systemen mit nicht-vorberechtigtem Aufruf können Tasks nicht durch höherpriorie Tasks unterbrochen werden.

HINWEIS



Welche Task-Typen unterstützt werden, hängt von der eingesetzten SPS ab.

2.2.4 Konfigurationselemente

Konfigurationselemente werden im Projektbaum graphisch dargestellt. Sie sind im Unterbaum 'Hardwarestruktur' zusammengefasst.

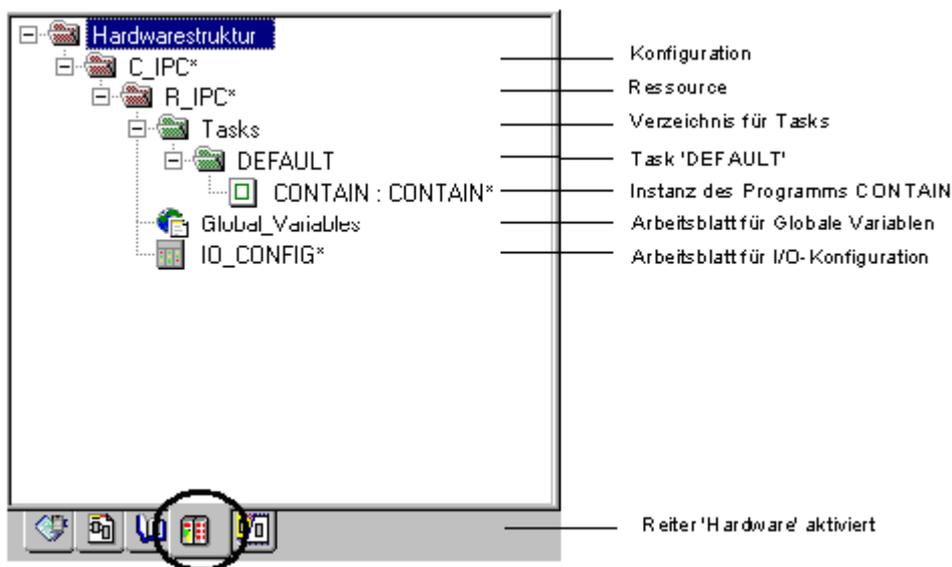


Abbildung 1: Beispiel für Konfigurationselemente

Das Programmiersystem spiegelt die Struktur der Konfigurationselemente im Unterbaum 'Hardwarestruktur' wieder. Diese Struktur kann von SPS zu SPS unterschiedlich sein. Die Abbildung 2-1 zeigt den Unterbaum 'Hardwarestruktur' mit den Konfigurationselementen für Simulation.

Im allgemeinen können eine oder mehrere Konfigurationen verwendet werden, wobei in jeder Konfiguration wiederum eine oder mehrere Ressourcen deklariert werden können. Innerhalb einer Ressource sind mehrere Tasks mit ihren zugewiesenen Programmen möglich.

2.3 POEs, Programme, Funktionsbausteine und Funktionen

2.3.1 Programm-Organisationseinheiten - POEs

Programm-Organisationseinheiten (oder kurz POEs) sind die Sprachelemente eines SPS-Programms. Sie sind kleine, unabhängige Softwareeinheiten, die den Programmcode enthalten. Der Name einer POE muss eindeutig sein, d. h. er darf innerhalb eines Projektes nur einmal vergeben werden.

Gemäß ihrer unterschiedlichen Verwendung wird in der IEC 61131 zwischen drei Typen von POEs unterschieden:

- * Funktionen
- * Funktionsbausteine
- * Programme

Funktionen

Funktionen sind POEs mit mehreren Eingangsparametern und genau einem Ausgangsparameter. Das Aufrufen einer Funktion mit den gleichen Eingangsparametern liefert immer das gleiche Ergebnis. Rückgabewerte können einfache Datentypen sein. In einer Funktion können andere Funktionen aufgerufen werden, jedoch keine Funktionsbausteine oder Programme. Rekursive Aufrufe sind nicht zulässig.

Die IEC 61131-3 definiert verschiedene Typen von Standardfunktionen:

- * Funktionen zur Typumwandlung, wie z. B. INT_TO_REAL
- * Numerische Funktionen, wie z. B. ABS und LOG
- * Arithmetische Standardfunktionen, wie z. B. ADD und MUL
- * Bitfolge-Funktionen, wie z. B. AND und SHL
- * Auswahl- und Vergleichsfunktionen, wie z. B. SEL und GE
- * Zeichenfolge-Funktionen, wie z. B. RIGHT und INSERT
- * Funktionen für Zeit-Datentypen, wie z. B. SUB mit dem Datentyp TIME

Funktionsbausteine

Funktionsbausteine sind POEs mit mehreren Eingangs- und Ausgangsparametern und internem Speicher. Der Wert, den ein Funktionsbaustein als Ergebnis zurückgibt, hängt vom aktuellen Wert seines internen Speichers ab. In einem Funktionsbaustein können weitere Funktionsbausteine oder andere Funktionen aufgerufen werden. Rekursive Aufrufe sind nicht zulässig.

Die IEC 61131-3 definiert verschiedene Typen von Standard-Funktionsbausteinen:

- * Bistabile Elemente, wie z. B. SR und RS
- * Funktionsbausteine für Flankenerkennung, wie z. B. R_TRIG und F_TRIG
- * Zähler, wie z. B. CTU und CTD
- * Funktionsbausteine für Zeitgeber, wie z. B. TON und TOF

Programme

Programme sind POEs, die eine logische Kombination von Funktionen und Funktionsbausteinen enthalten, entsprechend den Erfordernissen des Steuerungsprozesses. Das Verhalten und die Verwendung von Programmen ist ähnlich wie bei Funktionsbausteinen. Programme können einen internen Speicher enthalten. Programme müssen Tasks zugewiesen werden.

In einem Programm können Funktionen und Funktionsbausteine aufgerufen werden. Rekursive Aufrufe sind nicht zulässig.

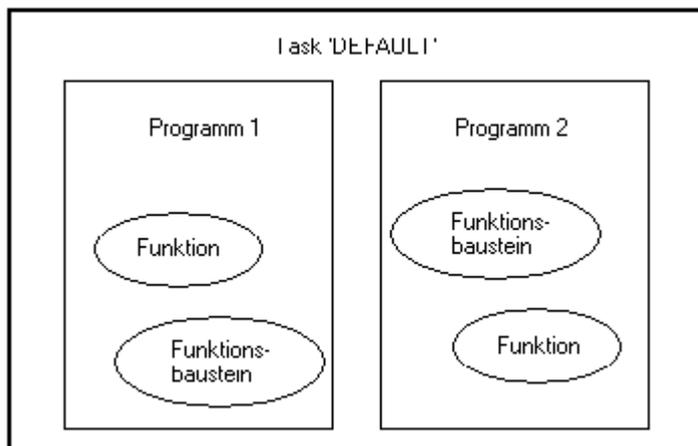


Abbildung 2: Darstellung der Default-Task mit zwei Programmen

2.3.2 POEs

Programme, Funktionsbausteine und Funktionen werden im Projektbaum als Symbole abgebildet und können dort bearbeitet werden. Sie können entweder den vollständigen Projektbaum anzeigen oder durch Klicken auf den Reiter 'POEs' nur die Unterbäume 'Datentypen' und 'Logische POEs' einblenden, wie in folgender Abbildung dargestellt.

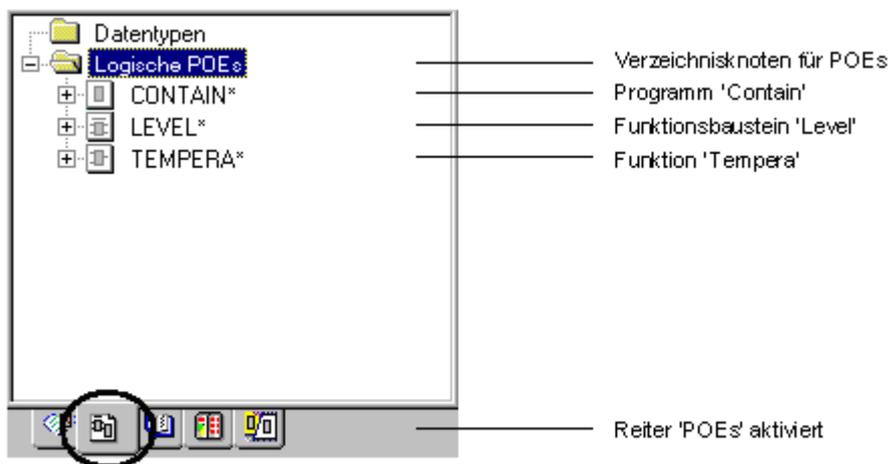


Abbildung 3: Unterbaum 'Logische POEs'

2.3.3 Instanziierung

Damit bereits definierte Funktionsbausteine mehrfach verwendet werden können, bietet die IEC 61131-3 die Möglichkeit der Instanziierung. Das bedeutet, dass der Code eines Funktionsbausteins einmal definiert wird und dann sein interner Speicher verschiedenen Instanzen, also verschiedenen Speicherbereichen, zugewiesen wird. Jede Instanz hat einen Bezeichner (den sogenannten Instanznamen) und enthält die Eingangs- und Ausgangsparameter sowie den internen Speicher des Funktionsbausteins. Ein

2.3 POEs, Programme, Funktionsbausteine und Funktionen

Funktionsbaustein kann in anderen Funktionsbausteinen oder Programmen instanziiert werden. Der Instanzname eines Funktionsbausteins muss in der VAR-Deklaration des Programms oder Funktionsbausteins deklariert werden, in dem er verwendet werden soll. Programme können nur innerhalb derselben Ressourcen instanziiert werden.

Die Instanzen werden in einem eigenen Unterbaum im Fenster angezeigt. Sie blenden den entsprechenden Instanzbaum ein, indem Sie mit der linken Maustaste auf den Reiter 'Instanzen' klicken, wie in der folgenden Abbildung dargestellt.

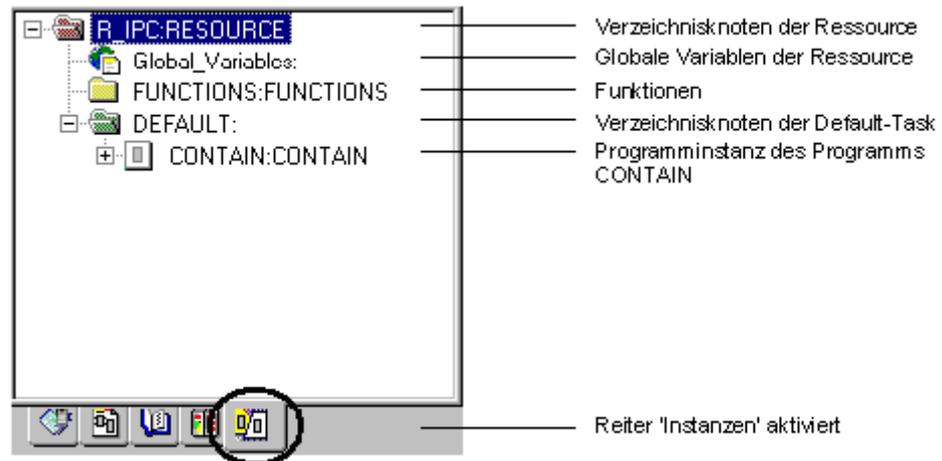


Abbildung 4: Instanzbaum mit den Instanzen der Ressource 'R_IPC'

2.3.4 Deklarations- und Anweisungsteil einer POE

Jede POE besteht aus zwei unterschiedlichen Teilen: Dem Deklarationsteil und dem Code-Teil.

Im **Deklarationsteil** werden alle verwendeten Variablen deklariert.

Der **Anweisungsteil** oder **Code-Teil** einer POE ist der Teil, in dem die Anweisungen in der gewünschten Sprache programmiert werden.

Eine POE enthält drei Typen von Arbeitsblättern, die im Projektbaum als Symbole dargestellt werden.

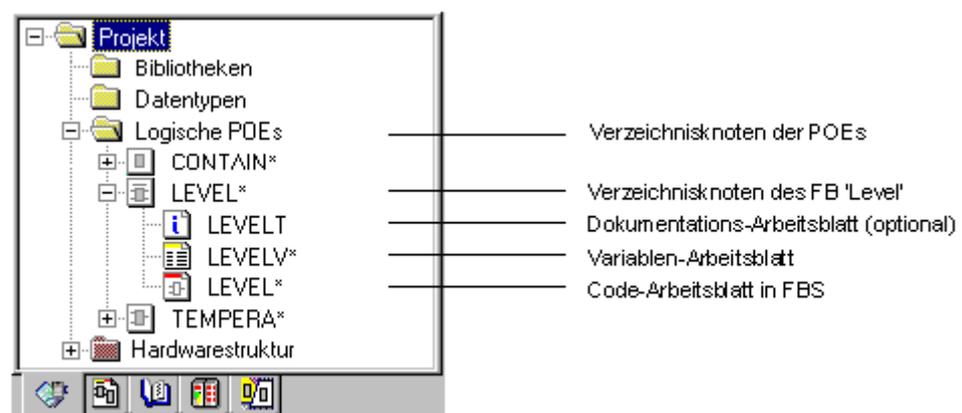


Abbildung 5: Arbeitsblätter eines Funktionsbausteins in FBS

Im Dokumentations-Arbeitsblatt können Anmerkungen zu Dokumentationszwecken eingefügt werden. Im Variablen-Arbeitsblatt werden alle Variablen deklariert. Das Code-Arbeitsblatt enthält die Anweisungen.

In einer POE in AS gibt es noch zwei weitere Symbole: Die Verzeichnisknoten für die Code-Arbeitsblätter (Lupen) der Aktionen und Transitionen.

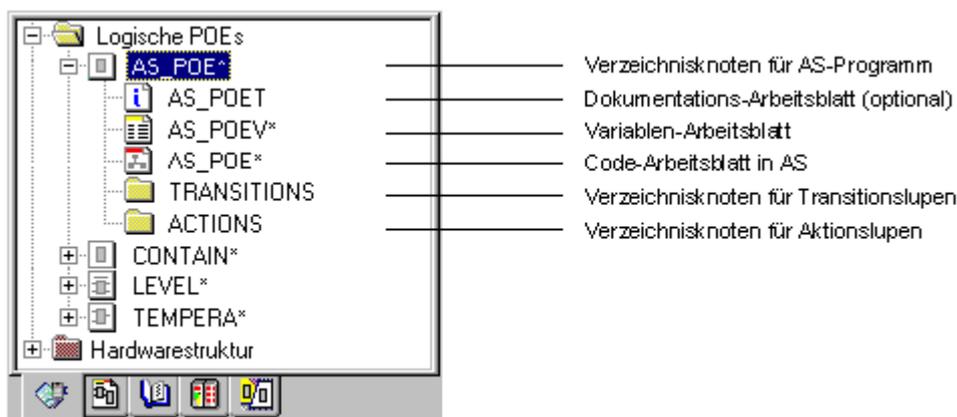


Abbildung 6: Symbole einer AS-POE

2.4 Variablen und Datentypen

In modernen Programmiersystemen gemäß IEC 61131-3 werden **Variablen** verwendet, statt der in früheren Systemen üblichen direkten Adressierung von Speicherbereichen. Während des Kompilierens werden die Variablen automatisch einem Speicherbereich zugewiesen. Die IEC 61131-3 unterscheidet verschiedene Typen von Variablen-Deklorationen, z. B. VAR oder VAR_INPUT. SPS-Eingänge und -Ausgänge können mit dem Schlüsselwort AT direkt adressiert werden.

Variablen werden im Variablen-Arbeitsblatt einer POE deklariert.

Datentypen legen die Eigenschaften für die Werte einer Variablen fest. Sie definieren den Anfangswert, den Bereich der möglichen Werte und die Anzahl der Bits.

Die IEC 61131-3 unterscheidet drei Arten von Datentypen:

- * Elementare Datentypen
- * Generische oder allgemeine Datentypen
- * Anwenderdefinierte Datentypen

Elementare Datentypen sind Datentypen, deren möglicher Wertebereich und Bitbreite durch die IEC 61131-3 definiert wird. Elementare Datentypen sind z. B. BYTE, WORD oder BOOL.

Generische Datentypen sind Datentypen, in denen Gruppen elementarer Datentypen zusammengefasst sind. Sie heißen z. B. ANY_BIT oder ANY_INT. ANY_INT beinhaltet beispielsweise die elementaren Datentypen INT, SINT, DINT, UINT, USINT und UDINT. Generische Datentypen sind notwendig, um die Arten elementarer Datentypen zu definieren, die mit den Eingängen oder Ausgängen von Funktionen verbunden werden können. Wenn eine Funktion beispielsweise mit ANY_INT verbunden werden kann, bedeutet

das, dass sie die Datentypen INT, SINT, DINT, UINT, USINT und UDINT verarbeiten kann.

Anwenderdefinierte Datentypen sind Datentypen, die durch den Anwender deklariert werden können. Sie müssen mit einer TYPE ... END_TYPE-Deklaration definiert werden. Anwenderdefinierte Datentypen können Strukturen oder Felder sein.

Anwenderdefinierte Datentypen werden im Datentyp-Arbeitsblatt im Unterbaum 'Datentypen' deklariert. Sie können entweder den vollständigen Projektbaum anzeigen oder durch Klicken auf den Reiter 'POEs' nur die Unterbäume 'Datentypen' und 'Logische POEs' einblenden, wie in folgender Abbildung dargestellt.

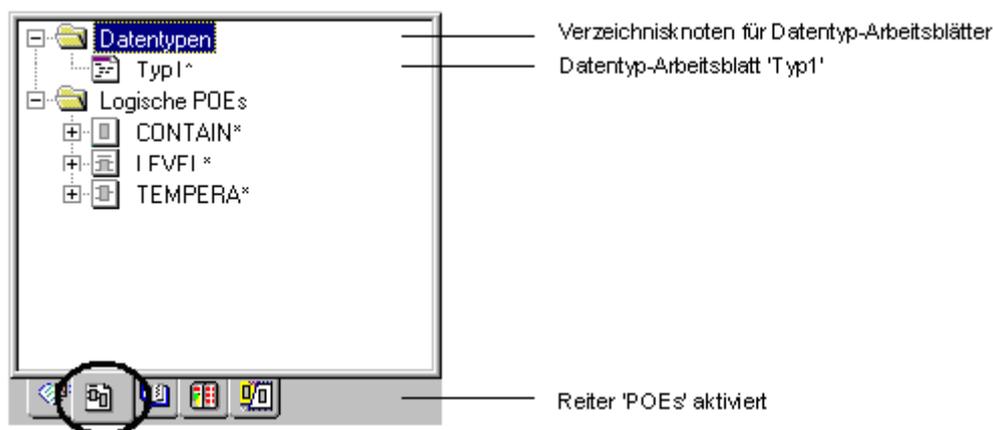


Abbildung 7: Unterbaum 'Datentypen'



Variablen, die anwenderdefinierten Datentypen und deren Deklaration werden in diesem Handbuch im Kapitel 'Literale, Datentypen und Variablen' beschrieben.

2.5 Projekte

Ein Projekt gemäß IEC 61131-3 enthält alle erforderlichen Elemente eines Automatisierungssystems. Es besteht aus Bibliotheken, Datentypen, POEs und sämtlichen Konfigurationselementen (Ressourcen, Tasks und Programme), die im Unterbaum 'Hardwarestruktur' vorhanden sind. Jedes Projekt wird als eigenständiger Projektbaum dargestellt.

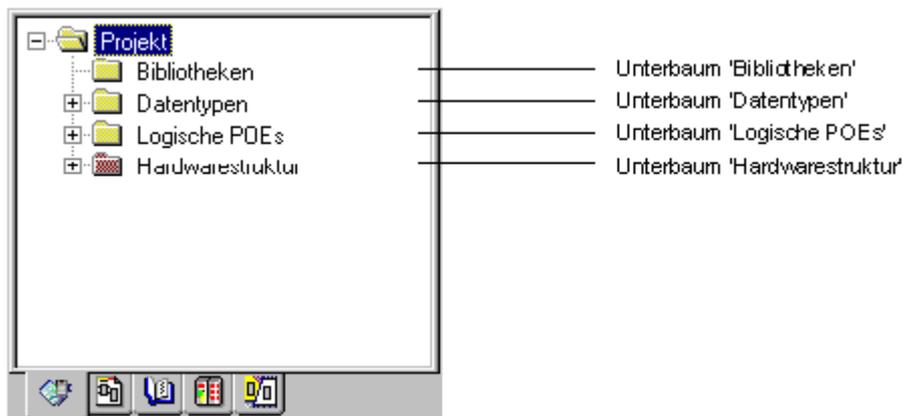


Abbildung 8: Das Projekt mit seinen Unterbäumen

2.6 Bibliotheken

Bibliotheken sind bestehende Projekte, aus denen Sie Teile wiederverwenden wollen und die Sie deshalb als Bibliotheken eingebunden haben. Sie können in dem Projekt, das Sie bearbeiten, die Programme, Funktionsbausteine, Funktionen und die anwenderdefinierten Datentypen aus der Bibliothek wiederverwenden.

Firmware-Bibliotheken sind herstellerspezifische Bibliotheken, deren POEs von Ihrem SPS-Hersteller bereitgestellt wurden. Die Dateierweiterung für Firmware-Bibliotheken ist *.fwl.

Anwenderbibliotheken sind Projekte, die Sie zuvor erstellt haben und deren POEs Sie wiederverwenden wollen. Die Dateierweiterung für Anwenderbibliotheken ist *.mwt.

Für Bibliotheken gibt es einen eigenen Unterbaum im Projektbaum. Sie können entweder den vollständigen Projektbaum anzeigen oder durch Klicken auf den Reiter 'Bibliotheken' nur den Unterbaum 'Bibliotheken' einblenden, wie in folgender Abbildung dargestellt.

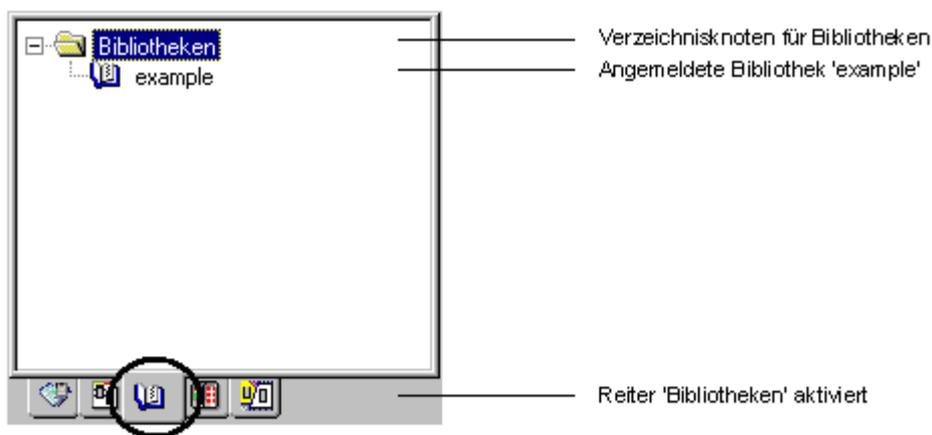


Abbildung 9: Unterbaum 'Bibliotheken'

Der Unterbaum 'Bibliotheken' besteht aus mindestens zwei Symbolen. Das erste Symbol ist der Verzeichnisknoten. Die Symbole in diesem Verzeichnis stellen die eingebundenen Bibliotheken dar. Abbildung 2-9 zeigt die eingebundene Anwenderbibliothek 'example'.

2.7 Programmiersprachen und AS

Die IEC 61131-3 definiert die Syntax, die Darstellungsweise und die verfügbaren Sprachelemente von vier Programmiersprachen.

Man unterscheidet zwei textuelle Sprachen und zwei graphische Sprachen:

- * Die **textuellen Sprachen** sind Strukturierter Text (ST) und Anweisungsliste (AWL).
- * Zu den **graphischen Sprachen** gehören die Funktionsbaustein-Sprache (FBS) und der Kontaktplan (KOP).

Zur Strukturierung der internen Organisation von Programmen und Funktionsbausteinen sieht die Norm die sogenannte **Ablaufsprache (AS)** vor.



Der Ablaufsprache (AS) und den vier Programmiersprachen ist in diesem Handbuch jeweils ein eigenes Kapitel gewidmet.

ALLGEMEINES ZUM PROGRAMM

3.1 Systemvoraussetzungen

3.1.1 Hardwarevoraussetzungen

Für die optimale Leistungsfähigkeit des SPS-Programmiersystems müssen mindestens die folgenden Hardwarevoraussetzungen erfüllt sein:

Gerät/Baugruppe	Minimal	Empfohlen
IBM-kompatibler PC mit Pentium-Prozessor	133 MHz	350 MHz
Arbeitsspeicher	32 MB	64 MB
Festplatte	150 MB freier Speicherplatz	
CD-ROM-Laufwerk	ja	
VGA-Monitor Farbeinstellungen Auflösung	256 Farben 800 x 600	True Color 1024 x 768
RS232-Schnittstelle	optional	
Maus	empfohlen	
Netzwerkkarte (Ethernet)	optional	

3.1.2 Softwarevoraussetzungen

Für die optimale Leistungsfähigkeit des SPS-Programmiersystems müssen mindestens die folgenden Softwarevoraussetzungen erfüllt sein:

- Microsoft Windows® 95, Microsoft Windows® 98, Microsoft Windows® NT, Microsoft Windows® 2000 oder Windows® XP
- Für die HTML-Hilfe wird der Microsoft® Internet Explorer 4.0 oder höher benötigt



HINWEIS

Administratorrechte müssen sowohl für das Installieren als auch für das Ausführen des Programms vorhanden sein.

3.2 Programminstallation

Mit Hilfe des Installationsprogrammes führen Sie alle Schritte durch, die zur vollständigen Installation der Software notwendig sind. Starten Sie das Installationsprogramm wie folgt:

- Legen Sie die CD-ROM in das CD-ROM-Laufwerk ein.
Falls die AutoPlay-Funktion aktiviert ist, startet das Installationsprogramm selbsttätig. In diesem Fall können Sie die beiden nächsten Schritte einfach überspringen.
- Falls das Installationsprogramm nicht automatisch gestartet wird, klicken Sie auf die Windows-Schaltfläche 'Start' und wählen die Menüoption 'Ausführen'. Es erscheint der Dialog 'Ausführen'.
- Geben Sie **d:\setup.exe** ein (wobei d der Laufwerksbuchstabe des CD-ROM-Laufwerkes ist) und drücken Sie <↵>.
- Während der Installation erscheinen mehrere Dialoge, in denen Sie die entsprechenden Eingaben vornehmen müssen.

Nachdem die Installation erfolgreich abgeschlossen ist, finden Sie die neu angelegte Programmgruppe im Windows®-Programm-Menü. Während der Installation werden Sie gefragt, ob Sie Programmsymbole auf Ihrem Desktop hinzufügen wollen.

3.3 Programmstart

Zum Starten des Programms klicken Sie auf die Windows-Schaltfläche 'Start', gehen in das Menü 'Programme' und wählen dort die Option 'PROPROG wt II' aus. Sie können das Programm auch durch Doppelklicken auf das entsprechende Programmsymbol in Ihrem Windows-Desktop starten.

Nach dem Programmstart wird automatisch das zuletzt bearbeitete Projekt geöffnet. Wenn Sie das Programm zum ersten Mal gestartet haben, wird kein Projekt geöffnet.

Wie Sie ein bereits vorhandenes Projekt öffnen, ist in dem Beispiel im nächsten Abschnitt ('Arbeiten mit Maus und Tastatur') beschrieben.

3.4 Arbeiten mit Maus und Tastatur

Das Programm unterstützt die Verwendung der Maus sowie der Tastatur. Für Anfänger dürfte es einfacher sein zunächst mit der Maus zu arbeiten, da es dann nicht notwendig ist, die Shortcuts für die Tastatur zu lernen. Für den Einsatz im industriellen Bereich ist unter Umständen die Verwendung der Tastatur angemessener oder erforderlich.

In diesem Handbuch werden beide Methoden beschrieben: Die Verwendung von Maus und Tastatur. In den nächsten Abschnitten wird die Verwendung der Maus und der Tastatur für das Menü und die Symbolleiste erläutert.

Nachfolgend sehen Sie ein Beispiel für eine Arbeitsanweisung mit Maus und Tastatur in diesem Handbuch.



Öffnen eines gespeicherten Projekts mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Projekt öffnen / Projekt entpacken...'.
Es erscheint der Dialog 'Projekt öffnen'.
- Stellen Sie im Listenfeld 'Dateityp' den gewünschten Dateityp ein.
- Öffnen Sie den entsprechenden Projektordner und suchen Sie die gewünschten Dateien (.mwt sind Projektdateien, .zwt sind gepackte (archivierte) Projektdateien).
- Doppelklicken Sie auf den gewünschten Dateinamen. Das zugehörige Projekt wird geöffnet. Wenn Sie ein gepacktes Projekt ausgewählt haben, wird dieses automatisch entpackt.



Öffnen eines gespeicherten Projekts mit Tastatur-Shortcuts

- Drücken Sie <STRG> + <O>.
- Es erscheint der Dialog 'Projekt öffnen'.
- Stellen Sie im Listenfeld 'Dateityp' den gewünschten Dateityp ein.
- Öffnen Sie den entsprechenden Projektordner, indem Sie den Ordernamen in das Feld 'Dateiname' eingeben und <↵> drücken.
- Suchen Sie die gewünschten Dateien (.mwt sind Projektdateien, .zwt sind gepackte (archivierte) Projektdateien).
- Öffnen Sie das gewünschte Projekt, indem Sie den zugehörigen Dateinamen in das Feld 'Dateiname' eingeben und <↵> drücken.

Das Arbeiten mit Maus und Tastatur in den verschiedenen Editoren wird in den nachfolgenden Kapiteln beschrieben.

3.5 Die Benutzeroberfläche

3.5 Die Benutzeroberfläche

Die Benutzeroberfläche besteht aus sechs Hauptkomponenten: Menüleiste, Symbolleiste, Hauptfenster, Statuszeile, Meldungsfenster und Querverweisfenster.

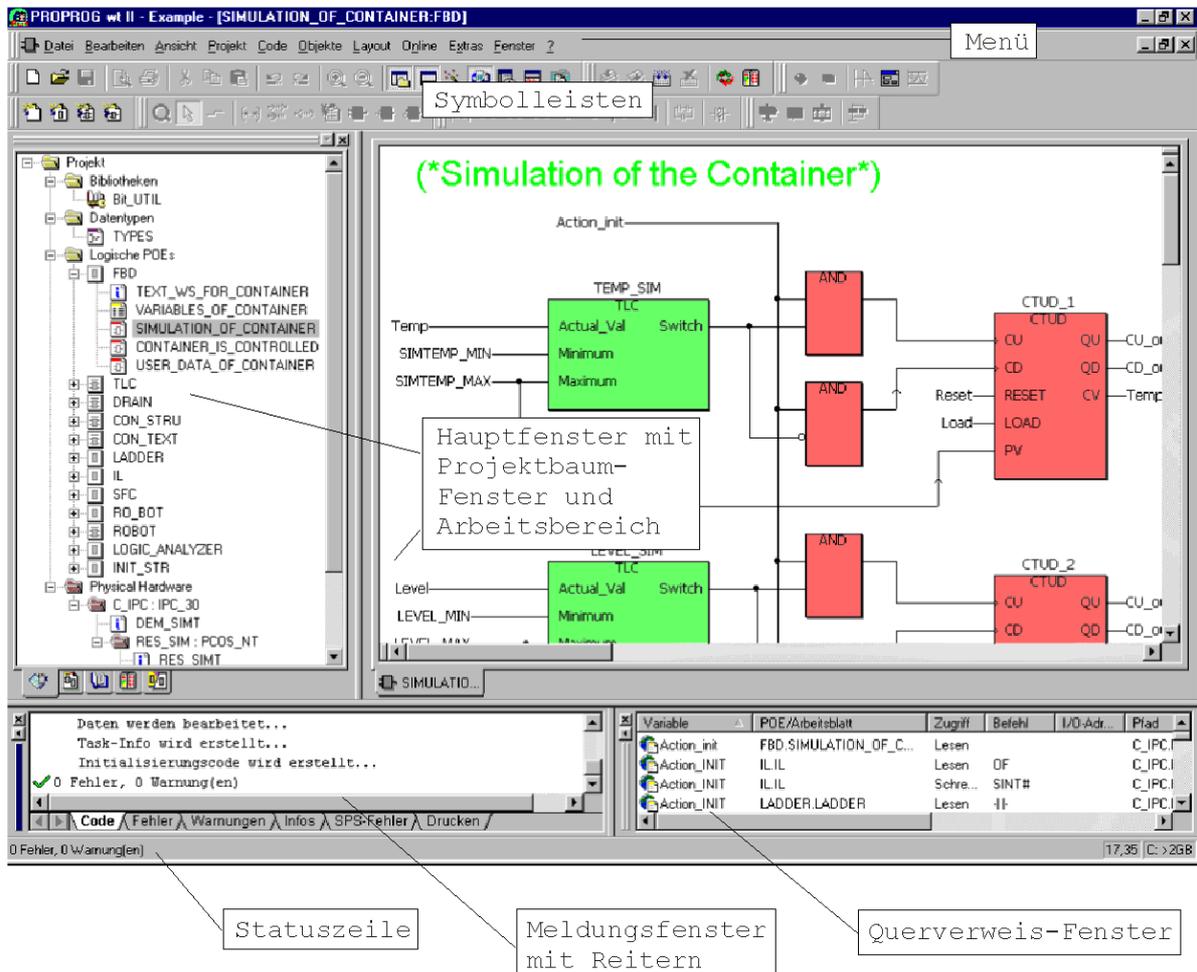


Abbildung 10: Benutzeroberfläche mit geöffnetem Beispielprojekt 'Example'

3.5.1 Anpassen der Benutzeroberfläche

Sie können die Benutzeroberfläche an Ihre persönlichen Bedürfnisse anpassen. Die Anpassung erfolgt im Dialog 'Optionen'. Dieser enthält mehrere Seiten, auf denen Sie das Erscheinungsbild und einige Voreinstellungen für die einzelnen Teile des Systems einstellen.

Mit Hilfe des Dialogs 'Optionen' können Sie:

- eigene neue Symbolleisten erzeugen und diese mit den gewünschten Symbolen "bestücken".
- allgemeine Einstellungen für die Benutzeroberfläche (wie z. B. das Erscheinungsbild des Arbeitsbereichs) vornehmen.
- Optionen für das Kompilieren und Debuggen einstellen.
- Standard-Pfade für das Projekt, die Bibliotheken und Seitenlayouts angeben.

- Standardseitenlayouts definieren.
- Einstellungen für das automatische Projekt-Backup und das Automatische Speichern vornehmen.
- die Darstellung im Texteditor und graphischen Editor festlegen (wie z. B. die verwendeten Farben, Schriften, Standard-Arbeitsblattgröße, usw.).

Detaillierte Information zum Anpassen der Benutzeroberfläche



finden Sie in der allgemeinen Online-Hilfe zum Programmiersystem im Thema 'Anpassen der Benutzeroberfläche'.

3.5.2 Das Menü

Das Menü befindet sich unter der Titelleiste. Es enthält mehrere Untermenüs.

HINWEIS



Die Menüpunkte dieser Untermenüs unterscheiden sich je nach Programmteil oder Editor, mit dem Sie gerade arbeiten.

- * Das Untermenü 'Datei' dient dazu, Projekte zu verwalten, zu speichern und zu packen/entpacken. Des weiteren enthält es Befehle zum Drucken, zum Einrichten des Druckers sowie zum Aufrufen der Seitenansicht.
- * Das Untermenü enthält auch Menüpunkte zum Exportieren und Importieren von Daten. Mit Hilfe dieser Menüpunkte ist es möglich, eine Datei zur Projektübersetzung (csv-Datei) zu exportieren, welche die Kommentare aus allen Arbeitsblättern und die Inhalte der Dokumentations-Arbeitsblätter enthält. Diese csv-Datei kann in jede Sprache übersetzt werden, wozu beispielsweise auch eine Übersetzungssoftware verwendet werden kann. Nachdem die übersetzte Datei wieder in das Projekt importiert wurde, können Sie die Projektsprache umschalten, indem Sie im Dialog 'Projekteigenschaften' die gewünschte Sprache auswählen.
- * Das Untermenü 'Bearbeiten' enthält alle notwendigen Befehle zum Editieren, wie z. B. Markieren, Auswählen verschiedener Arbeitsmodi oder Ausschneiden und Einfügen. Außerdem stellt es Funktionen zum Suchen und Ersetzen von Text zur Verfügung. Diese Funktionen können zum Editieren textueller Arbeitsblätter (dazu gehören z. B. Dokumentations-Arbeitsblätter, Variablen-Arbeitsblätter oder Code-Arbeitsblätter in ST und AWL) und graphischer Arbeitsblätter verwendet werden. Je nach aktivem Editor enthält das Menü auch Befehle zum Bearbeiten von Objekten (z. B. Auswählen oder Verschieben von Objekten, Aktivieren verschiedener Bearbeitungsmodi, etc.).
- * Im Untermenü 'Ansicht' befinden sich Befehle zum Verbergen und Anzeigen der verschiedenen Fenster der Benutzeroberfläche (Projektbaum-Fenster, Meldungsfenster, Querverweifenster, Watch-Fenster, Editor-Assistent, Logikanalyse) sowie der Statuszeile.
- * Das Untermenü 'Projekt' dient zum Einfügen neuer Objekte (z. B. Datentyp-Arbeitsblätter oder Bibliotheken), neuer POEs und Konfigurationen.

- * Das Untermenü 'Code' verfügt über verschiedene Befehle, mit denen Sie nach dem Editieren das Kompilieren starten können.
- * Das Untermenü 'Objekte' ist sichtbar, wenn ein Editor geöffnet ist. Mit dem Menüpunkt 'Variable' können Sie eine neue Variable in die Variablenliste der aktuellen POE einfügen. Wenn Sie ein graphisches Arbeitsblatt editieren, enthält dieses Untermenü weitere Menüpunkte zum Einfügen und Editieren graphischer Objekte (wie z. B. Konnektoren, Sprünge, Kontakte, Spulen, etc.). Je nachdem in welcher graphischen Programmiersprache Sie arbeiten, können einige der Menüpunkte inaktiv sein (d. h. sie sind in heller Schrift dargestellt).
- * Das Untermenü 'Layout' ist sichtbar, wenn Sie mit dem graphischen Editor arbeiten. Es enthält Hilfsfunktionen für die Arbeit in graphischen Arbeitsblättern. So können Sie beispielsweise Seitenränder (im Untermenü 'Ränder') einschalten oder ein Gitter einfügen, um den Inhalt Ihrer Arbeitsblätter besser anordnen zu können. Außerdem können Sie den Inhalt der Arbeitsblätter vergrößert und verkleinert anzeigen (Zoomfunktion), die Arbeitsblattgröße verändern und die Bildlaufgeschwindigkeit sowie die Objektgröße einstellen. Das Untermenü enthält außerdem auch einige Funktionen für das Online-Layout.
- * Das Untermenü 'Online' enthält Befehle zum Debuggen eines Projekts, zum Aufrufen des Kontrolldialogs und zum Aktivieren des Adressstatus. Außerdem kann in diesem Menü mit Hilfe des Befehls 'Online-Layout...' das Erscheinungsbild graphischer Arbeitsblätter im Online-Modus festgelegt werden. Ein zweites Untermenü mit der Bezeichnung 'Logikanalyse' enthält alle Befehle für das Arbeiten mit der Logikanalyse.
- * Das Untermenü 'Extras' dient zum Aufrufen der Dialoge 'Shortcut-Tasten' und 'Optionen'. Außerdem können hier weitere optionale Hilfsprogramme, wie zum Beispiel der Seitenlayout-Editor, implementiert sein. Der Dialog 'Shortcut-Tasten' (auch Shortcut-Manager genannt) ermöglicht es Ihnen, eigene Tastatur-Shortcuts zu definieren oder die vordefinierten Shortcuts nach Ihren Bedürfnissen zu verändern. Der Dialog 'Optionen' enthält alle Befehle, die Sie benötigen, um das Menü, die Symbolleisten, den Texteditor und die Textfarben auf Ihre Bedürfnisse anzupassen.
- * Mit Hilfe der Menüpunkte im Untermenü 'Fenster' können Sie die verschiedenen Fenster und Symbole auf Ihrem Bildschirm anordnen und alle Fenster auf einmal schließen.
- * Das Untermenü '?' enthält alle Befehle zum Aufrufen der Hilfe.

Die folgenden Anweisungen zeigen, wie Sie einen Menüpunkt mit Hilfe der Maus und der Tastatur aufrufen.



Aufrufen des Menüpunktes 'Neues Projekt...' mit der Maus

- Klicken Sie auf das Untermenü 'Datei'. Das Untermenü wird geöffnet und die Menüpunkte werden sichtbar.
- Klicken Sie mit der linken Maustaste auf den Menüpunkt 'Neues Projekt...'. Es erscheint der Dialog 'Neues Projekt'.



Aufrufen des Menüpunktes 'Neues Projekt...' mit der Tastatur

- Drücken Sie <ALT> + <D>. Das Untermenü 'Datei' wird geöffnet und die Menüpunkte werden sichtbar.
- Drücken Sie <r>. Dies ist der unterstrichene Buchstabe des Menüpunktes 'Neues Projekt...'. Es erscheint der Dialog 'Neues Projekt'.

**HINWEIS**

Alle Untermenüs oder Menüpunkte sowie Felder und Kästchen in Dialogen können aufgerufen bzw. aktiviert werden, indem der unterstrichene Buchstabe des entsprechenden Wortes gedrückt wird.

Die einfachste Art mit der Tastatur Menüpunkte aufzurufen, ist die Verwendung der Shortcuts. Aus diesem Grund ist die oben gezeigte Methode, wie Sie mit der Tastatur ein Untermenü öffnen und den Menüpunkt auswählen können nur einmal exemplarisch beschrieben. Die folgende Anweisung ist ein Beispiel für die Verwendung der Shortcuts.

**Aufrufen des Menüpunktes 'Neues Projekt...' mit Shortcut-Tasten**

- Drücken Sie <STRG> + <N>. Dies ist der vordefinierte Standard-Shortcut zum Erzeugen eines neuen Projekts.
Es erscheint der Dialog 'Neues Projekt'.

**HINWEIS**

Den wichtigsten Menüpunkten sind bereits vordefinierte Standard-Shortcuts zugewiesen. Falls ein Menüpunkt noch keinen Shortcut besitzt, können Sie im Dialog 'Shortcut-Tasten' (Shortcut-Manager) einen Shortcut definieren und zuweisen.

3.5.3 Symbolleisten

Die Symbolleisten befinden sich unter der Menüleiste. Standardmäßig sind alle Symbolleisten sichtbar. Die vorhandenen Symbole sind in zwei Reihen angeordnet.

Wenn Sie den Mauszeiger auf einem Symbol platzieren (ohne dieses anzuklicken), erscheint eine kurze Beschreibung zu diesem Symbol, der sogenannte Tooltip. Diese Tooltips zeigen Ihnen den Namen des aktuellen Symbols. Zusätzlich erscheint in der Statuszeile eine kurze Funktionsbeschreibung des Symbols. In der folgenden Abbildung sehen Sie ein Beispiel für einen Tooltip.

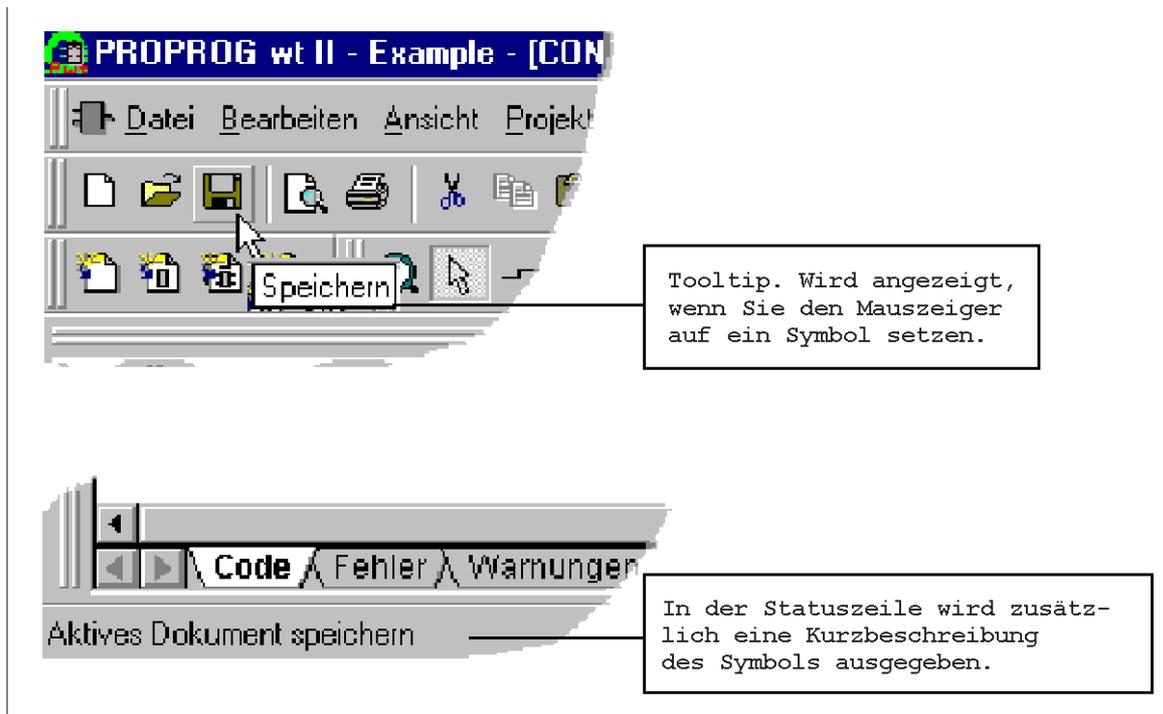


Abbildung 11: Beispiel für einen Tooltip (Symbol 'Speichern') und der zugehörigen Beschreibung in der Statuszeile



HINWEIS

Falls die Tooltips nicht angezeigt werden, öffnen Sie den Dialog 'Optionen', indem Sie im Untermenü 'Extras' den Menüpunkt 'Optionen' auswählen. Blättern Sie auf die Seite 'Symbolleisten' und aktivieren Sie dort das Kontrollkästchen 'Tooltips anzeigen'. Bestätigen Sie den Dialog anschließend mit 'OK'.

Durch die Symbolleisten ist es möglich, häufig anfallende Bedienschritte mit der Maus schnell durchzuführen. Ein Mausklick auf ein Symbol der Symbolleiste führt zu dem selben Ergebnis wie mehrere Bedienschritte ohne Verwenden der Symbolleiste. Sie können alle Symbolleisten entsprechend Ihren Bedürfnissen anpassen. Dazu öffnen Sie den Dialog 'Optionen', indem Sie im Untermenü 'Extras' den Menüpunkt 'Optionen' auswählen. Um eine bestimmte Symbolleiste nicht mehr anzuzeigen, deaktivieren Sie einfach das entsprechende Kontrollkästchen auf der Seite 'Symbolleisten'.

Es gibt zwei verschiedene Arten von Symbolleisten: Allgemeine und spezifische Symbolleisten.

- * Die allgemeinen Symbolleisten enthalten Symbole, die überall im Programm verfügbar sind.
- * Die spezifischen Symbolleisten dagegen enthalten Symbole, die nur in den einzelnen Editoren verwendet werden können. Die Symbole sind (sofern die Symbolleiste angezeigt wird) immer sichtbar; jedoch sind diejenigen Symbole, die in einem bestimmten Editor nicht verwendet werden können, inaktiv. Mit Hilfe des Dialogs 'Optionen' können einzelne Symbolleisten ausgeblendet werden (siehe oben).

**HINWEIS**

Sie können einzelne Symbolleisten aus dem Desktop herauslösen, indem Sie auf den grauen Hintergrund der gewünschten Symbolleiste doppelklicken. Die herausgelöste Symbolleiste wird dann in einem Fenster dargestellt, das Sie wie gewohnt in seiner Größe verändern und beliebig auf dem Bildschirm verschieben können. Um das Symbolleistenfenster wieder in den Desktop einzufügen, doppelklicken Sie in den blauen Titelbalken des Fensters.

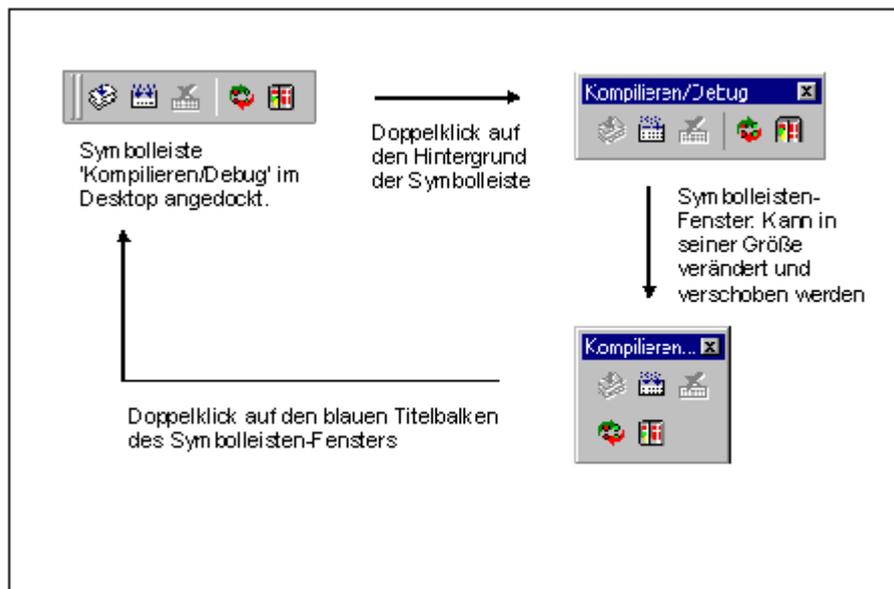


Abbildung 12: Beispiel für ein Symbolleistenfenster

**Verwenden der Symbolleiste zum Erzeugen eines neuen Projekts**

- Klicken Sie auf das Symbol 'Neues Projekt'. Es erscheint der Dialog 'Neues Projekt', in dem Sie eine Vorlage für das neue Projekt auswählen müssen.

**Erzeugen einer neuen Symbolleiste**

Das Erzeugen einer neuen Symbolleiste und das "Bestücken" mit den gewünschten Symbolen ist detailliert in der allgemeinen Online-Hilfe des Programmiersystems beschrieben.

3.5.4 Definieren von Tastatur-Shortcuts mit dem Shortcut-Manager

Wie im Abschnitt 'Das Menü' in diesem Kapitel bereits erwähnt, können Sie bestimmte Menüpunkte am einfachsten durch Drücken der zugeordneten Tastatur-Shortcuts aufrufen.

fen. Ein Tastatur-Shortcut führt durch einfaches Drücken einer Taste oder Tastenkombination dieselbe Operation aus, wie der Menüpunkt, dem der Shortcut zugewiesen ist. In Ihrem SPS-Programmiersystem können mehrere Shortcuts definiert und verwendet werden. Den wichtigsten Menüpunkten (d. h. Operationen) sind bereits vordefinierte Standard-Shortcuts zugewiesen. Welchen Shortcut ein Menüpunkt hat, wird jeweils im Untermenü neben dem Menüpunkt angezeigt.

Mit Hilfe des Shortcut-Managers können Sie einem Menüpunkt einen neuen Shortcut zuweisen oder bestehende Shortcuts ändern. Rufen Sie den Shortcut-Manager wie folgt auf.



HINWEIS

Viele Tastatur-Shortcuts sind bereits standardmäßig vordefiniert. Beachten Sie bitte, dass in diesem Handbuch grundsätzlich Standard-Shortcuts verwendet werden. Einzelne Beschreibungen können deshalb nicht mehr zutreffend sein, nachdem Sie benutzerdefinierte Shortcuts eingestellt haben.



Aufrufen des Shortcut-Managers zum Hinzufügen/Ändern von Shortcuts

- Öffnen Sie das Untermenü 'Extras'.
- Wählen Sie den Menüpunkt 'Shortcuts'. Es erscheint der Dialog 'Shortcut-Tasten'.

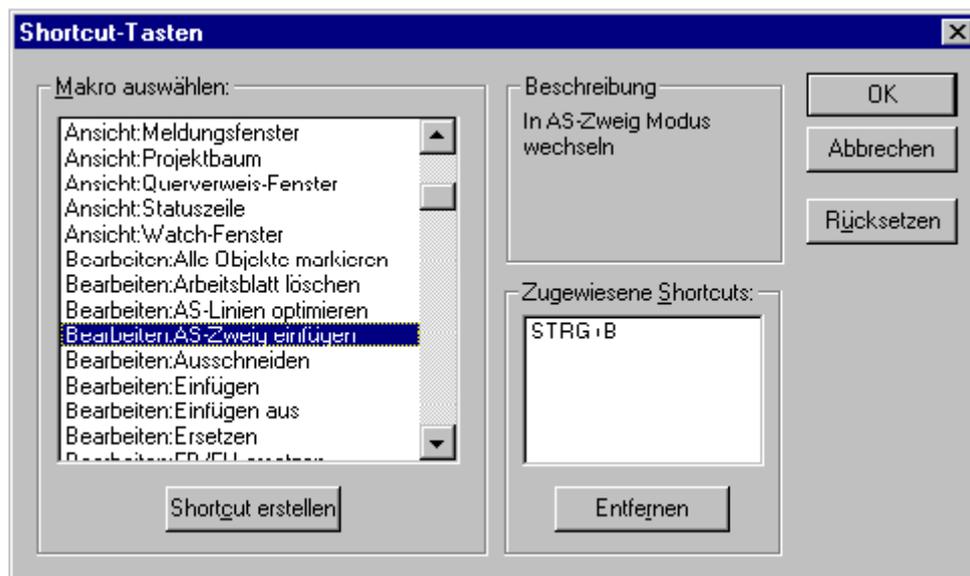


Abbildung 13: Der Shortcut-Manager zum Hinzufügen/Ändern von Shortcuts

Der in [Abbildung 13](#) auf Seite 30 gezeigte Dialog ist in drei Bereiche unterteilt: Die Liste 'Makro auswählen' bietet alle Operationen zur Auswahl an, denen ein Shortcut zugewiesen werden kann. Die meisten Einträge der Auswahlliste sind als Menüpunkte in den Untermenüs verfügbar. Im Feld 'Beschreibung' sehen Sie einen kurzen Text, der Ihnen

Informationen über die Funktion des in der Liste markierten Makros gibt (diese Kurzbeschreibung ist nicht für alle Makros vorhanden!). Wenn das markierte Makro bereits einen Shortcut besitzt, so wird dieser im Feld 'Zugewiesene Shortcuts' eingeblendet.



Hinzufügen/Ändern eines Tastatur-Shortcuts

- Klicken Sie in der Liste 'Makro auswählen' auf die gewünschte Operation bzw. das gewünschte Makro. Der gewählte Eintrag wird markiert. Falls für dieses Makro verfügbar, erscheint im Feld 'Beschreibung' ein kurzer Informationstext.
Wenn für das markierte Makro bereits ein Shortcut definiert wurde, wird dieser im Feld 'Zugewiesene Shortcuts' angezeigt.
- Um einen Shortcut zu **ändern**, müssen Sie zuerst den aktuell zugewiesenen Shortcut löschen und anschließend einen neuen Shortcut zuweisen (siehe nachfolgende Schritte).
- Um einen Shortcut zu **löschen**, wählen Sie ihn mit der linken Maustaste aus und klicken anschließend auf 'Entfernen'.
- Zum **Hinzufügen** eines neuen Shortcuts führen Sie die unten beschriebenen Schritte aus. In diesem Beispiel wollen wir annehmen, dass Sie dem Menüpunkt 'Kompilieren abbrechen' im Untermenü 'Code' einen neuen Shortcut zuweisen wollen.
- Zum **Rücksetzen** aller Shortcuts auf die vordefinierten Standardeinstellungen klicken Sie auf die Schaltfläche 'Rücksetzen'. Alle Shortcuts, die Sie erstellt haben, werden überschrieben!



Zuweisen des Shortcuts <ALT> + <F6> zum Makro 'Code:Kompilieren abbrechen'

Für die folgende Anweisung wird vorausgesetzt, dass der Dialog 'Shortcut-Tasten' bereits geöffnet ist.

- Doppelklicken Sie in der Liste 'Auswählen eines Makro' auf 'Code:Kompilieren abbrechen'. Es erscheint der Dialog 'Shortcut zuweisen'.



Abbildung 14: Dialog 'Shortcut zuweisen' bevor Sie die neuen Shortcut-Tasten gedrückt haben

- Drücken Sie die gewünschte Taste(nkombination). In unserem Beispiel ist dies <ALT> + <F6>. Die gedrückten Tasten werden im Textfeld angezeigt. Unterhalb des Textfeldes zeigt der Shortcut-Manager an, ob die gewünschte Taste(nkombination) bereits einem anderen Makro zugewiesen ist.

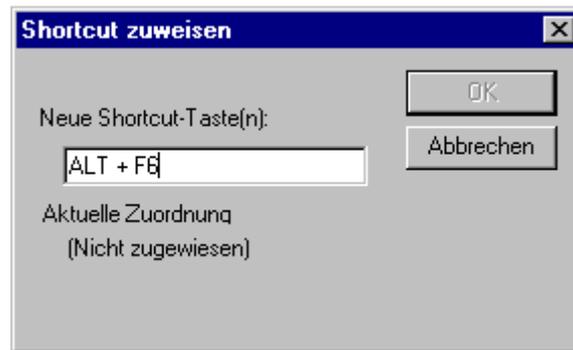


Abbildung 15: Dialog 'Shortcut zuweisen' nachdem Sie die neuen Shortcut-Tasten gedrückt haben

- Bestätigen Sie den neuen Shortcut mit 'OK'.



HINWEIS

Falls die gewünschte Taste(nkombination) bereits einem anderen Makro zugewiesen ist und Sie die Auswahl trotzdem bestätigen, wird die Zuweisung des Shortcuts ohne eine weitere Warnmeldung geändert.

Der Dialog 'Shortcut zuweisen' wird geschlossen und der neue Shortcut erscheint im Dialog 'Shortcut-Tasten'.

In der folgenden Tabelle sind die vom Softwarehersteller vordefinierten Standard-Shortcuts in alphabetischer Reihenfolge aufgelistet:

Operation/Menüpunkt	... Standard-Shortcut
Aktivieren: Editor-Assistent	<ALT> + <3>
Aktivieren: Editor-Fenster	<ALT> + <0>
Aktivieren: Logikanalyse	<ALT> + <6>
Aktivieren: Meldungsfenster	<ALT> + <2>
Aktivieren: Projektbaum-Fenster	<ALT> + <1>
Aktivieren: Übersichtsfenster	<ALT> + <9>
Aktivieren: Querverweise	<ALT> + <4>
Aktivieren: Watch-Fenster	<ALT> + <5>
Ansicht: Editor-Assistent	<UMSCHALT> + <F2>
Ansicht: Logikanalyse	<ALT> + <F11>
Ansicht: Meldungsfenster	<STRG> + <F2>

Operation/Menüpunkt	... Standard-Shortcut
Ansicht: Projektbaum-Fenster	<F2>
Ansicht: Querverweisfenster	<ALT> + <F2>
Ansicht: Watch-Fenster	<ALT> + <>F10
Bearbeiten: Alles auswählen	<STRG> + <A>
Bearbeiten: Ausschneiden	<UMSCHALT> + <ENTF> <STRG> + <X>
Bearbeiten: Bearbeitungsmodus: AS-Zweig einfügen	<STRG> +
Bearbeiten: Bearbeitungsmodus: KOP-Zweig einfügen	<STRG> + <T>
Bearbeiten: Bearbeitungsmodus: Objekt markieren	<STRG> + <M>
Bearbeiten: Bearbeitungsmodus: Objekte verbinden	<STRG> + <L>
Bearbeiten: Einfügen	<EINF>
Bearbeiten: Erweitern/Komprimieren: Spalten einfügen	<STRG> + <3>
Bearbeiten: Erweitern/Komprimieren: Spalten löschen	<STRG> + <4>
Bearbeiten: Erweitern/Komprimieren: Zeilen einfügen	<STRG> + <1>
Bearbeiten: Erweitern/Komprimieren: Zeilen löschen	<STRG> + <2>
Bearbeiten: Inhalt einfügen	<STRG> + <V> <UMSCHALT> + <EINF>
Bearbeiten: Kopieren	<STRG> + <C> <STRG> + <EINF>
Bearbeiten: Löschen	<ENTF>
Bearbeiten: Rückgängig	<STRG> + <Z>
Bearbeiten: Suchen nächstes (global)	F3
Bearbeiten: Suchen nächstes (lokal)	<STRG> + <F3>
Bearbeiten: Suchen vorheriges (global)	<UMSCHALT> + <F3>
Bearbeiten: Suchen vorheriges (lokal)	<ALT> + <F3>
Code: Arbeitsblatt kompilieren	<UMSCHALT> + <F9>
Code: Gehe zum nächsten Fehler	<STRG> + <F12>
Code: Gehe zum vorherigen Fehler	<UMSCHALT> + <F12>
Code: Make	<F9>
Code: Online-Änderungen	<ALT> + <F9>
Code: Projekt neu erzeugen	<STRG> + <F9>
Code: Querverweise erzeugen	<F12>
Datei: Drucken	<STRG> + <P>
Datei: Neues Projekt	<STRG> + <N>
Datei: Projekt öffnen / Projekt entpacken	<STRG> + <O>
Datei: Speichern	<STRG> + <S>

3.5 Die Benutzeroberfläche

Operation/Menüpunkt	... Standard-Shortcut
Ansicht: Projektbaum-Fenster	<F2>
Ansicht: Querverweifenster	<ALT> + <F2>
Ansicht: Watch-Fenster	<ALT> + <>F10
Bearbeiten: Alles auswählen	<STRG> + <A>
Bearbeiten: Ausschneiden	<UMSCHALT> + <ENTF> <STRG> + <X>
Bearbeiten: Bearbeitungsmodus: AS-Zweig einfügen	<STRG> +
Bearbeiten: Bearbeitungsmodus: KOP-Zweig einfügen	<STRG> + <T>
Bearbeiten: Bearbeitungsmodus: Objekt markieren	<STRG> + <M>
Bearbeiten: Bearbeitungsmodus: Objekte verbinden	<STRG> + <L>
Bearbeiten: Einfügen	<EINF>
Bearbeiten: Erweitern/Komprimieren: Spalten einfügen	<STRG> + <3>
Bearbeiten: Erweitern/Komprimieren: Spalten löschen	<STRG> + <4>
Bearbeiten: Erweitern/Komprimieren: Zeilen einfügen	<STRG> + <1>
Bearbeiten: Erweitern/Komprimieren: Zeilen löschen	<STRG> + <2>
Bearbeiten: Inhalt einfügen	<STRG> + <V> <UMSCHALT> + <EINF>
Bearbeiten: Kopieren	<STRG> + <C> <STRG> + <EINF>
Bearbeiten: Löschen	<ENTF>
Bearbeiten: Rückgängig	<STRG> + <Z>
Bearbeiten: Suchen nächstes (global)	F3
Bearbeiten: Suchen nächstes (lokal)	<STRG> + <F3>
Bearbeiten: Suchen vorheriges (global)	<UMSCHALT> + <F3>
Bearbeiten: Suchen vorheriges (lokal)	<ALT> + <F3>
Code: Arbeitsblatt kompilieren	<UMSCHALT> + <F9>
Code: Gehe zum nächsten Fehler	<STRG> + <F12>
Code: Gehe zum vorherigen Fehler	<UMSCHALT> + <F12>
Code: Make	<F9>
Code: Online-Änderungen	<ALT> + <F9>
Code: Projekt neu erzeugen	<STRG> + <F9>
Code: Querverweise erzeugen	<F12>
Datei: Drucken	<STRG> + <P>
Datei: Neues Projekt	<STRG> + <N>
Datei: Projekt öffnen / Projekt entpacken	<STRG> + <O>
Datei: Speichern	<STRG> + <S>

Operation/Menüpunkt	... Standard-Shortcut
Fenster: Alles schließen	F4
Hilfe	<UMSCHALT> + <F1>
Hilfe	<F1>
Layout: Gitter	<STRG> + <.>
Layout: Übersichtsfenster	<ALT> + <9>
Layout: Verkleinern	<STRG> + <->
Layout: Vergrößern	<STRG> + <+>
Layout: Vorherige Ansicht	<STRG> + <<>
Objekte: Aktionsblock	<ALT> + <F8>
Objekte: Eigenschaften Kontakt/Spule ändern	<UMSCHALT> + <F7>
Objekte: Funktion/Funktionsbaustein	<ALT> + <F5>
Objekte: FP duplizieren	<CTRL> + <F5>
Objekte: FP Negation ein/aus	<UMSCHALT> + <F5>
Objekte: Kontakt links	<ALT> + <F7>
Objekte: Kontaktnetzwerk	<F6>
Objekte: Kontakt rechts	<F7>
Objekte: Kontakt unterhalb	<STRG> + <F7>
Objekte: Schritt/Transition	<F8>
Objekte: Simultane / Alternative Verzweigung	<STRG> + <F8>
Objekte: Variable	<F5>
Online: Debuggen	<F10>
Online: Logikanalyse: Aufzeichnung starten	<F11>
Online: Logikanalyse: Triggerbedingungen	<UMSCHALT> + <F11>
Online: Logikanalyse: Variable einfügen/löschen	<STRG> + <F11>
Online: Ressourcekontrolle	<STRG> + <F10>
Querverweise: Nächster Verweis	<ALT> + <F12>
Querverweise: Vorheriger Verweis	<UMSCHALT> + <STRG> + <F12>

Abbildung 16: Standard-Shortcuts in alphabetischer Reihenfolge

3.5.5 Hauptfenster und Arbeitsbereich

Das Hauptfenster (siehe ► [Abbildung 10](#) ◀ auf Seite 24) ist in zwei Teile untergliedert: Das Projektbaum-Fenster und der Arbeitsbereich. Im Arbeitsbereich werden die geöffneten Arbeitsblätter angezeigt. Sie öffnen ein Arbeitsblatt, indem Sie auf das zugehörige Symbol im Projektbaum doppelklicken.

Wenn mehrere Arbeitsblätter geöffnet sind, ist nur ein Arbeitsblatt sichtbar. Jedes geöffnete Arbeitsblatt hat am unteren Rand einen Reiter (siehe folgende Abbildung). Sie wäh-

len ein bestimmtes Arbeitsblatt aus (d. h. Sie aktivieren es), indem Sie entweder auf den entsprechenden Reiter klicken oder mit <STRG> + <TAB> zum gewünschten Arbeitsblatt blättern.



Abbildung 17: Die Reiter der Arbeitsblätter im Arbeitsbereich



HINWEIS

Falls die Reiter der Arbeitsblätter nicht sichtbar sind, können Sie diese im Dialog 'Optionen' aktivieren. Dazu wählen Sie im Untermenü 'Extras' den Menüpunkt 'Optionen' aus. Blättern Sie im Dialog auf die Seite 'Allgemein' und aktivieren Sie das Kontrollkästchen 'Register (Reiter sichtbar)'. Bestätigen Sie Ihre Auswahl mit 'OK'.

Sie haben außerdem die Möglichkeit, mehrere geöffnete Arbeitsblätter in verschiedener Weise im Arbeitsbereich anzuordnen. Zu diesem Zweck wählen Sie einen der Menüpunkte 'Überlappend' oder 'Nebeneinander' aus dem Untermenü 'Fenster' aus. Auf diese einfache Weise können Sie den Arbeitsbereich für die verschiedenen Projektphasen präparieren.

Der Arbeitsbereich kann maximiert werden. Das bedeutet, dass der Projektbaum nicht angezeigt wird und sich dafür der Arbeitsbereich über die gesamte Bildschirmbreite erstreckt. Dies kann zum Beispiel von Vorteil sein, wenn Sie große Netzwerke im graphischen Editor bearbeiten.



Anzeigen/Verbergen des Projektbaum-Fensters mit der Maus

- Klicken Sie auf das Symbol 'Projektbaum'.
Abhängig von seinem vorherigen Status wird der Projektbaum jetzt ein- oder ausgeblendet.



Anzeigen/Verbergen des Projektbaum-Fensters mit der Tastatur

- Drücken Sie <F2>.
Abhängig von seinem vorherigen Status wird der Projektbaum jetzt ein- oder ausgeblendet.

3.5.6 Meldungsfenster

Das Meldungsfenster befindet sich unterhalb des Hauptfensters. Es enthält mehrere Seiten, auf die Sie durch Anklicken des jeweiligen Reiters blättern können.



Abbildung 18: Das Meldungsfenster mit seinen verschiedenen Reitern

Im Meldungsfenster werden die gerade aktiven Prozessschritte während des Kompilierens, die vom Kompiler ausgegebenen Fehler- und Warnmeldungen sowie andere Informationen angezeigt.

Einer der wichtigsten Vorzüge des Meldungsfensters ist, dass Sie von hier direkt auf ein Arbeitsblatt zugreifen können, in dem der Kompiler einen Fehler ermittelt hat. Dazu doppelklicken Sie einfach auf die entsprechende Fehlermeldung.

Sie können das Meldungsfenster wahlweise anzeigen oder verbergen.



Anzeigen/Verbergen des Meldungsfensters mit der Maus

- Klicken Sie auf das Symbol 'Meldungen'.
Abhängig von seinem vorherigen Status wird das Meldungsfenster jetzt ein- oder ausgeblendet.



Anzeigen/Verbergen des Meldungsfensters mit der Tastatur

- Drücken Sie <STRG> + <F2>.
Abhängig von seinem vorherigen Status wird das Meldungsfenster jetzt ein- oder ausgeblendet.

Das Meldungsfenster ist ein andockbares Fenster. Das bedeutet, dass Sie das Fenster durch einen Doppelklick auf den grauen Fensterrand aus dem Desktop herauslösen können. Es wird dann als "normales" Fenster dargestellt, das Sie in seiner Größe verändern und beliebig auf dem Bildschirm verschieben können. Um das Fenster wieder anzudocken, d. h. wieder in den Desktop einzufügen, doppelklicken Sie in den blauen Titelbalken. Das Handling ist gleich wie bei den Symbolleisten.

Arbeiten mit dem Meldungsfenster



Detailierte Informationen zum Arbeiten mit dem Meldungsfenster finden Sie in der allgemeinen Online-Hilfe zum Programmiersystem.

3.5.7 Querverweisfenster

Im Querverweisfenster werden alle externen Variablen, lokalen Variablen und Funktionsbausteine aufgelistet, die im aktuellen Projekt verwendet werden. Dadurch wird das Querverweisfenster zu einem hilfreichen Werkzeug für die Fehlersuche und -beseitigung.

HINWEIS



Jede POE enthält ihre eigenen lokalen Daten. Das bedeutet, dass lokale Variablen in der Querverweisliste jedesmal aktualisiert werden, wenn Sie ein bestimmtes Arbeitsblatt öffnen.

Um die Querverweisliste in einem bestehenden oder neuen Projekt nutzen zu können, müssen die Querverweise zunächst erzeugt werden. Durch das Erzeugen werden die benötigten Information in das Querverweisfenster eingetragen.

Sie können das Querverweisfenster wahlweise anzeigen oder verbergen.

Erzeugen der Querverweisliste mit der Maus



- Falls das Querverweisfenster nicht sichtbar ist, klicken Sie auf das Symbol 'Querverweise'.
Das Querverweisfenster wird nun eingeblendet.
- Platzieren Sie den Mauszeiger im Querverweisfenster.
- Klicken Sie mit der rechten Maustaste in den weißen Fensterhintergrund, um das Kontextmenü zu öffnen.
- Wählen Sie den Menüpunkt 'Querverweise erzeugen'.
Die Querverweisliste wird nun automatisch erzeugt, wie Sie im folgenden Bild sehen können.



HINWEIS



Sie können den Menüpunkt 'Querverweise erzeugen' auch im Untermenü 'Code' oder im Kontextmenü der Editoren aufrufen.

Anzeigen/Verbergen des Querverweisfensters mit der Tastatur



- Drücken Sie <ALT> + <F2>.
Abhängig von seinem vorherigen Status wird das Querverweisfenster jetzt ein- oder ausgeblendet.

Variable	POE/Arbeitsblatt	Zugriff	Befehl	I/O-Adresse	Typ	Kommentar	Zeile/Position(X/Y)
 %K0.2	POE\Arbeitsblatt	Read	LD	%K0.2	EOO_		1
 %K0.3	POE\Arbeitsblatt	Read	AND	%K0.3	EOO_		2
 Action_nit	POE\FOD\CIMUL...	Read			EOO_	Action NIT is active	29/10
 Action_NIT	POE\Arbeitsblatt	Read	OR		EOO_	Action NIT is active	3
 Action_NIT	POE\Arbeitsblatt	Write	ST		EOO_	Action NIT is active	16

Abbildung 19: Querverweisfenster

Für jeden Eintrag enthält das Fenster die folgenden Informationen:

- * Das Symbol am Anfang jeder Zeile verweist auf den angezeigten Datentyp. Die verschiedenen Symbole sind in der folgenden Tabelle aufgeführt.
- * 'POE/Arbeitsblatt': Der Name der POE, in der die Variable/der FB deklariert ist und der Name des Arbeitsblattes, in dem die Variable/der FB verwendet wird.
- * 'Zugriff': Die möglichen Zugriffsarten auf eine Variable sind 'Read' (Lesen d. h. Laden) oder 'Write' (Schreiben d. h. gespeichert).
- * 'Befehl': Hier wird der Befehl angegeben, mit dem die Variable verwendet wurde (nur in AWL- und KOP-Arbeitsblättern).
- * 'I/O-Adresse': Die physikalische SPS-Adresse.
- * 'Pfad': Instanzpfad von globalen Variablen. Dieser Pfad besteht aus dem Namen der Konfiguration und dem Namen der Ressource, in der die globale Variable deklariert ist.
- * 'Typ': Der Datentyp der Variablen.
- * 'Kommentar': Ein vom Benutzer eingegebener Text.
- * 'Zeile/Position(X/Y)': Für textuelle Arbeitsblätter wird hier die Zeilennummer angegeben, für graphische Arbeitsblätter die Position (Koordinaten) des Elements.

HINWEIS



Um das Arbeitsblatt zu öffnen, in dem eine bestimmte Variable verwendet wird, doppelklicken Sie einfach auf den entsprechenden Eintrag im Querverweisfenster. Das Arbeitsblatt wird dann automatisch geöffnet und die gesuchte Variable ist markiert. Wenn Sie eine Variable in einem Editor auswählen, wird diese im Querverweisfenster markiert.

3.5.8 Symbole im Querverweisfenster

Damit Sie einen schnellen Überblick über die angezeigten Objekte erhalten, wird für jeden Querverweis ein Symbol angezeigt, das den Variablentyp angibt. Die folgende Tabelle stellt die verschiedenen Typen und das zugehörige Symbol in der Spalte 'Variable' gegenüber:

Symbol	Bedeutung	Symbol	Bedeutung
	lokale Variable		Schritt
	globale Variable		Transitionslupe

Symbol	Bedeutung	Symbol	Bedeutung
	Eingangsvariable (VAR_IN)		Aktionslupe
	Ausgangsvariable (VAR_OUT)		Sprung
	Ein-/Ausgangsvariable (VAR_IN_OUT)		Marke
	Funktionsbaustein		Konnektor

Abbildung 20: Symbole im Querverweisfenster



Filtern der Querverweisliste

Sie können die Querverweisliste filtern, damit nur eine bestimmte Gruppe von Variablen oder Funktionsbausteinen angezeigt wird.

- Platzieren Sie den Mauszeiger im Querverweisfenster.
- Klicken Sie mit der rechten Maustaste in den weißen Fensterhintergrund, um das Kontextmenü zu öffnen.
- Wählen Sie den Menüpunkt 'Filter...'.
Es erscheint der Dialog 'Querverweis-Filter'.

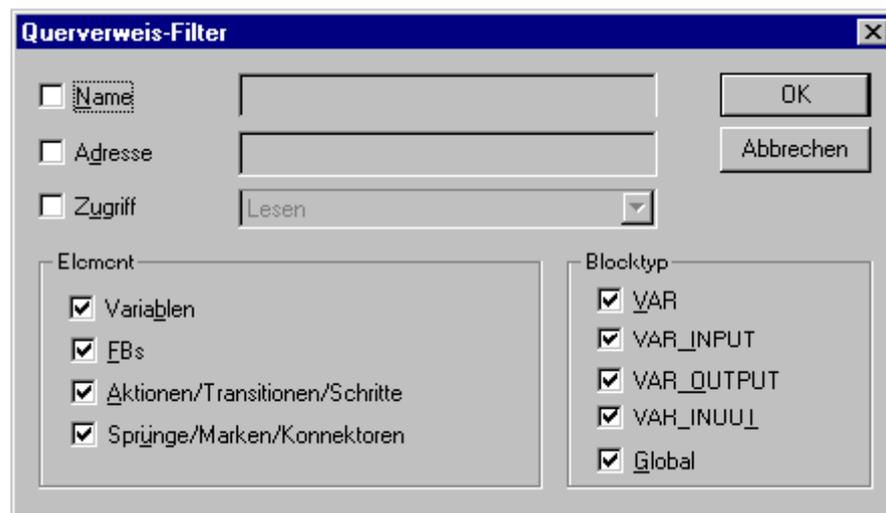


Abbildung 21: Dialog 'Querverweis-Filter' zum Filtern der in der Querverweisliste angezeigten Einträge

- Stellen Sie die gewünschten Filterkriterien ein, indem Sie eines oder mehrere der Kontrollkästchen markieren.
- Um die Liste nach einem spezifischen Namen, einer Adresse oder nach der Zugriffsart zu filtern, markieren Sie das zugehörige Kontrollkästchen. Das nebenstehende Textfeld wird dadurch aktiviert und Sie können den gewünschten Text eingeben. Dabei können Sie Wildcard-Symbole verwenden. Es werden dann in der Liste nur Elemente angezeigt, die diesem Namen entsprechen.
- Nachdem Sie den Dialog bestätigt haben, enthält das Querverweisfenster nur noch Elemente, die den eingegebenen Filterkriterien entsprechen. Im Kopf der Spalte 'Variable' wird angezeigt, dass ein Filter gesetzt ist.
- Der Dialog 'Querverweis-Filter' erlaubt die Verwendung sogenannter **Wildcards** (Stellvertretersymbole).
Das Wildcard-Symbol '*' wird in das Textfeld eingegeben. Es dient als Platzhalter für beliebige Zeichen und wird immer in Verbindung mit anderen Zeichen verwendet. Das Wildcard-Symbol kann am Anfang und am Ende eines Bezeichners eingegeben werden.

Beispiel: Nehmen wir an, Sie wollen in der Querverweisliste nur die Variablen var1, var2 bis var10 anzeigen. Zu diesem Zweck geben Sie die Zeichenfolge 'var*' in das Textfeld ein.



Abbildung 22: Arbeiten mit Wildcards im Dialog 'Querverweis-Filter'



Sortieren der Querverweisliste

Die Einträge der Liste können alphabetisch sortiert werden. Dabei kann jede Spalte als Sortierkriterium verwendet werden. Die Einträge können wahlweise in aufsteigender oder absteigender Reihenfolge sortiert werden. Sortieren Sie die Liste wie folgt:

- Klicken auf den Spaltennamen, der als Sortierkriterium dienen soll.
Die Spalte wird durch einen Pfeil gekennzeichnet, welcher gleichzeitig die Sortierreihenfolge angibt.



Aufsteigende oder absteigende Sortierreihenfolge.

Wenn Sie eine Zeichenkette als Filterkriterium definiert haben, wird der Spaltenkopf entsprechend gekennzeichnet.



Diese Abbildung entspricht dem oben beschriebenen Beispiel, in dem die Kontrollkästchen 'Lokal', 'Extern' und 'FBs' aktiviert sind und die Zeichenkette 'var1' als Filterkriterium eingestellt ist.

- Um die Sortierreihenfolge umzudrehen, klicken Sie noch einmal auf denselben Spaltennamen.



Andocken/Herauslösen des Querverweisfensters

Wie das Meldungsfenster ist auch das Querverweisfenster andockbar. Das bedeutet, dass Sie das Fenster durch einen Doppelklick auf den grauen Fensterrand aus dem Desktop herauslösen können. Es wird dann als "normales" Fenster dargestellt, das Sie in seiner Größe verändern und beliebig auf dem Bildschirm verschieben können. Um das Fenster wieder anzudocken, d. h. wieder in den Desktop einzufügen, doppelklicken Sie in den blauen Titelbalken. Das Handling ist gleich wie bei den Symbolleisten.



Arbeiten mit der Querverweis-Symbolleiste

Das System bietet eine Symbolleiste, die einen schnellen Zugriff auf die Verweise im Querverweisfenster ermöglicht.

Um diese Symbolleiste anzuzeigen, öffnen Sie den Dialog 'Optionen' mit Hilfe des Menüpunktes 'Optionen' im Untermenü 'Extras'.

Wechseln Sie zur Seite 'Symbolleisten' und aktivieren Sie dort das Kontrollkästchen 'Querverweise'. Bestätigen Sie anschließend den Dialog. Die Querverweis-Symbolleiste wird jetzt angezeigt:



Die Symbole haben folgende Bedeutung (von links nach rechts):
Gehe zum ersten/vorigen/nächsten/letzten Querverweis.

3.5.9 Statuszeile

Die Statuszeile gibt während der Arbeit mit dem Programm verschiedene Meldungen aus.

Im linken Teil der Statuszeile sehen Sie Meldungen über ausgeführte Operationen oder Systemmeldungen. Wenn Sie den Mauszeiger auf ein Symbol oder einen Menüpunkt setzen (ohne die Operation auszuführen), dann zeigt die Statuszeile eine kurze Funktionsbeschreibung dieses Symbols oder Menüpunktes an.

Das Feld auf der rechten Seite der Statuszeile gibt die aktuelle Cursorposition an. Falls Sie mit dem graphischen Editor arbeiten, sehen Sie hier die x-y-Koordinaten, beim Texteditor zeigt dieses Feld die aktuelle Zeile und Spalte. Rechts neben der Cursorposition wird der freie Speicherplatz auf der Festplatte angezeigt. Dieses Feld ist rot, wenn auf Ihrer Festplatte zu wenig Speicherplatz frei ist.

Die Farbe der Statuszeile deutet auf den Status des Programmiersystems hin. Dabei gilt folgende Zuordnung zwischen Systemstatus und Farbe:

- * Grau: Offline
- * Grün: Online
- * Rot: Timeout

Sie können die Statuszeile wahlweise anzeigen oder verbergen, indem Sie im Untermenü 'Ansicht' den Menüpunkt 'Statuszeile' aktivieren bzw. deaktivieren.

3.6 Verwenden der Hilfe

3.6.1 Welche Hilfe-Dateien sind verfügbar?

In der kontextsensitiven Hilfe finden Sie zu jedem Programmteil die entsprechenden Hilfethemen. Die Hilfeinformationen sind in zwei oder mehr Teile gegliedert:

- * Eine allgemeine Hilfe beschreibt die generell zutreffenden und nicht auf eine bestimmte SPS ausgelegten Themen.
- * Eine oder mehrere spezifische Hilfen enthalten Beschreibungen zu allen Objekten, Dialogen und Funktionen, die von SPS zu SPS unterschiedlich sind.
- * Hilfe zu Funktionen und Funktionsbausteinen. Bestandteil des Programms ist eine Hilfe zu allen Funktionen und Funktionsbausteinen. Die Hilfethemen sind für IEC-Standardfunktionen/FBs, SPS-spezifische und Firmware-Funktionen/FBs sowie für anwenderdefinierte Funktionen und Funktionsbausteine verfügbar (siehe hierzu die untenstehenden Informationen).

3.6.2 Verwenden der allgemeine und SPS-spezifischen Hilfe

Sowohl die allgemeine als auch die spezifische Hilfe sind hierarchisch strukturiert, wobei drei Typen von Hilfethemen verwendet werden:

- * Hauptthemen: In diesen Themen wird der generelle Umgang mit dem Programm und den Editoren beschrieben. Auf Hauptthemen greifen Sie zu, indem Sie F1 drücken, während ein Editor geöffnet ist.
- * Objektspezifische Themen: Diese Themen liefern Hintergrundinformationen zu den Objekten, die in einem Editor verwendet wurden. Sie rufen ein objektspezifisches The-

ma auf, indem Sie das Objekt, zu dem Sie Hilfe benötigen, im Editor markieren und anschließend <UMSCHALT> + <F1> drücken.

- * Dialogspezifische Themen: In diesen Themen wird die Bedeutung und die Verwendung der einzelnen Felder jedes Dialogs erklärt. Sie öffnen ein dialogbezogenes Thema, indem Sie im Dialog auf die Schaltfläche 'Hilfe' klicken.

Diese Themen können alle kontextsensitiv aufgerufen werden. Es ist aber auch möglich, das Inhaltsverzeichnis oder den Index der Hilfe aufzurufen, um einen Überblick zu erhalten und daraus die gewünschten Themen auszuwählen. Wählen Sie dazu im Menü '?' den Menüpunkt 'Hilfethemen' aus.

3.6.3 Verwenden der Hilfe zu Funktionen und Funktionsbausteinen

Im Editor-Assistent können Sie zu jeder Funktion und zu jedem Funktionsbaustein spezifische Hilfetexte anzeigen. Gehen Sie dazu wie folgt vor:

Klicken Sie im Auswahlfeld des Editor-Assistenten mit der rechten Maustaste auf die Funktion oder den Funktionsbaustein, zu der/dem Sie Hilfe benötigen. Sie öffnen damit das Kontextmenü. Wählen Sie im Kontextmenü den Menüpunkt 'Hilfe zu FU/FB'. Es erscheint das entsprechende Hilfefenster.

Bei Standard-, Firmware- und SPS-spezifischen Funktionen und Funktionsbausteinen wird das kontextsensitive Thema der Hilfe 'Hilfe zu FU/FB' aufgerufen. Diese ist Bestandteil des Programmiersystems. Wenn Sie die Hilfe für eine anwenderdefinierte Funktion oder einen Funktionsbaustein aufrufen, wird entweder das Dokumentations-Arbeitsblatt innerhalb dieser POE geöffnet oder die Inhaltsseite einer vom Benutzer erstellten HTML-Datei in einem speziellen Hilfefenster aufgerufen. Dieses Hilfefenster ist nicht identisch mit dem Standard-Hilfefenster.

Erzeugen einer Hilfe für anwenderdefinierte Funktionen und Funktionsbausteine



Die Vorgehensweisen, wie Sie eine Hilfe für anwenderdefinierte Funktionen und Funktionsbausteine erzeugen, ist in der allgemeinen Hilfe des Programmiersystems beschrieben.



Suchen eines Themas im Hilfesystem

- Wählen Sie den Menüpunkt 'Hilfethemen' aus dem Untermenü '?'. Es erscheint das Inhaltsverzeichnis der allgemeinen Hilfe, mit einer Liste aller Hilfethemen in ihrer hierarchischen Struktur.
- Um nach einem gesamten Hilfethema zu suchen, klicken Sie auf den Reiter 'Index' und geben das gesuchte Thema ein.
- Weitere Informationen zu einem bestimmten Hilfethema erhalten Sie, indem Sie auf die Hypertext-Sprünge innerhalb des Textes oder am Ende eines Themas klicken. Diese farbigen oder unterstrichenen Sprünge führen Sie zu verwandten Themen.
- Der Hilfe-Assistent ermöglicht Ihnen das Suchen nach bestimmten Informationen oder nach einem bestimmten Wort in der Hilfe. Sie bedienen diesen wie folgt:
Klicken Sie im Hilfefenster auf 'Ihre Frage...'. Es erscheint der Fragedialog. Geben Sie in das Textfeld Ihre Frage ein und klicken Sie anschließend auf 'Suchen'. Es werden jetzt alle Themen aufgelistet, in denen die von Ihnen eingegebene Frage beantwortet wird. Um ein bestimmtes Hilfethema anzuzeigen, klicken Sie in der Ergebnisliste auf das gewünschte Thema.

3.7 Die Editoren

3.7.1 Der Projektbaum - ein leistungsstarkes Werkzeug zur Programmorganisation

Der Projektbaum (er wird auch als Projekt-Manager bezeichnet) ist ein einfach zu bedienendes und leistungsstarkes Werkzeug zur Projektorganisation und -verwaltung. Er besteht aus dem Projektbaum-Editor und dem Instanzbaum.

Jedes Projekt wird im Projektmanager als eigener Projektbaum dargestellt. Im Projektbaum können Sie die Struktur Ihres Projektes editieren.

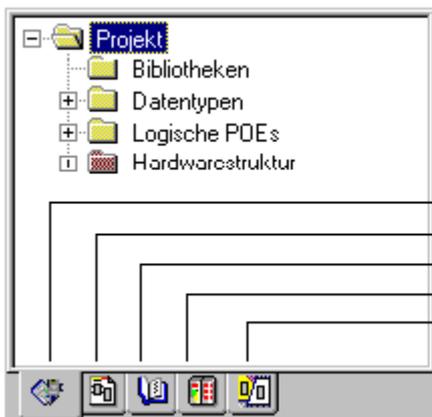


HINWEIS

Es empfiehlt sich, den Projektbaum während der Arbeit mit dem Programm ständig geöffnet zu lassen. Er verschafft Ihnen einen besseren Überblick, auf welcher Ebene Sie gerade arbeiten und was zu tun ist.

3.7.2 Projektbaum

Die Projektstruktur ist in vier Unterbäume gegliedert. Im Unterbaum 'Bibliotheken' können Sie Bibliotheken einbinden. In den Unterbäumen 'Datentypen' und 'Logische POEs' können Sie neue POEs und Arbeitsblätter zum Editieren von Datentypen und Variablen sowie Code-Arbeitsblätter einfügen. Im Unterbaum 'Hardwarestruktur' können Konfigurationselemente eingefügt, globale Variablen deklariert und Programme zu Tasks zugeordnet werden.



- Projekt - zeigt den gesamten Baum an
- POEs - zeigt die Unterbäume 'Datentypen' und 'Logische POEs'
- Bibliotheken - zeigt den Unterbaum 'Bibliotheken'
- Hardware - zeigt den Unterbaum 'Hardwarestruktur'
- Instanzen - zeigt den Instanzbaum an

Abbildung 23: Die Reiter des Projektbaums

Im Projektbaum können mehrere Objekte ausgewählt werden, d. h. Sie können mehrere Symbole markieren und auf alle markierten Symbole gleichzeitig denselben Befehl anwenden (z. B. Kopieren, Ausschneiden oder Einfügen von Symbolen, Drucken von Arbeitsblättern, globales Suchen/Ersetzen von Text, usw.). Sie markieren mehrere Objekte im Projektbaum, indem Sie entweder die <UMSCHALT>-Taste gedrückt halten und auf die gewünschten Symbole klicken (um aufeinanderfolgende (benachbarte) Symbole zu markieren) oder die <STRG>-Taste gedrückt halten und die gewünschten Symbole anklicken (um einzelne, nicht aufeinanderfolgende Symbole zu markieren).

Detaillierte Informationen zur Funktionalität des Projektbaums



finden Sie in der allgemeinen Online-Hilfe zum Programmiersystem im Thema 'Allgemeine Funktionen des Projektbaums'.

3.7.3 Instanzbaum

Der Instanzbaum zeigt die Projektstruktur in der SPS an und kann dazu verwendet werden, Arbeitsblätter im Online-Modus zu öffnen.

Sie können wahlweise den gesamten Projektbaum oder nur einzelne Unterbäume anzeigen. Zu diesem Zweck befinden sich an der Unterseite des Projektbaumes einige Reiter. Durch Anklicken eines Reiters wechseln Sie die Ansicht des Projektbaums.

3.7.4 Der graphische Editor - einfaches Programmieren in AS, FBS und KOP

Der graphische Editor ist einer der Editoren zum Programmieren der Programmcodes in den graphischen Arbeitsblättern der POEs. Das System unterstützt drei graphische Programmiersprachen: AS, FBS und KOP.

Die folgenden Leistungsmerkmale erleichtern Ihnen das Programmieren in den graphischen Sprachen:

- * Ein Editor-Assistent ermöglicht das einfache Einfügen und Ersetzen von Funktionen und Funktionsbausteinen.
- * Drag & Drop wird unterstützt, um Objekte mit der Maus zu bewegen und zu kopieren.
- * Einfache Bedienung mit der Tastatur, d. h. Sie können Elemente einfügen und den sichtbaren Bereich des Arbeitsblattes verschieben. Es können folgende Modi angewendet werden:
 - o Objektmodus:
Cursor-Tasten oder <STRG> + Cursor-Tasten
 - o Mauszeiger-Modus:
<UMSCHALT>-Taste + Cursor-Tasten oder <UMSCHALT> + <STRG> + Cursor-Tasten.
- * Duplizieren von Funktionseingängen direkt mit der Tastatur oder über die Symbolleiste oder Menüpunkte.
- * Negieren von Eingängen, Ausgängen, Kontakten und Spulen direkt mit der Tastatur oder über die Symbolleiste oder Menüpunkte.
- * Einfaches Autorouting.
- * Elemente können direkt in eine bestehende Verbindung oder an die Eingänge/Ausgänge bestehender Elemente eingefügt werden (nur in FBS).
- * Sie können Fenster splitten und Übersichtsfenster öffnen.
- * Elemente können beliebig angeordnet werden.
- * Gitter zum Anordnen von neu eingefügten Objekten.
- * Durch Doppelklicken auf das entsprechende Objekt im graphischen Arbeitsblatt oder mit Hilfe des Menüpunktes 'Objekt öffnen' im Kontextmenü des Objekts wird automatisch das Code-Arbeitsblatt von anwenderdefinierten POEs geöffnet.

Detaillierte Informationen zur Funktionalität des graphischen Editors



finden Sie in der allgemeinen Online-Hilfe zum Programmiersystem in den Themen 'Graphischer Editor' und 'Allgemeine Funktionen des graphischen Editors'.

Die verschiedenen graphischen Programmiersprachen können vermischt werden. Das System prüft alle Eingaben des Benutzers und entdeckt bzw. verhindert somit ungültige Verbindungen (z. B. die Verbindung zweier Ausgänge). Beim Einfügen neuer graphischer Elemente wird das Layout des bereits bestehenden Netzwerkes automatisch angepasst.

3.7.5 Der Texteditor - einfaches Programmieren in AWL und ST

Mit dem Texteditor können textuelle Programmcodes in den Arbeitsblättern der POEs programmiert werden. Das System unterstützt zwei textuelle Programmiersprachen: AWL und ST.

Der Texteditor wird auch zum Editieren der Datentyp-Arbeitsblätter und der Variablen-Arbeitsblätter verwendet.

Das Arbeiten mit dem Texteditor ist im Prinzip gleich wie bei einem üblichen ASCII-Editor. Die folgenden Leistungsmerkmale erleichtern Ihnen das Programmieren in den Sprachen AWL und ST:

- * Ein Editor-Assistent ermöglicht das einfache Einfügen und Ersetzen von vorbereiteten Datentypen, Operanden, Schlüsselwörtern, Funktionen und Funktionsbausteinen.
- * Drag & Drop wird unterstützt, um Textelemente mit der Maus zu bewegen und zu kopieren.
- * Es können Zeilennummern angezeigt werden.
- * Syntaxhervorhebung kann aktiviert werden.
- * Mehrfaches Rückgängig/Wiederherstellen ist möglich.
- * Mehrstufige Zoomfunktion.
- * Verschiedene Ansichten durch den Einsatz von Übersichtsfenstern.
- * Mit Hilfe des Menüpunkts 'FB öffnen' aus dem Kontextmenü des Instanznamens in einem textuellen Arbeitsblatt wird das Code-Arbeitsblatt von anwenderdefinierten Funktionsbausteinen geöffnet.

Detaillierte Informationen zur Funktionalität des Texteditors



finden Sie in der allgemeinen Online-Hilfe zum Programmiersystem in den Themen 'Texteditor' und 'Allgemeine Funktionen des Texteditors'.

3.7.6 Der Seitenlayout-Editor - Gestaltung von Seitenlayouts zum Drucken

Mit Hilfe des Seitenlayout-Editors können Sie in einfacher Weise Seitenlayouts zum Drucken Ihrer Projektdokumentation erstellen.

Ein Seitenlayout ist eine Vorlage, die beim Drucken des Inhalts eines Arbeitsblattes verwendet wird. Diese Vorlage definiert den Bereich, in dem der Programmcode gedruckt wird.

3.8 Der Editor-Assistent

Der Editor-Assistent ist ein hilfreiches Werkzeug, welches Sie beim Einfügen und Ersetzen folgender Elemente unterstützt:

- * Schlüsselwörter und Anweisungen (in ST-Arbeitsblättern).
- * Operatoren in AWL.
- * Funktionen und Funktionsbausteine (in allen Sprachen).

Der Editor-Assistent für die verschiedenen Sprachen und Arbeitsblätter ist in der folgenden Abbildung dargestellt.

Die prinzipielle Verwendung des Editor-Assistenten sieht wie folgt aus: Suchen Sie im Code-Arbeitsblatt die Position, an der ein neues Schlüsselwort, eine Anweisung, Funktion oder ein neuer Funktionsbaustein eingefügt werden soll. Setzen Sie an dieser Stelle den Textcursor bzw. die Einfügemarke. Anschließend suchen Sie im Auswahlbereich des Editor-Assistenten das einzufügende Element (Schlüsselwort, Anweisung, Funktion oder FB) und fügen es mit einem Doppelklick in den Programmcode ein.

Speziell im Texteditor bringt der Einsatz des Editor-Assistenten gravierende Vorteile mit sich:

- * Syntaktische Fehler (wie z. B. fehlende Strichpunkte, Auswahl- oder Wiederholungsanweisungen ohne END-Anweisung, usw.) werden vermieden, indem vorbereitete Anweisungen, Funktionen und Funktionsbausteine eingefügt werden, die durch Platzhalter ergänzt sind. Die Variablen und Werte erscheinen als Kommentare, die anschließend einfach überschrieben werden müssen.
- * Es ist nicht notwendig, dass der Anwender die Syntax der verschiedenen Anweisungen, wie z. B. von Funktionen oder Funktionsbausteinen kennt.

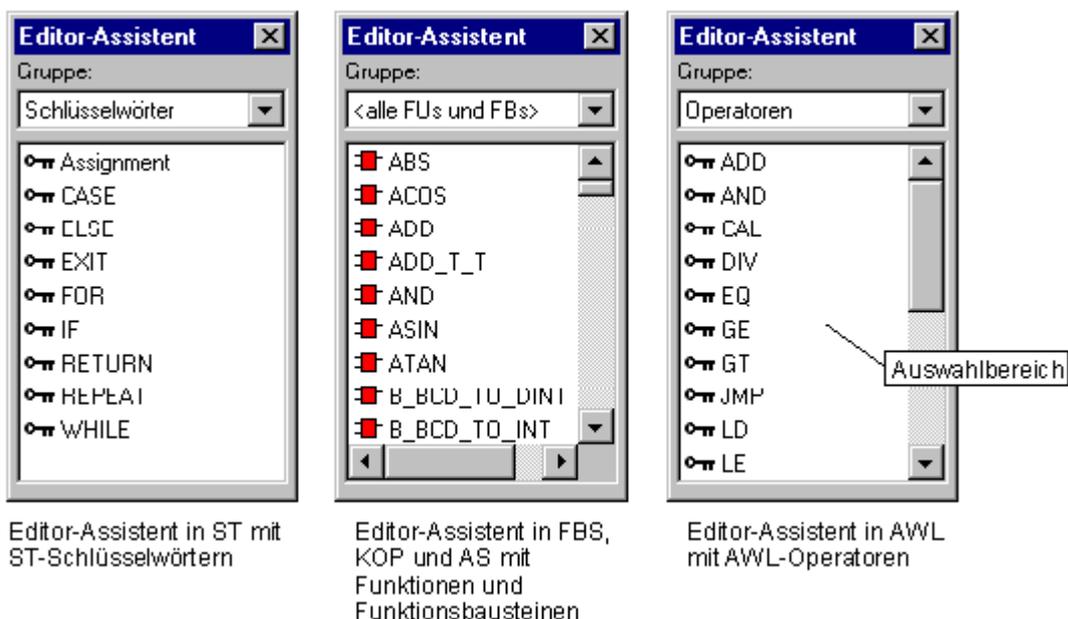


Abbildung 24: Der Editor-Assistent für die verschiedenen Sprachen

Sie können mit dem Editor-Assistenten auch Variablendefinitionen in die Variablen-Arbeitsblätter (aller Sprachen) und Datentypen in die Datentyp-Arbeitsblätter einfügen.

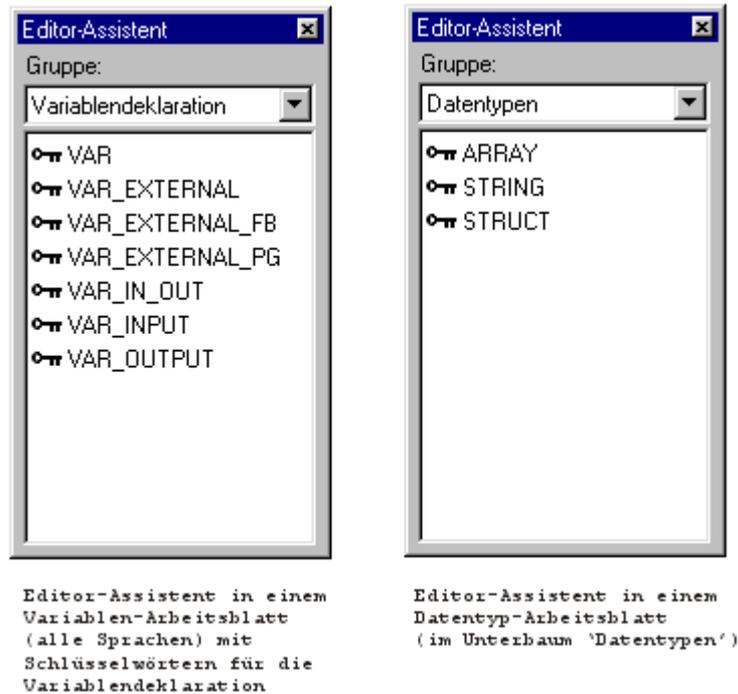


Abbildung 25: Der Editor-Assistent für Variablen- und Datentyp-Arbeitsblätter

Der andockbare Editor-Assistent ist in zwei Bereiche unterteilt: Das Listenfeld 'Gruppe' und der Auswahlbereich. Je nach eingestellter 'Gruppe' zeigt der Auswahlbereich die verschiedenen Elemente an, die in die Arbeitsblätter eingefügt werden können.

Das Listenfeld 'Gruppe' enthält alle verfügbaren Gruppen von Code-Elementen für die Sprache, mit der Sie gerade arbeiten. Der Auswahlbereich bietet alle Elemente an, die in der eingestellten 'Gruppe' zusammengefasst sind.

Abhängig von der Sprache, bietet das Listenfeld folgende Gruppen zur Auswahl an:

- * 'Schlüsselwörter', dargestellt durch das Schlüsselsymbol  (nur in ST).
- * 'Operatoren', dargestellt durch das Schlüsselsymbol  (nur in AWL).
- * '<alle FUs und FBs>', wobei Funktionen durch das Symbol  und FBs durch das Symbol  dargestellt werden.
Die Farbe eines Symbols deutet auf die Herkunft der jeweiligen Funktion oder des FB hin. FIRMWARE-Funktionen und -FBs werden in rot dargestellt. Bibliotheks-Funktionen und -FBs erscheinen blau. Anwenderdefinierte Funktionen und FBs sind grün.
- * 'Funktionsbausteine'. Der Auswahlbereich bietet nur Funktionsbausteine an. Die Symbole und deren Farben sind oben beschrieben.
- * 'Funktionen'. Der Auswahlbereich bietet nur Funktionen an. Die Symbole und deren Farben sind oben beschrieben.
- * 'SPS-spezifisch'. Der Auswahlbereich enthält Elemente, die nur für die verwendete SPS verfügbar sind.
- * 'PROZESSOR-spezifisch'. Der Auswahlbereich enthält Elemente, die nur für den verwendeten Prozessor verfügbar sind.

- * 'String FUs'. Der Auswahlbereich bietet nur Zeichenfolge-Funktionen an. Die Symbole und deren Farben sind oben beschrieben.
- * '*Projektname*'. Eine Gruppe trägt immer den Namen des aktuellen Projekts. Diese Gruppe enthält die benutzerdefinierten Funktionen und Funktionsbausteine, die im Rahmen des aktuellen Projekts erstellt wurden. Aus diesem Grund befinden sich in dieser Gruppe nur grüne Symbole.
- * 'Favoriten'. Diese Gruppe enthält alle Objekte, die Sie zuvor zu den Favoriten hinzugefügt haben.

Einfügen von Objekten in die Gruppe 'Favoriten'



Detaillierte Informationen zur Gruppe 'Favoriten' finden Sie im zugehörigen Thema der Online-Hilfe.

- * 'Typ-Konv. FUs'. Diese Gruppe enthält nur Funktionen für die Datentypumwandlung, wie zum Beispiel 'BOOL_TO_INT' oder 'BOOL_TO_WORD'.
- * '*Bibliotheksname*'. Jede eingebundene Bibliothek wird als eigene Gruppe angeboten.

Wenn der Editor-Assistent nicht geöffnet ist, führen Sie die folgenden Schritte durch:



Aufrufen des Editor-Assistenten mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Editor-Assistent'.



Aufrufen des Editor-Assistenten mit der Tastatur

- Drücken Sie <UMSCHALT> + <F2>.



Detaillierte Informationen zum Editor-Assistenten

Detaillierte Informationen zum Arbeiten mit dem Editor-Assistenten in den verschiedenen Sprachen finden Sie jeweils im Kapitel zu den einzelnen Sprachen.



Aufrufen der Hilfe für eine Funktion oder einen Funktionsbaustein aus dem Editor-Assistenten

Bevor Sie eine Funktion oder einen Funktionsbaustein in ein Code-Arbeitsblatt einfügen, können Sie die Hilfeseite zu dem gewünschten Objekt aufrufen.

Dazu markieren Sie die Funktion bzw. den Funktionsbaustein im Auswahlbereich des Editor-Assistenten und klicken mit der rechten Maustaste, um das zugehörige Kontextmenü zu öffnen. Wählen Sie anschließend den Menüpunkt 'Hilfe zu FU/FB'.

Die kontextsensitive Hilfeseite wird geöffnet. Die Hilfethemen sind für IEC-Standardfunktionen/FBs, SPS-spezifische und Firmware-Funktionen/FBs sowie für anwenderdefinierte Funktionen und Funktionsbausteine verfügbar.



Welches Hilfethema aufgerufen wird und wie Sie verschiedene Hilfeseiten für anwenderdefinierte Funktionen und Funktionsbausteine erzeugen, ist im Abschnitt 'Verwenden der Hilfe' in diesem Handbuch beschrieben. Weitere detaillierte Informationen finden Sie in der Online-Hilfe.



3.9 Das Übersichtsfenster für graphische Arbeitsblätter

Bei der Arbeit mit dem graphischen Editor können Sie ein Übersichtsfenster aufrufen, um einen Überblick über den gesamten Inhalt des aktuellen Arbeitsblattes zu erhalten. Dies erleichtert Ihnen die Orientierung und Navigation im Arbeitsblatt.



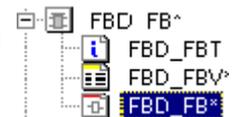
Aufrufen des Übersichtsfensters mit der Maus

- Öffnen Sie ein graphisches Arbeitsblatt, indem Sie auf das entsprechende Symbol im Projektbaum doppelklicken.
- Klicken Sie auf das Untermenü 'Layout'.
- Wählen Sie den Menüpunkt 'Übersichtsfenster'.
Es erscheint das Übersichtsfenster.



Aufrufen des Übersichtsfensters mit der Tastatur

- Öffnen Sie ein graphisches Arbeitsblatt im graphischen Editor. Dazu markieren Sie im Projektbaum das gewünschte Arbeitsblatt-Symbol und drücken <_J>.
- Drücken Sie <ALT> + <9>.
Es erscheint das Übersichtsfenster.



Das Übersichtsfenster zeigt Ihnen den gesamten Inhalt des aktuellen graphischen Arbeitsblattes an. Sie können den im graphischen Editor sichtbaren Ausschnitt des Arbeitsblattes mit Hilfe des Übersichtsfensters verschieben.



Navigieren im graphischen Arbeitsblatt mit Hilfe des Übersichtsfensters

- Klicken Sie in die Zeichnung im Übersichtsfenster. Neben dem Mauszeiger erscheint ein Handsymbol und der im Editor sichtbare Ausschnitt des Arbeitsblattes wird durch ein Rechteck dargestellt.
- Halten Sie die linke Maustaste gedrückt und ziehen Sie das Rechteck auf den Bereich, den Sie im Editor sehen wollen. Nachdem Sie die linke Maustaste losgelassen haben, erscheint im graphischen Editor der Ausschnitt des Arbeitsblattes, der im Übersichtsfenster vom Rechteck eingerahmt ist.

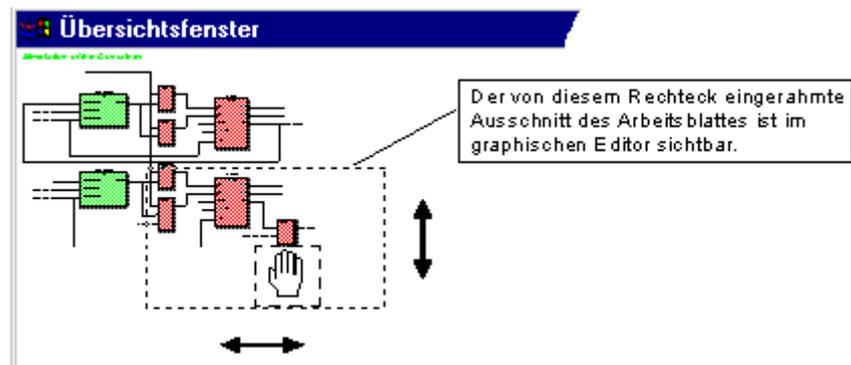


Abbildung 26: Navigieren im graphischen Arbeitsblatt mit Hilfe des Übersichtsfensters

3.10 Schließen von Arbeitsblättern

Wenn Sie das Editieren Ihres Arbeitsblattes beendet haben, können Sie das Arbeitsblatt schließen. Es erscheint ein Dialog, in dem Sie Ihre Änderungen speichern oder verwerfen können.



HINWEIS

Um alle geöffneten Arbeitsblätter gleichzeitig zu speichern und zu schließen, wählen Sie den Menüpunkt 'Alles schließen' aus dem Untermenü 'Fenster'.



Schließen eines Arbeitsblattes mit der Maus

- Klicken Sie auf das Untermenü 'Datei'.
- Wählen Sie den Menüpunkt 'Schließen'.
Es erscheint der Dialog 'PROPROG wt II'.



Schließen eines Arbeitsblattes mit der Tastatur

- Drücken Sie <STRG> + <F4>.
Es erscheint der Dialog 'PROPROG wt II'.



Abbildung 27: Der Dialog 'PROPROG wt II'

Verwenden des Dialogs 'PROPROG wt II'



- Klicken Sie auf 'Ja', um beim Schließen des Arbeitsblattes die Änderungen zu speichern.
- Klicken Sie auf 'Nein', um das Arbeitsblatt ohne Speichern der Änderungen zu schließen.
- Klicken Sie auf 'Abbrechen', um das Arbeitsblatt nicht zu schließen.

HINWEIS



Wenn Sie ein Arbeitsblatt schließen, wird es automatisch kompiliert.

3.11 Beenden des Programms

Sie können das Programm auch dann beenden, wenn eines oder mehrere Arbeitsblätter in den Editoren geöffnet sind. Wenn Sie die vorgenommenen Änderungen noch nicht gespeichert haben, erscheint ein Dialog, in dem Sie die Änderungen speichern oder die entsprechenden Fenster schließen können, ohne die Änderungen zu speichern.



Beenden des Programms mit der Maus

- Wählen Sie aus dem Untermenü 'Datei' den Menüpunkt 'Beenden'.
Das Programm wird geschlossen.



Beenden des Programms mit der Tastatur

- Drücken Sie <ALT> + <F4>.
Das Programm wird geschlossen.



HINWEIS

Um das Programm zu beenden, können Sie entweder auch auf das Symbol im Systemmenü doppelklicken (Symbol in der linken oberen Ecke des Programmfensters) oder Sie klicken auf das Schließen-Symbol rechts oben im Programmfenster.

ARBEITEN MIT PROJEKTEN

4.1 Erstellen eines neuen Projekts

Nachdem Sie das Programm gestartet haben, müssen Sie zuerst ein neues Projekt erstellen oder ein vorhandenes Projekt öffnen. In diesem Abschnitt wird beschrieben, wie Sie ein neues Projekt erstellen.

Sie haben zwei Möglichkeiten ein Projekt zu erstellen: Entweder mit Hilfe des Projekt-Assistenten (wird empfohlen) oder manuell über eine Vorlage. Beide Methoden sind in den folgenden Unterabschnitten beschrieben.

4.1.1 Erzeugen eines neuen Projekts mit Hilfe des Projekt-Assistenten

Am einfachsten lässt sich ein neues Projekt mit Hilfe des Projekt-Assistenten erzeugen, der Sie in 6 Schritten durch die Projekterstellung führt. So können Sie auf einfache Weise den Namen und den Typ des Projekts, die erste POE, die Konfiguration, die erste Resource und die erste Task definieren.



Aufrufen des Projekt-Assistenten mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Neues Projekt'. Es erscheint der Dialog 'Neues Projekt', den Sie in der folgenden Abbildung sehen können.
- Doppelklicken Sie im Dialog 'Neues Projekt' auf das Symbol 'Projekt-Assistent'. Es erscheint der Dialog 'Projekt-Assistent (Schritt 1 von 6)'.



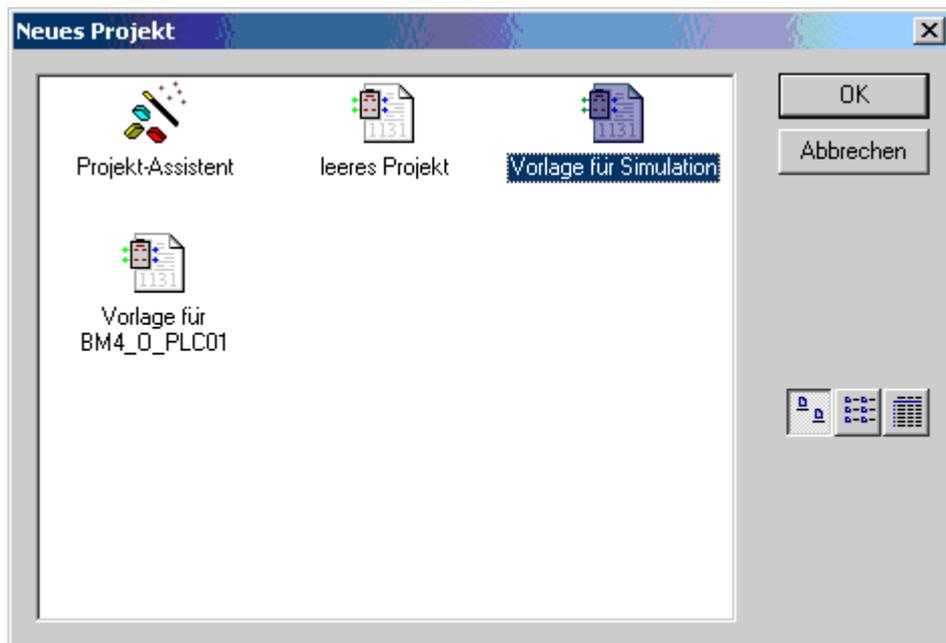


Abbildung 28: Dialog 'Neues Projekt' mit den verfügbaren Projekt-Vorlagen und dem Projekt-Assistenten



Aufrufen des Projekt-Assistenten mit der Tastatur

- Drücken Sie <STRG> + <N>.
Es erscheint der Dialog 'Neues Projekt'.
- Markieren Sie im Dialog 'Neues Projekt' mit Hilfe der Cursor-Tasten das Symbol 'Projekt-Assistent'.
- Drücken Sie <RETURN>, um den Dialog zu bestätigen.
Es erscheint der Dialog 'Projekt-Assistent (Schritt 1 von 6)'.



Verwenden des Projekt-Assistenten



- Geben Sie nun nacheinander die gewünschten Informationen in die verschiedenen Schrittdialoge ein und bestätigen Sie jeden Dialog mit der Tastenfläche 'Weiter'.
- Nachdem Sie den letzten Schritt (6 von 6) mit der Tastenfläche 'Fertig stellen' bestätigt haben, wird das neue Projekt automatisch erstellt und erscheint mit allen eingegebenen Daten im Projektbaum-Fenster.



Detaillierte Informationen zu den einzelnen Schrittdialogen finden Sie in der Online-Hilfe.

4.1.2 Erzeugen eines neuen Projekts mit Hilfe einer Projektvorlage

Bei dieser Methode kopiert das Programm die von Ihnen ausgewählte Projektvorlage in ein Projekt mit dem Namen 'Untitled'. Die Projektvorlage gibt POEs, Arbeitsblätter bzw. Konfigurationselemente, die für den entsprechenden SPS-Typ erforderlich sind, für das neue Projekt vor.

Sie können das Projekt 'Untitled' editieren und es anschließend unter einem beliebigen Namen speichern.



Erstellen eines neuen Projekts mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Neues Projekt'. Es erscheint der Dialog 'Neues Projekt' (siehe ► [Abbildung 28](#)◄ auf Seite 56).



Erstellen eines neuen Projekts mit der Tastatur

- Drücken Sie <STRG> + <N>. Es erscheint der Dialog 'Neues Projekt' (siehe ► [Abbildung 28](#)◄ auf Seite 56).



Verwenden des Dialogs 'Neues Projekt'

- Wählen Sie die Projektvorlage für Ihren SPS-Typ.
- Bestätigen Sie den Dialog. Das neue Projekt mit dem Namen 'Untitled' wird erstellt.



Ihr Projektbaum müsste nun ähnlich wie in der folgenden Abbildung aussehen:

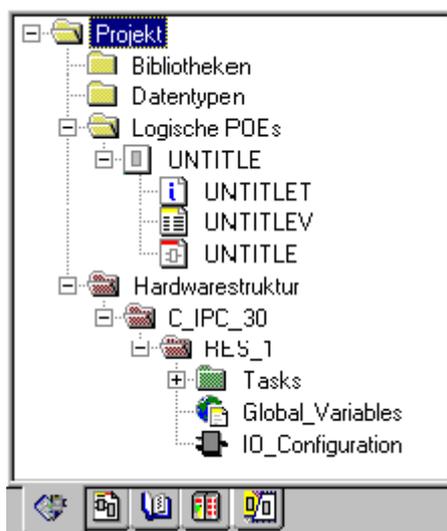


Abbildung 29: Projekt 'Untitled' mit Programm 'Untitle' und den entsprechenden Arbeitsblättern

4.2 Speichern eines bestehenden Projekts als Vorlage

Das Projekt 'Untitled' wird automatisch mit einer POE erstellt: Dem Programm 'Untitle'. Diese POE enthält 3 Arbeitsblätter:

- * Das Dokumentations-Arbeitsblatt 'UntitleT' zur Dokumentation der POE (optional).
- * Das Variablen-Arbeitsblatt 'UntitleV' zur Deklaration von Variablen und Funktionsbaustein-Instanzen.
- * Das Code-Arbeitsblatt 'Untitle' zur Definition des Programmcodes.



HINWEIS

Einige Eigenschaften des Programms können geändert werden. Die Standardsprache für das erste automatisch eingefügte Programm ist AWL, d. h. die Sprache kann nicht geändert werden. Wenn Sie für das Programm eine andere Sprache einsetzen möchten, müssen Sie eine neue POE einfügen.

4.2 Speichern eines bestehenden Projekts als Vorlage

Sie können ein bestehendes Projekt als Projektvorlage speichern. Nachdem Sie ein Projekt als Vorlage gespeichert haben, kann diese neue Vorlage im Dialog 'Neues Projekt' ausgewählt werden, um ein neues Projekt anzulegen.



Speichern eines bestehenden Projekts als Vorlage



- Wählen Sie aus dem Untermenü 'Datei' den Menüpunkt 'Als Vorlage speichern...'. Es erscheint der Dialog 'Als Vorlage speichern'.



Abbildung 30: Dialog 'Als Vorlage speichern'

- Geben Sie den gewünschten Namen für die neue Vorlage ein.
- Bestätigen Sie den Dialog.
Die neue Vorlage wird automatisch im dafür vorgesehenen Vorlagenverzeichnis gespeichert.

Löschen einer bestehenden Projektvorlage



Detaillierte Informationen zum Löschen von Projektvorlagen finden Sie in der Online-Hilfe zum System.

4.3 Ändern der Eigenschaften vorhandener POEs

Nehmen wir an, Sie wollen dem Programm 'Untitled' einen anderen Namen geben. Dazu müssen Sie die Eigenschaften dieses Programms ändern.

HINWEIS



Sie können die Eigenschaften aller Objekte im Projektbaum ändern, indem Sie aus dem Kontextmenü des Objekts die Option 'Eigenschaften...' wählen. Das Kontextmenü öffnen Sie, indem Sie das Objekt markieren und anschließend mit der rechten Maustaste klicken. Die verfügbaren Menüoptionen beziehen sich immer nur auf das markierte Objekt.

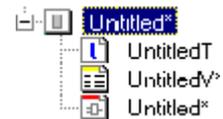
Ändern der Eigenschaften vorhandener POEs mit der Maus



- Wählen Sie aus dem Kontextmenü des Symbols '*Programmname*' die Menüoption 'Eigenschaften...'.
Es erscheint der Dialog 'Eigenschaften'.

Oder

- Klicken Sie im Projektbaum mit der rechten Maustaste auf das Symbol '*Programmname*', um das Kontextmenü zu öffnen.
- Wählen Sie den Menüpunkt 'Eigenschaften...'.
Es erscheint der Dialog 'Eigenschaften'.



Ändern der Eigenschaften vorhandener POEs mit der Tastatur



- Drücken Sie <↓> oder <↑>, um das Symbol '*Programmname*' zu markieren.
- Drücken Sie <ALT> + <↓>.
Es erscheint der Dialog 'Eigenschaften'.





Abbildung 31: Dialog 'Eigenschaften' zum Ändern der Eigenschaften vorhandener POEs



Verwenden des Dialogs 'Eigenschaften' zum Ändern der Eigenschaften vorhandener POEs



- Geben Sie bei Bedarf einen neuen Namen für die POE ein.
- Ändern Sie gegebenenfalls den POE-Typ. Die gewählte Sprache für diese POE kann hier aber nicht geändert werden.
- Bestätigen Sie den Dialog.



HINWEIS

Im Dialog 'Eigenschaften' können Sie auch den Namen eines bestimmten Arbeitsblattes ändern. Markieren Sie dazu das gewünschte Arbeitsblatt-Symbol im Projektbaum, klicken mit der rechten Maustaste zum Öffnen des Kontextmenüs und wählen den Menüpunkt 'Eigenschaften...'. Es erscheint der oben abgebildete Dialog 'Eigenschaften'. Geben Sie im Feld 'Name' einen neuen Namen für das Arbeitsblatt ein und bestätigen Sie dann den Dialog.

**HINWEIS**

Wenn Sie den Namen eines Programms geändert haben, müssen Sie auch den Namen des in der Task instanziierten Programms ändern. Andernfalls tritt beim Kompilieren des Projekts ein Fehler auf, da das System die umbenannte POE nicht finden kann.

Wie Sie ein Programm einer Task zuweisen wird detailliert im Abschnitt 'Zuweisen von Programmen zu Tasks' des Kapitels 'Kompilieren, Senden und Debuggen' beschrieben.

4.4 Einfügen neuer POEs

Der nächste Schritt, den Sie sicherlich vornehmen möchten, ist das Einfügen neuer POEs in unterschiedlichen Programmiersprachen.

**Einfügen einer POE mit der Maus**

- Um ein neues Programm einzufügen, klicken Sie auf das Symbol 'Programm hinzufügen'. Es erscheint der Dialog 'Einfügen'.
- Um einen neuen Funktionsbaustein einzufügen, klicken Sie auf das Symbol 'FB hinzufügen'. Es erscheint der Dialog 'Einfügen'.
- Um eine neue Funktion einzufügen, klicken Sie auf das Symbol 'Funktion hinzufügen'. Es erscheint der Dialog 'Einfügen'.

**Einfügen einer POE mit der Tastatur**

- Drücken Sie <↓> oder <↑>, um das Symbol 'Logische POEs' zu markieren.
- Drücken Sie <EINFG>. Es erscheint der Dialog 'Einfügen'.



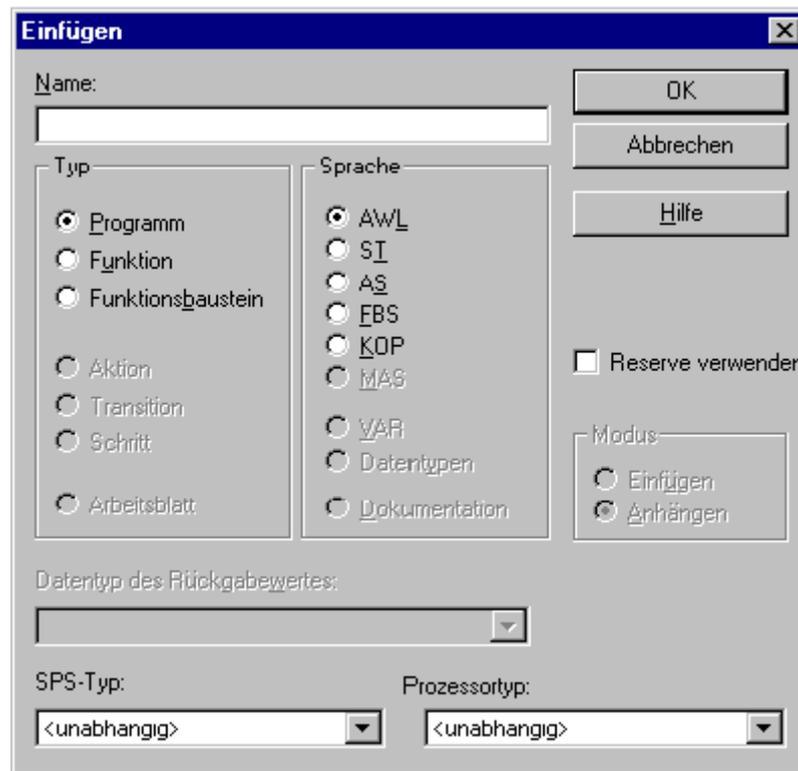


Abbildung 32: Dialog 'Einfügen' zum Einfügen einer neuen POE



Verwenden des Dialogs 'Einfügen' zum Einfügen einer neuen POE

- Geben Sie einen Namen für die neue POE ein.
- Wählen Sie den Typ der POE aus.
- Wählen Sie die Programmiersprache aus. Die Standardsprache für eine neu eingefügte POE ist FBS.
- Bestätigen Sie den Dialog.
Die neue POE mit den entsprechenden Arbeitsblättern wird in den Projektbaum eingefügt.



HINWEIS

Falls in Ihrem System eine Sprache nicht installiert ist, ist das zugehörige Optionsfeld im Dialog inaktiv und kann nicht ausgewählt werden.

Die neuen Arbeitsblätter werden im Projektbaum durch ein Sternchen gekennzeichnet. Die Sternchen signalisieren, dass die Arbeitsblätter neu eingefügt oder geändert, jedoch noch nicht kompiliert wurden.



HINWEIS

Wenn ein Bibliotheksbaustein in einer POE nicht sichtbar bzw. nicht auswählbar ist, kann das folgende Gründe haben:

- Der POE-Name hat den gleichen Namen wie der Bibliotheksbaustein, den man einfügen will.
Abhilfe: POE-Name ändern.
- Der SPS-Typ und der Prozessortyp der Bibliothek stimmen nicht mit der POE überein.
Abhilfe: Einbinden der für das Zielsystem richtigen Bibliothek bzw. Auswahl der richtigen Projektvorlage.
- Die Bibliothek ist beschädigt (es wurden z. B. Unterverzeichnisse auf der Festplatte gelöscht).
Abhilfe: Installieren Sie die Original-Bibliotheken von der CD.

4.5 Einfügen von Arbeitsblättern

Sie können neue Arbeitsblätter in POEs einfügen. Dies wird notwendig, wenn die Codes sehr groß sind und zur besseren Übersicht in mehrere Teile aufgedgliedert werden sollten. Beim Einfügen neuer Arbeitsblätter in POEs ist die Sprache für das neue Arbeitsblatt bereits durch die Sprache der POE festgelegt. Dies bedeutet, dass alle Arbeitsblätter einer POE dieselbe Programmiersprache (z. B. FBS) haben. Die Sprache eines neu eingefügten Arbeitsblattes wird dadurch automatisch festgelegt.



Einfügen eines Code-Arbeitsblattes in eine POE mit der Maus

- Klicken Sie im Projektbaum auf das Symbol des Code-Arbeitsblattes, dem ein neues Arbeitsblatt angehängt werden soll.
- Wählen Sie aus dem Kontextmenü den Menüpunkt 'Einfügen'.

Oder:

Klicken Sie auf das Symbol 'Objekt hinzufügen'.
Es erscheint jeweils der Dialog 'Einfügen'.



- Geben Sie einen Namen für das neue Arbeitsblatt in das Feld 'Name' ein.
- Aktivieren Sie entweder die Option 'Anhängen', um das neue Arbeitsblatt nach dem markierten Arbeitsblatt einzufügen oder die Option 'Einfügen', um das Arbeitsblatt vor dem markierten Arbeitsblatt einzufügen.
- Bestätigen Sie den Dialog 'Einfügen'.
Das neue Code-Arbeitsblatt wird in den Projektbaum eingefügt.



Einfügen eines Variablen-Arbeitsblattes in eine POE mit der Maus

- Klicken Sie im Projektbaum auf das Symbol des Variablen-Arbeitsblattes, dem ein neues Arbeitsblatt angehängt werden soll.
- Wählen Sie aus dem Kontextmenü den Menüpunkt 'Einfügen'.

Oder:

Klicken Sie auf das Symbol 'Objekt hinzufügen'.
Es erscheint jeweils der Dialog 'Einfügen'.



- Aktivieren Sie unter 'Sprache' die Option 'VAR'.
- Geben Sie einen Namen für das neue Arbeitsblatt in das Feld 'Name' ein.
- Aktivieren Sie entweder die Option 'Anhängen', um das neue Arbeitsblatt nach dem markierten Arbeitsblatt einzufügen oder die Option 'Einfügen', um das Arbeitsblatt vor dem markierten Arbeitsblatt einzufügen.
- Bestätigen Sie den Dialog 'Einfügen'. Das neue Variablen-Arbeitsblatt wird in den Projektbaum eingefügt.

Das Einfügen eines neuen Dokumentations-Arbeitsblattes erfolgt in gleicher Weise wie das Einfügen eines Code- oder Variablen-Arbeitsblattes. Der einzige Unterschied dabei ist, dass Sie vor dem Aufrufen des Dialogs 'Einfügen' ein vorhandenes Dokumentations-Arbeitsblatt im Projektbaum markieren müssen.

Die erforderlichen Schritte zum Einfügen neuer Arbeitsblätter für Datentypen im Unterbaum 'Datentypen' ähnelt ebenfalls den beschriebenen Schritten zum Einfügen neuer Code-Arbeitsblätter.



Einfügen eines Datentyp-Arbeitsblattes mit der Maus

- Klicken Sie im Projektbaum auf das Symbol 'Datentypen'.
- Wählen Sie aus dem Kontextmenü den Menüpunkt 'Einfügen'.

Oder:

Klicken Sie auf das Symbol 'Objekt hinzufügen'.
Es erscheint jeweils der Dialog 'Einfügen'.



- Geben Sie einen neuen Namen ein und bestätigen Sie den Dialog. Das neue Datentyp-Arbeitsblatt wird in den Unterbaum 'Datentypen' eingefügt.

Sie können auch jedes Arbeitsblatt mit der Tastatur im Projektbaum einfügen.



Einfügen eines Arbeitsblattes mit der Tastatur

- Drücken Sie <↓> oder <↑>, um das gewünschte Symbol zu markieren. Welches Symbol Sie markieren müssen, hängt davon ab, welchen Typ von Arbeitsblatt Sie einfügen möchten (siehe hierzu auch die oben aufgeführten Anweisungen).
- Drücken Sie <EINFG>. Es erscheint der Dialog 'Einfügen'.
- Stellen Sie die gewünschten Optionen ein und bestätigen Sie den Dialog 'Einfügen'.



4.6 Einbinden von Bibliotheken

Nachdem Sie ein Projekt abgeschlossen haben, können Sie die POEs und Arbeitsblätter dieses Projekts in einem neuen Projekt wiederverwenden. Dadurch ist es nicht mehr notwendig, Codes neu zu entwickeln, die schon in einem anderen Projekt vorhanden sind.

Anwender-Bibliotheken

Um POEs und Arbeitsblätter eines vorhandenen Projekts wiederzuverwenden, müssen Sie dieses Projekt als **Anwenderbibliothek** einbinden. Die Dateierweiterung für Anwenderbibliotheken ist *.mwt.



HINWEIS

Die Arbeitsblätter von POEs in eingebundenen **Anwenderbibliotheken** können mit Hilfe des entsprechenden Editors im Anzeigemodus geöffnet werden. 'Anzeigemodus' bedeutet, dass die Bibliothekenarbeitsblätter zwar angezeigt aber nicht geändert werden können.

Firmware-Bibliotheken

Sie können auch Firmware-Funktionen und -Funktionsbausteine verwenden, die von Ihrem SPS-Hersteller bereitgestellt wurden. Firmware-Funktionen und -Funktionsbausteine müssen als **Firmware-Bibliotheken** eingebunden werden. Die Dateierweiterung für Firmware-Bibliotheken ist *.fwl.



HINWEIS

Arbeitsblätter aus eingebundenen Firmware-Bibliotheken können weder im Anzeigemodus noch im Online-Modus geöffnet werden.

Im folgenden Abschnitt wird beschrieben, wie eine Anwenderbibliothek eingebunden wird. Zum Einbinden von Firmware-Bibliotheken wählen Sie einfach die Dateierweiterung '.fwl' aus.



Einbinden einer Bibliothek mit der Maus

- Klicken Sie auf das Symbol 'Bibliotheken', um es zu markieren.
 - Wählen Sie aus dem Kontextmenü den Menüpunkt 'Einfügen'.
- Oder:**



Klicken Sie auf das Symbol 'Objekt hinzufügen'.
Es erscheint jeweils der Dialog 'Bibliothek einbinden'.



- Wählen Sie das Projekt aus, das Sie als Bibliothek einbinden wollen.
- Klicken Sie auf die Schaltfläche 'OK', um den Dialog zu bestätigen.



Einbinden einer Bibliothek mit der Tastatur

- Drücken Sie <↓> oder <↑>, um das Symbol 'Bibliotheken' zu markieren.
- Drücken Sie <EINFG>.
Der Dialog 'Bibliothek einbinden' erscheint.
- Wählen Sie das Projekt aus, das Sie als Bibliothek einbinden wollen.
- Drücken Sie <↵>, um den Dialog zu bestätigen.



Ihr Projektbaum müsste nun ähnlich wie in der folgenden Abbildung aussehen:

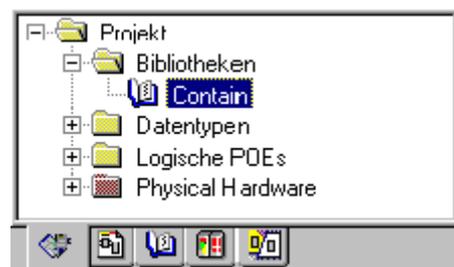


Abbildung 33: Projekt mit eingebundener Bibliothek 'Contain'



Öffnen eines Arbeitsblattes einer Anwenderbibliothek mit der Maus

- Markieren Sie das Bibliothekarbeitsblatt, das Sie öffnen wollen. Wenn das Arbeitsblatt-Symbol nicht sichtbar ist, doppelklicken Sie auf den Ordner 'Bibliotheken'.
- Suchen Sie die gewünschte Bibliothek und öffnen Sie den POE-Ordner.
- Doppelklicken Sie auf das gewünschte Arbeitsblattsymbol. Das Arbeitsblatt wird im entsprechenden Editor geöffnet. Der Inhalt kann nicht editiert werden.



**HINWEIS**

Da Arbeitsblätter von Anwenderbibliotheken im Anzeigemodus geöffnet werden können, können diese auch in den Online-Modus geschaltet werden, d. h. das Debuggen der Arbeitsblätter ist möglich.



Das Umschalten eines Bibliothekarbeitsblattes in den Online-Modus ist identisch mit dem Umschalten der anderen Arbeitsblätter im Projekt. Informationen hierzu finden Sie im Kapitel **Kompilieren, Senden und Debuggen** im Abschnitt **Aufrufen von Arbeitsblättern im Online-Modus**.

4.7 Löschen von Arbeitsblättern, POEs und Bibliotheken

Sie können Arbeitsblätter, vollständige POEs oder Bibliotheken löschen. Dazu müssen Sie jeweils zuerst das Objekt, das gelöscht werden soll, im Projektbaum markieren.

In den nächsten Schritten nehmen wir an, dass Sie die POE 'Untitled' löschen möchten.

**HINWEIS**

Beachten Sie beim Löschen einer vollständigen POE, dass alle Arbeitsblätter in dieser POE ebenfalls gelöscht werden.

**Löschen der POE 'Untitled'**

- Klicken Sie auf das Symbol der POE, die gelöscht werden soll.

Oder:

Drücken Sie <↓> oder <↑>, um die POE zu markieren, die gelöscht werden soll.

- Drücken Sie <ENTF>.

Es erscheint ein Dialog, indem Sie den Vorgang bestätigen oder abbrechen können.

- Bestätigen Sie den Dialog.
Die POE wird gelöscht.



4.8 Speichern von Änderungen in Arbeitsblättern beim Editieren



HINWEIS

Wenn Sie die Sicherheitsabfrage bestätigt haben, besteht keine Möglichkeit, die Daten wiederherzustellen. Sie sollten daher sehr vorsichtig mit der Löschfunktion umgehen!

4.8 Speichern von Änderungen in Arbeitsblättern beim Editieren

Beim Editieren von Arbeitsblättern sollten Sie die vorgenommenen Änderungen regelmäßig speichern. So vermeiden Sie, dass Ihnen bei einem Stromausfall oder bei anderen Zwischenfällen die geänderten Daten verlorengehen. Sie können die Änderungen im aktuellen Arbeitsblatt jederzeit speichern.



HINWEIS

Der Projektbaum wird automatisch gespeichert.



Speichern der Änderungen im aktuellen Arbeitsblatt mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Speichern'.
Das Arbeitsblatt wird gespeichert.



Speichern der Änderungen im aktuellen Arbeitsblatt mit der Tastatur

- Drücken Sie <STRG> + <S>.
Das Arbeitsblatt wird gespeichert.



HINWEIS

Automatisches Speichern

Das System verfügt über eine automatische Speicherfunktion für nicht gespeicherte Arbeitsblätter. Um diese Funktion einzuschalten und festzulegen, nach welchem Zeitraum das Speichern erfolgen soll, öffnen Sie den Dialog 'Optionen' und blättern dort auf die Seite 'Backup'. Aktivieren Sie im entsprechenden Dialogbereich die Option 'Alle' und geben Sie ein Zeitintervall ein.

Automatisches Kompilieren beim Schließen

Wenn Sie ein Arbeitsblatt speichern, wird dieses automatisch kompiliert, d. h. es wird eine Syntaxprüfung in diesem Arbeitsblatt durchgeführt. Alle Fehler die dabei entdeckt werden, erscheinen im Blatt 'Fehler' des Meldungsfensters. Außerdem wird das erste Variablen-Arbeitsblatt in dieser POE ebenfalls kompiliert. Aus diesem Grund ist es wichtig, dass Sie neu eingefügte I/O-Parameter im ersten Arbeitsblatt der POE deklarieren.

Durch das automatische Kompilieren ist unmittelbar nach dem Schließen des Arbeitsblattes jede anwenderdefinierte Funktion bzw. jeder FB im Editor-Assistent verfügbar.

4.9 Speichern eines Projekts unter einem neuen Namen

Sie können ein bestehendes Projekt unter einem neuen Dateinamen speichern. Dies bedeutet, dass alle Dateien, die zu diesem Projekt gehören (wie z. B. Projektdateien, die POEs, usw.) in einen neuen Projektordner kopiert werden. Das Originalprojekt wird dabei nicht geändert.



Aufrufen des Dialogs 'Projekt speichern/packen unter' mit der Maus

- Klicken Sie auf das Untermenü 'Datei'.
- Wählen Sie den Menüpunkt 'Projekt speichern unter / Projekt packen unter...'. Es erscheint der Dialog 'Projekt speichern/packen unter'.



Aufrufen des Dialogs 'Projekt speichern/packen unter' mit der Tastatur

- Drücken Sie <ALT> + <D>. Das Untermenü 'Datei' wird geöffnet.
- Markieren Sie mit <↓> oder <↑> den Menüpunkt 'Projekt speichern unter / Projekt packen unter...'.
Es erscheint der Dialog 'Projekt speichern/packen unter'.
- Drücken Sie <↵>.



Verwenden des Dialogs 'Projekt speichern/packen unter'



- Vergewissern Sie sich, dass im Dialog 'Projekt speichern/packen unter' im Listenfeld 'Dateityp' der Eintrag 'Projektdateien (*.mwt)' ausgewählt ist.
- Geben Sie in das Feld 'Dateiname' den neuen Dateinamen mit der Dateierweiterung 'mwt' ein. Der hier eingegebene Name wird für den neuen Projektordner und die neue Projektdatei verwendet.
- Wählen Sie bei Bedarf einen neuen Pfad aus.
- Bestätigen Sie den Dialog.

Es wird jetzt das gesamte Projekt in einen neuen Projektordner kopiert und die kopierte Projektdatei wird entsprechend umbenannt. Nach dem Speichern des neuen Projekts wird das Originalprojekt geschlossen. Das neue Projekt mit einem leeren Arbeitsbereich bleibt geöffnet.

4.10 Packen der Projektdateien in eine Archivdatei

Sie können das gesamte Projekt in einer Archivdatei packen. Dies erfolgt mit Hilfe des Dialogs 'Projekt speichern/packen unter'. Die Archivdatei enthält anschließend alle Dateien des aktuellen Projekts, d. h. die Projektdatei '*Projektname.mwt*', die Code-Dateien, die Variablendeklarationen und einige interne Dateien, die zum Wiederherstellen des Projekts aus der Archivdatei benötigt werden. Auf diese Weise kann die Archivfunktion zum Erstellen von Projekt-Backups verwendet werden.



HINWEIS

Es ist empfehlenswert, dass Sie Ihre Dateien regelmäßig, beispielsweise täglich, in einem Archiv packen und auf Diskette speichern. Dadurch wird gewährleistet, dass Ihnen keine Daten verlorengehen.

Neben dem manuellen Packen von Projektdateien in einer Archivdatei verfügt das System über eine automatische Backup-Funktion. Dadurch können Projekt-Backups automatisch in bestimmten Zeitintervallen oder beim Schließen des Projekts erzeugt werden. Detaillierte Informationen hierzu finden Sie in der Vorgehensweise am Ende dieses Abschnitts.



Packen der Projektdateien mit der Maus

- Klicken Sie auf das Untermenü 'Datei'.
- Wählen Sie den Menüpunkt 'Projekt speichern unter / Projekt packen unter...'. Es erscheint der Dialog 'Projekt speichern/packen unter'.
- Stellen Sie im Listenfeld 'Dateityp' den Eintrag 'Gepackte Projektdateien (*.zwt)' ein. Die Kontrollkästchen 'Anwender-Bibliotheken packen', 'Zwischencode (Frontend) packen' und 'Seitenlayout packen' können nun aktiviert bzw. deaktiviert werden.
- Geben Sie in das Feld 'Name' einen Namen für die Archivdatei mit der Erweiterung '.zwt' ein (z. B. contain.zwt).
- Aktivieren Sie das Kontrollkästchen 'Anwender-Bibliotheken packen', wenn Sie auch die Bibliothek packen wollen.
- Aktivieren Sie das Kontrollkästchen 'Zwischencode (Frontend) packen', wenn Sie auch die Kompilationsdateien packen wollen.
- Aktivieren Sie das Kontrollkästchen 'Seitenlayout packen', wenn Sie Seitenlayouts und Bitmaps packen wollen.
- Klicken Sie auf 'Packen', um den Vorgang zu starten. Der Verlauf des Prozesses wird in der Statuszeile angezeigt.
- Bestätigen Sie die Statusmeldung nach dem Packen.

**HINWEIS** zum Packen von Projekten und Bibliotheken:

Über die Kontrollkästchen (siehe oben) kann man im Einzelnen entscheiden, was im .zwt mitgepackt werden soll:

- „Anwender-Bibliotheken packen“
- „FW-Libraries packen“
- „Zwischencode (Frontend) packen“ (sollte immer aktiviert sein, wenn "Anwender-Bibliotheken packen" gewählt wurde)
- „Seitenlayouts packen“

Wenn alle Kontrollkästchen aktiviert sind, werden alle zum Projekt gehörenden Teile, also auch alle darin verwendeten Bibliotheken sowie die vom Anwender selbst angelegten Seitenlayouts, vollständig gepackt.

Bei sehr großen Projekten und täglich durchzuführenden Sicherungskopien bietet sich jedoch folgende praktische Vorgehensweise an: Die im Projekt verwendeten Bibliotheken und der eigentliche Anwendercode werden getrennt verwaltet.

Beispiel:

Der Anwender möchte sein Projekt "Prj.mwt" sinnvoll verwalten.

Vorgehensweise:

1. Der Anwender legt ein neues Projekt an (z. B. "PrjBIB.mwt") und fügt dort die gleichen Anwender- und FW-Bibliotheken sowie von ihm angelegte Seitenlayouts ein wie im Hauptprojekt "Prj.mwt" selbst. Dann speichert er das Bibliotheks-Projekt als .zwt, wobei er alle Kontrollkästchen aktiviert hat.

Besonders wichtig: Das Kontrollkästchen "Anwender-Bibliotheken packen" und "Zwischencode (Frontend) packen" muss gesetzt sein, damit der Bibliothekscode vollständig mitgepackt wird.

2. Für die tägliche Sicherung seiner Projektstände genügt dann das Packen seines Projektes "Prj.mwt" ohne aktivierten Kontrollkästchen, so dass das erzeugte .zwt bezüglich der Dateigröße sehr klein gehalten werden kann.

Schritt 1 braucht er nur dann wieder auszuführen, wenn neue Bibliotheken zugefügt oder vorhandene Bibliotheken geändert werden müssen.

PS: Die von Baumüller ausgelieferten Bibliotheken sind über die Namensvergabe (z. B. "UNIVERSAL_20bd00.mwt") einem bestimmten Versionsstand (im Beispiel "20bd00") fest zugeordnet und ändern sich daher nicht! Bei einem Update wird die Versionsnummer dann inkrementiert (z. B. auf "20bd01"). Alle Baumüller-Bibliotheken sind bei Auslieferung immer mit Zwischencode gepackt.

Somit ist sichergestellt, dass es weltweit immer nur die "eine" Bibliothek gibt.



HINWEIS für die Weitergabe bzw. Transfer von Projekten:

Wenn ein Anwender-Projekt ohne Zwischencode, aber mit Bibliotheken gepackt wird, werden die Bibliotheken ebenfalls ohne Code gepackt, auch wenn "Anwender-Bibliotheken packen" selbst angeklickt ist!

Damit sich beim anschließendem Entpacken auf einen anderen Rechner oder einem anderen PROPROG wt mit unterschiedlichen Bibliotheksverzeichnissen das Gesamtprojekt erfolgreich übersetzen lässt, müssen dort alle verwendeten Bibliotheken zuvor mit Zwischencode rückgesichert worden sein!

Beim Rücksichern des Projektes ohne Zwischencode dürfen diese Bibliotheken dann nicht wieder überschrieben werden!

Wenn "Anwender-Bibliotheken packen" gewählt wurde, sollte daher vom Anwender auch immer "Zwischencode (Frontend) packen" geklickt sein!



Detaillierte Informationen zum Dialog 'Projekt speichern/packen unter' finden Sie in der Online-Hilfe zum System. Zusätzlich wird dort beschrieben, wie Projektdateien wieder entpackt werden können.



Aktivieren der automatischen Backup-Funktion

Das System kann automatische Backups durchführen. Sie können einstellen, ob das System jedesmal beim Schließen eines Projekts oder nach einer festgelegten Zeitspanne automatisch eine Sicherungskopie des aktuellen Projekts erstellen soll.

Sie aktivieren die Backup-Funktion wie folgt:

- Öffnen Sie den Dialog 'Optionen', indem Sie im Untermenü 'Extras' den Menüpunkt 'Optionen' auswählen.
- Blättern Sie zur Seite 'Backup'.
- Stellen Sie im Bereich 'Backup-Projekt' die gewünschte Option ein und bestätigen Sie anschließend den Dialog 'Optionen'.



HINWEIS

Wichtig:

Wenn die automatische Backup-Funktion aktiviert ist, werden die manuell erstellten Sicherungskopien, die Sie mit Hilfe des Menüpunktes 'Projekt speichern unter / Projekt packen unter...' (Untermenü 'Datei') gepackt haben, **nicht** überschrieben. Statt dessen erzeugt das System eine Backup-Datei mit der Bezeichnung *Projektname_back.zwt*.

Wenn Sie das Kontrollkästchen „Anwender-Bibliotheken packen“ aktivieren, müssen Sie auch „Zwischencode (Frontend) packen“ anwählen, sonst werden die Bibliotheken nicht vollständig gepackt und Sie erhalten beim Rücksichern „Sternchen“ (Bibliothek nicht übersetzt).

4.11 Übersetzen der Projektsprache

Nehmen wir an, ein Projekt wurde in Deutsch entwickelt, d. h. alle Kommentare und Texte in den Dokumentations-Arbeitsblättern sind in deutscher Sprache verfasst. Nehmen wir weiter an, dass das Wartungspersonal einer Maschine dieses Projekt erweitern oder verändern muss, aber kein Deutsch kann.

Um solche Problemsituationen zu entschärfen, bietet das System ein Leistungsmerkmal an, das es ermöglicht, alle Kommentare und die beschreibenden Texte eines Projekts in jede beliebige Landessprache zu übersetzen.

Der Übersetzungsprozess läuft im wesentlichen in vier Schritten ab.

1. Exportieren aller übersetzbaren Texte des Projekts in eine Übersetzungsdatei mit der Dateierweiterung 'txt'. Diese Datei enthält dann alle Kommentare und die Inhalte der Dokumentations-Arbeitsblätter.
2. Übersetzen der Projektübersetzungsdatei. Die Übersetzung kann entweder manuell erfolgen oder unter Einsatz spezieller Übersetzungsprogramme, beispielsweise mit Datenbankunterstützung.
3. Auswählen der neuen Projektsprache und (re-)importieren der übersetzten txt-Datei in das Programmiersystem.
4. Umschalten der Projektsprache im Dialog 'Projekteigenschaften'.

4.11.1 Exportieren einer Projektübersetzungsdatei

Um die Landessprache eines Projekts übersetzen zu können, müssen zunächst alle übersetzbaren Texte in eine sogenannte Projektübersetzungsdatei exportiert werden. Zu den übersetzbaren Texten gehören die Kommentare und die Inhalte der Dokumentations-Arbeitsblätter. Selbstverständlich können programmiersprachenspezifische Textelemente nicht übersetzt werden.

Bei der Übersetzungsdatei handelt sich um eine übliche ASCII-Datei mit der Erweiterung 'txt'. Die Datei kann mit jedem ASCII-Editor geöffnet und bearbeitet werden. Somit ist es auch möglich, handelsübliche Übersetzungsprogramme für die Übersetzung der Datei in die Zielsprache zu verwenden.

HINWEIS



Vor dem Exportieren einer Projektübersetzungsdatei muss im Dialog "Projekteigenschaften" die Standardsprache ausgewählt werden. Nähere Informationen dazu finden Sie im Kapitel "Umschalten der Projektsprache".



Exportieren einer Projektübersetzungsdatei

- Öffnen Sie das Untermenü 'Datei' und wählen Sie den Menüpunkt 'Export...' aus. Es erscheint der Dialog 'Import / Export', der alle Systemerweiterungen anzeigt, die zum Exportieren von Dateien aus dem Projekt installiert sind.

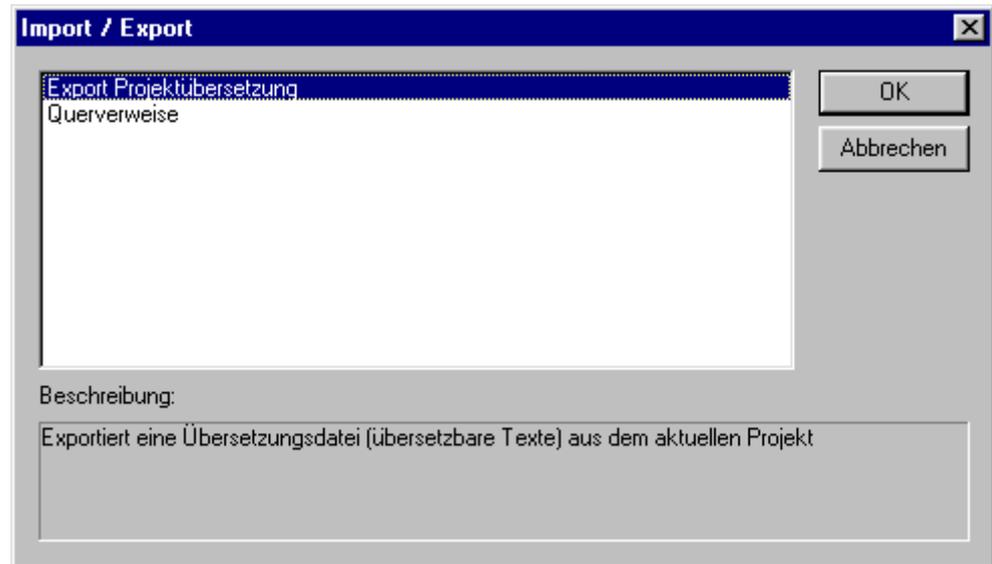


Abbildung 34: Dialog 'Import / Export' (nach Auswählen des Menüpunkts 'Export...')

- Markieren Sie im Auswahlfeld den Eintrag 'Export Projektübersetzung' (wie in der Abbildung gezeigt) und bestätigen Sie den Dialog. Es erscheint der Suchdialog 'Export Projektübersetzung'.

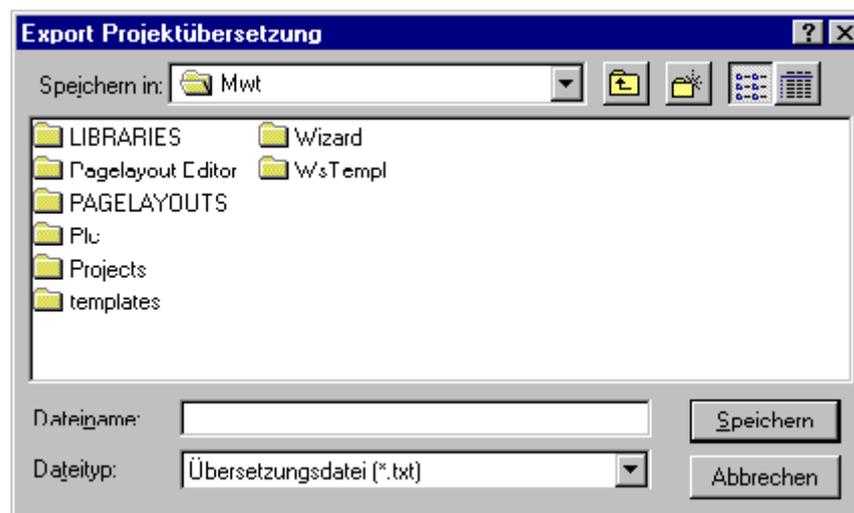


Abbildung 35: Dialog 'Export Projektübersetzung'

- Wechseln Sie im Dialog 'Export Projektübersetzung' in das gewünschte Zielverzeichnis und geben Sie den Namen der Exportdatei in das Dialogfeld 'Dateiname' ein.
- Bestätigen Sie den Dialog.
Das System exportiert nun alle übersetzbaren Texte aus dem Projekt in die Projektübersetzungsdatei.

Im nächsten Schritt muss die Exportdatei geöffnet und in die Zielsprache übersetzt werden (entweder manuell oder mit Hilfe einer Übersetzungssoftware).

4.11.2 Importieren einer Projektübersetzungsdatei

Nachdem Sie die Projektübersetzungsdatei in die gewünschte Landessprache übersetzt haben, müssen Sie zuerst die Sprache der zu (re-)importierenden Datei einstellen und dann die übersetzte Datei importieren. Erst dann kann die Projektsprache umgeschaltet werden.



Importieren einer Projektübersetzungsdatei

- Öffnen Sie das Untermenü 'Datei' und wählen Sie den Menüpunkt 'Import...!' aus. Es erscheint der Dialog 'Import / Export', der alle Systemerweiterungen anzeigt, die zum Importieren von Dateien in das Projekt installiert sind.

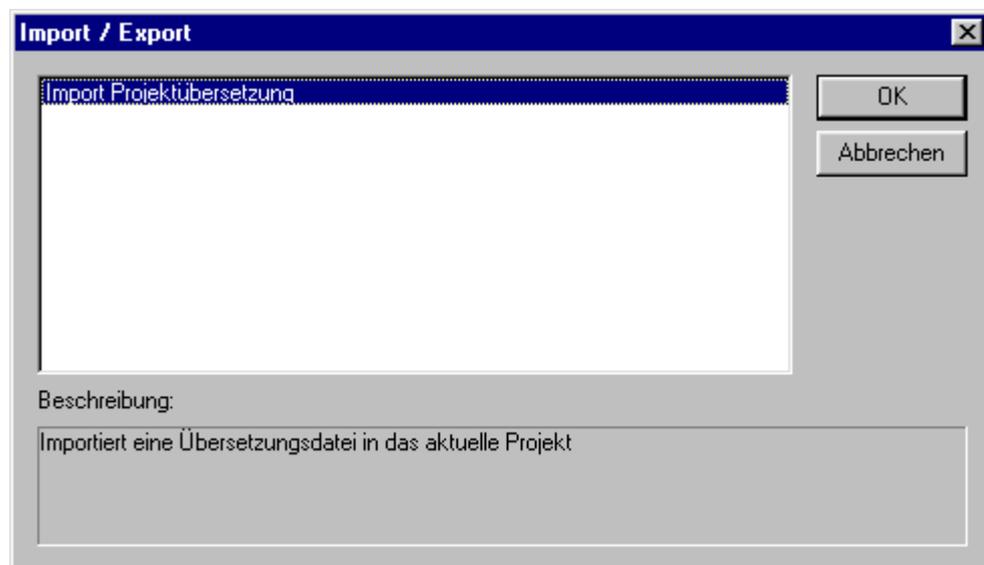


Abbildung 36: Dialog 'Import / Export' (nach Auswählen des Menüpunkts 'Import...!')

- Markieren Sie im Auswahlfeld den Eintrag 'Import Projektübersetzung' (wie in der Abbildung gezeigt) und bestätigen Sie den Dialog. Es erscheint der Suchdialog 'Import Projektübersetzung'.

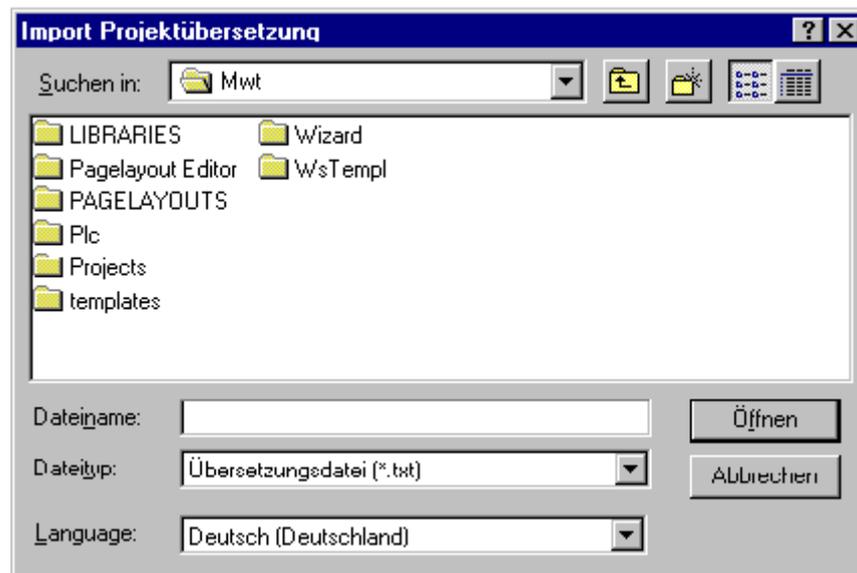


Abbildung 37: Dialog 'Import Projektübersetzung'

- Wählen Sie im Dialog 'Import Projektübersetzung' die Sprache der zu importierenden Textdatei aus. Zu diesem Zweck öffnen Sie das Listenfeld 'Sprache' und wählen den gewünschten Eintrag aus.



HINWEIS

Die hier ausgewählte Sprache ist anschließend im Eigenschaftsdialog des Projekts verfügbar, in dem Sie später die Sprache umschalten.

- Gehen Sie zum gewünschten Verzeichnis und wählen Sie die Datei, die importiert werden soll.
- Bestätigen Sie den Dialog. Die Projektübersetzungsdatei wird in das Projekt importiert. Nachdem der Import erfolgreich abgeschlossen ist, erhalten Sie eine entsprechende Meldung.

Die neue Landessprache ist nun zwar im Projekt verfügbar, wird aber nicht automatisch aktiviert. Der nächste Schritt ist deshalb der Aufruf des Dialogs 'Projekteigenschaften' und das Umschalten der Projektsprache.

4.11.3 Umschalten der Projektsprache

Nachdem Sie die Projektübersetzungsdatei importiert haben, können Sie die Projektsprache umschalten.



Umschalten der Projektsprache

- Öffnen Sie im Projektbaum das Kontextmenü des Ordners 'Projekt', indem Sie das Symbol mit der rechten Maustaste anklicken.
- Wählen Sie im Kontextmenü den Menüpunkt 'Eigenschaften...'. Es erscheint der Dialog 'Projekteigenschaften'.



Abbildung 38: Dialog 'Projekteigenschaften'

- Öffnen Sie im Eigenschaftsdialog das Listenfeld 'Sprache' und wählen den gewünschten Eintrag aus.

HINWEIS



Das Listenfeld enthält nur Sprachen, die vor dem Reimport der übersetzten Textdatei im Dialog 'Import Projektübersetzung' eingestellt wurden.

Die Projektsprache wird nun auf die gewählte Sprache umgeschaltet.

4.12 Konvertierung des Quell-Codes für AWL, FBS und KOP

Zusätzlich zu den Anforderungen, die die IEC 61131-3 an das System stellt, bietet das Programmiersystem die Möglichkeit der 'Konvertierung des Quell-Codes'. Dadurch können die Quelltexte einer POE, die in einer der Programmiersprachen AWL, FBS oder KOP geschrieben sind, in die jeweils anderen Sprachen konvertiert werden.

Die folgende Tabelle zeigt die verschiedenen Möglichkeiten der Konvertierung des Quell-Codes:

Ausgangssprache	Mögliche Zielsprache
AWL	FBS, KOP
FBS	AWL, KOP
KOP	AWL, FBS

Voraussetzung für jede Konvertierung des Quell-Codes ist, dass das Projekt mit der zu konvertierenden POE zuvor kompiliert wurde (entweder mit dem Befehl 'Make' oder 'Projekt neu erzeugen' im Untermenü 'Code'). Dies ist erforderlich, da das System nicht die in den Arbeitsblättern angezeigten textuellen oder grafischen Sprachelemente konvertiert, sondern den Zwischencode, der beim Kompilieren des Projekts erzeugt wird.



Konvertieren der Ausgangssprache einer POE

- Stellen Sie sicher, dass das Projekt zuvor kompiliert wurde.
- Öffnen Sie im Projektbaum den Unterbaum 'Logische POEs' und markieren Sie das Symbol der POE, die konvertiert werden soll.
- Klicken Sie mit der rechten Maustaste auf das Symbol, um das Kontextmenü zu öffnen.
- Wählen Sie den Menüpunkt 'Konvertierung des Quell-Codes'. Es erscheint der Dialog 'Konvertierung des Quell-Codes'.

Beispiel:

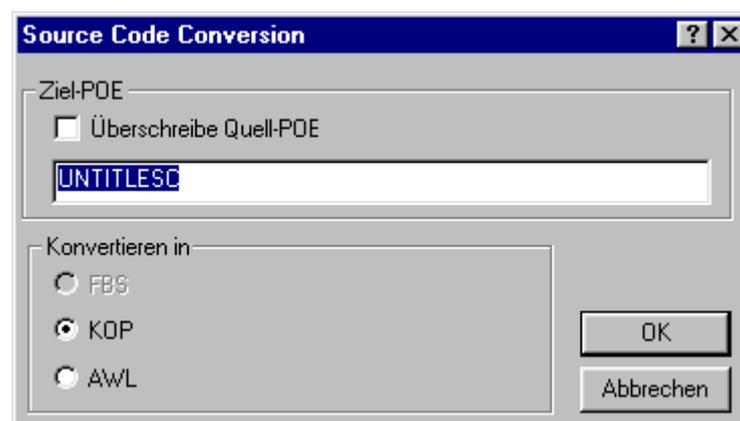
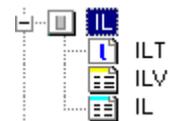


Abbildung 39: Dialog 'Konvertierung des Quell-Codes'

- Aktivieren Sie bei Bedarf das Kontrollkästchen 'Überschreibe Quell-POE'. Wenn aktiviert, wird die Quell-POE durch die konvertierte POE überschrieben. Ansonsten wird die Quell-POE nicht geändert, d. h. Sie kann weiterhin verwendet werden.
- Geben Sie den Namen für die Ziel-POE ein. Nach dem Öffnen des Dialogs wird vom System automatisch ein Name vorgeschlagen. Sie können diesen Namen entweder übernehmen oder einen neuen Namen eingeben.
- Wählen Sie die gewünschte Zielsprache.
- Bestätigen Sie den Dialog.
Entsprechend Ihren Einstellungen wird die Quell-POE konvertiert und die neue POE in den Projektbaum eingefügt.

HINWEIS

Die neue POE ist nach dem Konvertieren noch nicht kompiliert.

LITERALE, DATENTYPEN UND VARIABLEN

5.1 Literale

Literale werden verwendet, um externe Darstellungen von Daten einzugeben. Sie sind für die Darstellung von numerischen Daten, Zeichenfolgen und Zeitdaten erforderlich. Beim Eingeben von Werten müssen Sie immer Literale verwenden.

5.1.1 Numerische Literale

Die verwendbaren numerischen Literale sind in der folgenden Tabelle dargestellt:

Typ	Beispiele
Integer-Literale	-12 0 123_456 +986
Gleitkomma-Literale	-12.0 0.0 0.4560 3.14159_26
Gleitkomma-Literale mit Exponenten	-1.34E-12 -1.34e-12 1.0E+6
Basis-2-Literale	INT#2#1111_1111
Basis-8-Literale	INT#8#377
Basis-16-Literale	INT#16#FF SINT#16#ff
Boolesche Werte FALSE und TRUE	FALSE TRUE
Boolesche Werte 0 und 1	0, 1

Abbildung 40: Numerische Literale



HINWEIS

Literale in Variablen-Arbeitsblättern und Literale für die Datentypen INT oder BOOL können ohne Schlüsselwort verwendet werden, wie in den folgenden Beispielen dargestellt:

Für INT#16#ff ist folgende Schreibweise möglich: 16#ff

Für BOOL#FALSE ist folgende Schreibweise möglich: FALSE.



HINWEIS

In Variablen-Deklarationen ist folgende Schreibweise möglich:

```
"var1 : DINT :=10"
```

Im Code dagegen müssen Sie die Schreibweise "LD DINT#10" verwenden.

5.1.2 Zeichenfolge- (String-) Literale

Ein Zeichenfolge-Literal ist eine Sequenz von null oder mehr Zeichen, die durch zwei einfache Anführungszeichen eingeschlossen sind.

Die verwendbaren Zeichenfolge-Literale sind in der folgenden Tabelle dargestellt:

Typ	Beispiele
Leere Zeichenfolge	"
Zeichenfolge mit einem Leerzeichen	' '
Zeichenfolge mit Text	'dies ist ein Text'

Abbildung 41: Zeichenfolge- (String-) Literale

5.1.3 Zeitdauer-Literale

Zeitdauer-Literale können in Stunden, Minuten, Sekunden, Millisekunden und in Kombination dieser Angaben dargestellt werden.

Die verwendbaren Zeitdauer-Literale sind in der folgenden Tabelle dargestellt:

Typ	Beispiele
Kurzes Präfix	T#14ms t#14ms t#12m18s3ms T#25h_15m t#25h_15m
Langes Präfix	TIME#14ms time#14ms TIME#25h_15m time#25h_15m

Abbildung 42: Zeitdauer-Literale



HINWEIS

Bei der Angabe von Millisekunden (ms) werden Kommastellen ignoriert.

D. h. TIME#15,8ms wird als TIME#15ms interpretiert.

5.2 Die IEC-Datentypen

Datentypen definieren die Anzahl der Bits (Bitgröße), den Wertebereich und den Anfangswert einer Variablen. Sie können elementare und anwenderdefinierte Datentypen verwenden. Generische Datentypen werden in erster Linie verwendet, um die verfügbaren Datentypen für Eingänge und Ausgänge von Funktionen zu beschreiben.

5.2.1 Elementare Datentypen

Die Wertebereiche und die Bitgröße von elementaren Datentypen sind in der IEC 61131-3 beschrieben.

Die elementaren Datentypen sind in der folgenden Tabelle dargestellt:

Datentyp	Beschreibung	Größe	Bereich	Std.-Anfangswert
BOOL	Boolesch	1	0 und 1	0
SINT	8-Bit-Integer	8	-128 bis 127	0
INT	Integer	16	-32768 bis 32767	0
DINT	Double Integer	32	-2.147.483.648 bis +2147.483.647	0
USINT	8-Bit-Integer ohne Vorzeichen	8	0 bis 255	0
UINT	Integer ohne Vorzeichen	16	0 bis 65535	0
UDINT	Double Integer ohne Vorzeichen	32	0 bis 4.294.967.295	0
REAL	Gleitkomma-Zahl	32	1.18×10^{-38} bis 3.40×10^{38}	0.0
TIME	Zeitdauer	32	über 24 Tage	t#0s
BYTE	Bitfolge der Länge 8	8	16#00 bis 16#FF	0
WORD	Bitfolge der Länge 16	16	16#0000 bis 16#FFFF	0
DWORD	Bitfolge der Länge 32	32	16#00000000 bis 16#FFFFFFFF	0

Abbildung 43: Elementare Datentypen

HINWEIS



Die Verwendung von Datentypen hängt auch von Ihrer Hardware ab. Informationen über die unterstützten Datentypen finden Sie in Ihrer Hardware-Dokumentation.



HINWEIS

Der Datentyp `STRING` ist zwar ein elementarer Datentyp, gehört aber nicht zu der oben angegebenen Gruppe. Sein Format im Speicher hängt vom SPS-Typ ab.

Der Datentyp `STRING` hat folgende Struktur:

Byte 0-1	Offset zur maximalen Länge (0 entspricht 80)
Byte 2-3	aktuelle Länge
Byte 4-83	Zeichen
Byte 84	Null-Terminator

5.2.2 Generische Datentypen

Generische Datentypen sind Datentypen, in denen Gruppen elementarer Datentypen zusammengefasst sind. Beispielsweise beinhaltet `ANY_INT` die elementaren Datentypen `DINT`, `INT`, `SINT`, `UDINT`, `UINT` und `USINT`. Wenn eine Funktion mit `ANY_INT` verbunden werden kann, bedeutet das, dass sie Variablen der Datentypen `DINT`, `INT`, `SINT`, `UDINT`, `UINT` und `USINT` verarbeiten kann.

Die folgende Liste zeigt die generischen Datentypen:

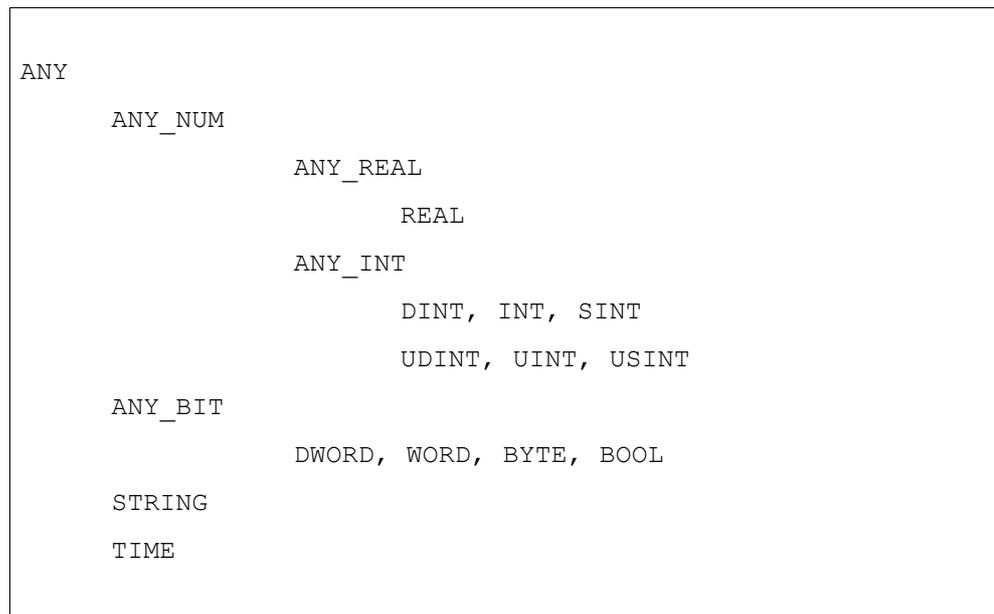


Abbildung 44: Generische Datentypen



HINWEIS

Die Verwendung von Datentypen hängt auch von Ihrer Hardware ab. Bitte schlagen Sie zu Einschränkungen bei der Verwendung von Datentypen in der Dokumentation zu Ihrer Hardware nach.

5.2.3 Anwenderdefinierte Datentypen

Anwenderdefinierte Datentypen können deklariert werden, um zusätzlich zu den in der IEC 61131-3 definierten elementaren Datentypen neue Datentypen zu erstellen. Durch die Definition eigener Datentypen mit entsprechenden Eigenschaften können Sie die Programmierarbeit wesentlich einfacher und effizienter gestalten. Anwenderdefinierte Datentypen werden auch abgeleitete Datentypen genannt.

Sie können drei verschiedene anwenderdefinierte Datentypen deklarieren:

- * Datentypen für Felder
- * Datentypen für Strukturen
- * Datentypen für Zeichenfolgen



Informationen über verfügbare anwenderdefinierte Datentypen und ihre Einschränkungen finden Sie in Ihrer SPS-Dokumentation. Es kann sein, dass Ihre SPS nicht alle beschriebenen Möglichkeiten zur Verwendung von anwenderdefinierten Datentypen unterstützt!

5.3 Datentypen für Felder

5.3.1 Deklarieren von Feldern

Felder setzen sich aus mehreren Elementen vom gleichen Datentyp zusammen. Auf die einzelnen Elemente eines Felds kann mit einem Feldindex zugegriffen werden. Ein Feld kann verwendet werden, um mehrere Elemente des gleichen Typs in einer einzigen Zeile der Typendeklaration zu deklarieren.

```
TYPE
    graph      :      ARRAY[1..23] OF INT;
END_TYPE
```

Abbildung 45: Deklaration eines Datentyps für Felder

In diesem Beispiel beinhaltet das Feld 'graph' 23 Elemente vom Datentyp 'INT'. Alle Elemente eines Felds werden fortlaufend nacheinander im SPS-Speicher abgelegt.

HINWEIS



Um ein Feld zu deklarieren, muss dieses zuerst in der Datentyp-Deklaration des Datentyp-Arbeitsblatts deklariert werden. Danach kann die Variable im Variablen-Arbeitsblatt der POE deklariert werden (siehe hierzu das folgende Programmierbeispiel).

5.3.2 Programmierbeispiel

Ein Feld sollte typischerweise für Daten verwendet werden, die die gleiche Thematik beschreiben. Nehmen wir einmal an, dass ein Prozess alle drei Sekunden einen Eingangswert ändert. Jeder dieser Eingangswerte muss gespeichert werden, um ihn mit dem jeweiligen Sollwert zu vergleichen. Alle Eingangswerte sind vom gleichen Datentyp. In diesem Fall ist die Deklaration eines Felds sinnvoll, da in der Code-Deklaration zwei Werte mit Hilfe einer Wiederholungsanweisung einfach verglichen werden können (z. B. in einer FOR-Schleife). Auf die einzelnen Komponenten eines Felds kann mit dem Feldindex zugegriffen werden.

```
Datentyp-Deklaration:
TYPE
    graph      :   ARRAY[1..23] OF INT;
    set_point  :   ARRAY[1..23] OF INT;
END_TYPE

Variablen-Deklaration:
VAR
    eingang    :   graph; (* Eingangswerte der
                    Maschine *)
    werte      :   sollwert; (* Vergleichswerte *)
    i          :   INT :=1; (* Variable für Feldindex *)
    betrieb    :   BOOL :=TRUE;
    ERROR      :   BOOL;
    Zeitgeber  :   FB_TIMER; (* FB-Instanz
                    deklarieren *)
END_VAR

Code-Deklaration in ST:
zeitgeber (pt:=t#3s;in:=betrieb);
IF zeitgeber.Q THEN (* Eingangswerte an Feld 'graph'
                    liefern *)
    eingang[i] := %IW0; (* Eingangswert dem Feld zuweisen
*)
    betrieb := 0; (* Flankenerkennung, um Zeitgeber
                  erneut zu starten *)
    i := i+1; (* höherer Feldindex *)
ELSE
    betrieb :=1; (* aufwärts zählen *)
END_IF;
IF i = 23
    FOR i:=1 TO 23 BY 1 DO
        IF eingang[i] <> werte[i] THEN (* Feld
            'graph' mit 'sollwert' vergleichen*)
            ERROR := TRUE;
        END_IF;
    END_FOR;
    i := 1;
END_IF;
```

Abbildung 46: Programmierbeispiel für Felder

5.3.3 Mehrdimensionale Felder

Mehrdimensionale Felder können als Felder von Feldern realisiert werden. Ein Beispiel für ein Feld eines Felds ist in der folgenden Abbildung dargestellt:

```
TYPE
    graph      :      ARRAY [1..10] OF INT;
    mein_feld  :      ARRAY [1..3] OF graph;
END_TYPE
```

Abbildung 47: Deklaration eines Felds eines Felds

Auf ein einzelnes Element in einem Feld eines Felds kann mit zwei Indizes zugegriffen werden, wie in der folgenden Abbildung dargestellt.

```
Variablen-Deklaration:
VAR
    var1      :      mein_feld;
    var2      :      INT;
END_VAR

Code-Deklaration in ST:
var2 := var1[1][3];
```

Abbildung 48: Zugreifen auf Elemente eines Felds eines Felds

5.3.4 Initialisieren von Feldern

Felder können initialisiert werden, d. h. jedem Element eines Felds kann ein Anfangswert zugewiesen werden. Wie bereits erwähnt, wird auf jedes einzelne Feldelement mit einem Index zugegriffen. Felder können beim Editieren der Code-Deklaration initialisiert werden.

```
Variablen-Deklaration:
VAR
    graph      :      ARRAY [1..10] OF INT;
END_VAR

Code-Deklaration in ST:
graph[1] :=7;
graph[2] :=1092;
.
.
.
graph[10] :=13;
```

Abbildung 49: Initialisieren von Feldelementen

Es müssen nicht alle Elemente eines Felds initialisiert werden. Wenn kein Anfangswert verwendet wird, wird das Feldelement beim Starten der Programmausführung mit dem Standard-Anfangswert initialisiert.

5.4 Datentypen für Strukturen

5.4.1 Deklarieren von Strukturen

Strukturen enthalten mehrere Elemente gleicher oder unterschiedlicher Datentypen.

```
TYPE
    maschine:
        STRUCT
            x_pos :      REAL;
            y_pos :      REAL;
            tiefe :      INT;
            u_min :      INT;
        END_STRUCT;
END_TYPE
```

Abbildung 50: Deklaration einer Datenstruktur

Im Beispiel besteht die Struktur 'maschine' aus vier Komponenten. Alle Komponenten beschreiben die Eigenschaften der Maschine.

5.4.2 Programmierbeispiel

Strukturen sollten verwendet werden, wenn Daten deklariert werden müssen, die gleiche Objekte beschreiben. Beispielsweise bohrt eine Bohrmaschine mehrere Löcher in ein Werkstück. Alle Löcher haben eine Position auf der x- und der y-Achse des Werkstücks, eine Bohrtiefe und werden mit einer bestimmten Umdrehungszahl gebohrt. Die konkreten Werte für die Löcher sind unterschiedlich, die benötigten Variablen sind jedoch immer die gleichen. In diesem Fall ist es sinnvoll, eine Struktur zu deklarieren, die aus vier Komponenten besteht, für die x- und y-Position, die Bohrtiefe sowie die Umdrehungen pro Minute. Für jedes Loch können den Komponenten verschiedene Werte zugewiesen werden. Der Funktionsbaustein für den Bohrvorgang arbeitet auf der gleichen Variable, die eine Struktur ist.

5.4.3 Felder von Strukturen

Eine Struktur kann auch in Feldern verwendet werden, wie im folgenden Beispiel dargestellt:

```
TYPE
  maschine:
  STRUCT
      x_pos  :      REAL;
      y_pos  :      REAL;
      tiefe  :      INT;
      u_min  :      INT;
  END_STRUCT;
  mein_feld :      ARRAY[1..10] OF maschine;
END_TYPE
```

Abbildung 51: Deklaration eines Felds einer Struktur

Ein Anwendungsbeispiel für Felder von Strukturen könnte eine Fertigungsstraße mit mehreren Bohrmaschinen sein. Über den Feldindex kann auf die entsprechende Bohrmaschine zugegriffen werden. Über die Strukturkomponenten können die verschiedenen Werte zum Bohren zugewiesen werden.

5.4.4 Strukturen mit Feldern

Felder können in Strukturen verwendet werden, wie im folgenden Beispiel dargestellt:

```
TYPE
  graph      :      ARRAY[1..10] of INT;
  antrieb:
  STRUCT
      u_min      :      INT;
      eingaenge :      IN_BOOL;
      leistung   :      graph;
  END_STRUCT;
END_TYPE
```

Abbildung 52: Deklaration einer Struktur eines Felds

5.4.5 Initialisieren von Strukturen

Strukturen können initialisiert werden, indem den Komponenten beim Editieren der Code-Deklaration die Werte zugewiesen werden. Ein Beispiel ist in der folgenden Abbildung dargestellt:

```
Variablen-Deklaration:
VAR
  var1      :      maschine;
  start     :      BOOL   :=TRUE;
END_VAR

Code-Deklaration in ST:
IF start THEN
  var1.x_pos := REAL#1.3E+2;
  var1.u_min := 3000;
  ...
  start := FALSE;
END_IF;
...
```

Abbildung 53: Zuweisen von Werten zu Komponenten einer Struktur

5.5 Datentypen für Zeichenfolgen

5.5.1 Deklarieren von Zeichenfolgen

Anwenderdefinierte Datentypen für Zeichenfolgen sind Zeichenfolgen mit einer variablen Anzahl von Zeichen. Beim Deklarieren einer anwenderdefinierten Zeichenfolge wird die Länge hinter dem Datentyp in runden Klammern angegeben.

5.5.2 Programmierbeispiel

Im folgenden Beispiel beträgt die Länge der Zeichenfolge 10 Zeichen.

```
TYPE
    STRING10      :      STRING (10) ;
END_TYPE
```

Abbildung 54: Deklaration eines Datentyps für Zeichenfolgen

5.6 Aufrufen des Texteditors mit einem Datentyp-Arbeitsblatt

Bevor Sie anwenderdefinierte Datentypen editieren können, müssen Sie zunächst den Texteditor mit dem Datentyp-Arbeitsblatt aufrufen. Hierzu verwenden Sie den Projektbaum. In der folgenden Beschreibung nehmen wir an, dass Sie das Datentyp-Arbeitsblatt 'Type' definieren wollen. Hierzu müssen Sie zunächst das Datentyp-Arbeitsblatt 'Type' in den Projektbaum einfügen, wie im Kapitel 'Arbeiten mit Projekten' beschrieben.



Aufrufen des Texteditors mit der Maus

- Doppelklicken Sie auf das Symbol 'Datentyp-Arbeitsblatt'.
Es wird der Texteditor mit dem Datentyp-Arbeitsblatt aufgerufen.



Aufrufen des Texteditors mit der Tastatur

- Drücken Sie <↓> oder <↑>, um das Symbol 'Datentyp-Arbeitsblatt' zu markieren.
- Drücken Sie <↵>.
Es wird der Texteditor mit dem Datentyp-Arbeitsblatt aufgerufen.



5.7 Editieren von Datentyp-Deklarationen mit dem Editor-Assistenten

Felder und Strukturen können mit dem Editor-Assistenten deklariert werden.

- Sie rufen den Editor-Assistenten für das Datentyp-Arbeitsblatt auf, indem Sie <UMSCHALT> + <F2> drücken. Das Listenfeld 'Gruppe' enthält nur den Eintrag 'Datentypen'.

Der Auswahlbereich enthält die Schlüsselwörter ARRAY, STRING und STRUCT.

Sie können diese Schlüsselwörter in Ihrem Datentyp-Arbeitsblatt einfügen. Mit Hilfe des Editor-Assistenten werden vorbereitete Konstrukte im Arbeitsblatt eingefügt, in denen Sie nur noch die grün dargestellten Kommentare durch die tatsächlichen Werte und Variablen ersetzen müssen.



Abbildung 55: Editor-Assistent in einem Datentyp-Arbeitsblatt

Die folgende Abbildung zeigt ein Feld und eine Struktur, die jeweils mit dem Editor-Assistenten eingefügt wurden. Für jedes Schlüsselwort wird dabei ein neuer Deklarationsblock verwendet.

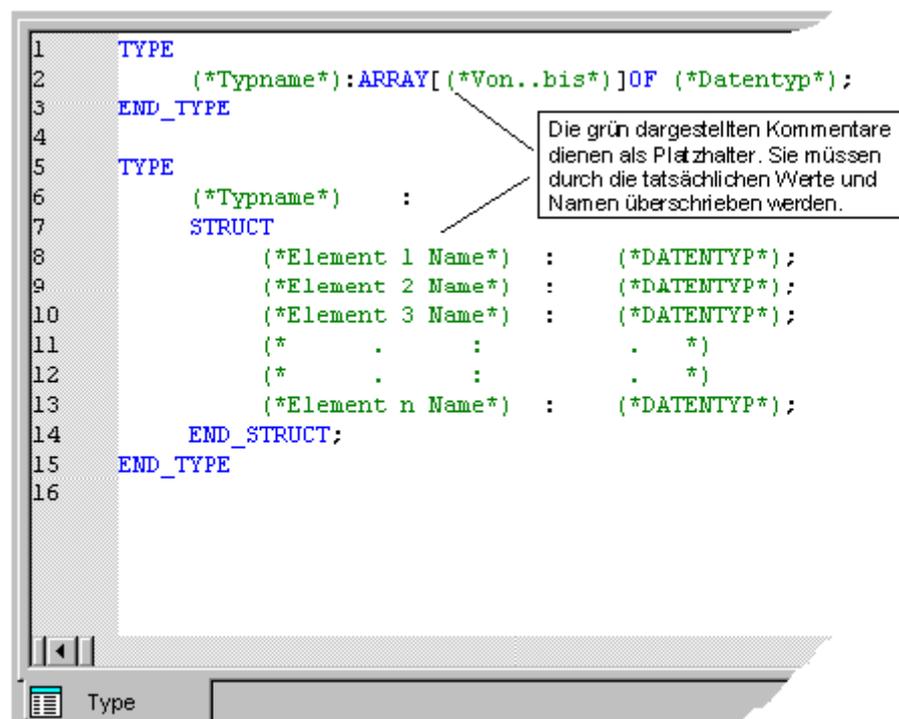


Abbildung 56: Datentyp-Arbeitsblatt 'Type' mit vorbereiteten Schlüsselwörtern, eingefügt mit dem Editor-Assistenten



HINWEIS

Allgemeine Informationen zum Editor-Assistenten finden Sie im Kapitel 'Allgemeines zum Programm'.



Einfügen einer Struktur über den Editor-Assistenten mit der Maus

- Legen Sie fest, an welcher Stelle im Datentyp-Arbeitsblatt die neue Struktur eingefügt werden soll. Beachten Sie, dass jedes Feld oder jede Struktur, die mit dem Editor-Assistenten eingefügt wird, in einem neuen Deklarationsblock eingeschlossen ist. Klicken Sie mit der linken Maustaste, um den Textcursor an der gewünschten Stelle zu positionieren.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Doppelklicken Sie im Editor-Assistenten in der Liste der Schlüsselwörter auf das Schlüsselwort 'STRUCT'. Das Schlüsselwort wird automatisch im Arbeitsblatt an der Cursorposition eingefügt. Kommentare dienen dabei als Platzhalter für die aktuellen Variablen und Werte. Sie erscheinen grün und sind durch runde Klammern und Sternchen gekennzeichnet, wie in [▶Abbildung 56◀](#) auf Seite 92 dargestellt.
- Überschreiben Sie die Kommentare mit den aktuellen Namen und Werten, die Sie in Ihrem Projekt verwenden.



Einfügen einer Struktur über den Editor-Assistenten mit der Tastatur

- Drücken Sie <↓> oder <↑>, um den Textcursor an der Stelle im Datentyp-Arbeitsblatt zu positionieren, an der die neue Struktur eingefügt werden soll. Beachten Sie, dass jedes Feld oder jede Struktur, die mit dem Editor-Assistenten eingefügt wird, in einem neuen Deklarationsblock eingeschlossen ist.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Drücken Sie <ALT> + <3>, um den Editor-Assistenten zu aktivieren.
- Drücken Sie <↓> oder <↑>, um das Schlüsselwort 'STRUCT' zu markieren.
- Drücken Sie <↵>, um die Struktur einzufügen. Das Schlüsselwort wird automatisch im Arbeitsblatt an der Cursorposition eingefügt. Kommentare dienen dabei als Platzhalter für die aktuellen Variablen und Werte. Sie erscheinen grün und sind durch runde Klammern und Sternchen gekennzeichnet, wie in [▶Abbildung 56◀](#) auf Seite 92 dargestellt.
- Überschreiben Sie die Kommentare mit den aktuellen Namen und Werten, die Sie in Ihrem Projekt verwenden.

5.8 Symbolische, adressierte und direkt dargestellte Variablen

Gemäß IEC 61131-3 werden anstelle der direkten Adressierung von Eingängen, Ausgängen oder Merkern Variablen zum Programmieren verwendet. Es können symbolische, adressierte oder direkt dargestellte Variablen deklariert werden.

Die Deklaration einer symbolischen Variablen setzt sich aus einem Variablennamen und einem Datentyp zusammen. Die Deklaration einer adressierten Variablen besteht aus einem Variablennamen, dem Speicherort der Variablen und einem Datentyp. Die Deklara-

tion einer direkt dargestellten Variablen besteht aus dem Speicherort der Variablen und einem Datentyp.



HINWEIS

Direkt dargestellte Variablen können nur in der Deklaration von globalen Variablen oder in Programmen deklariert werden.

In der folgenden Abbildung ist ein Beispiel jedes Variablentyps dargestellt:

```
VAR
Name                               :      Datentyp;
Name      AT      %Speicherort :      Datentyp;
              AT      %Speicherort :      Datentyp;
END_VAR
```

Abbildung 57: Deklaration einer symbolischen, einer adressierten und einer direkt dargestellten Variablen

Der Speicherort der Variablen setzt sich aus dem Präfix für den Speicherort und dem Präfix für die Größe zusammen. Präfixe für den Speicherort sind I für Eingänge, Q für Ausgänge und M für interne Speicher. Präfixe für die Größe sind X für Einzelbits, B für Byte-, W für Wort- und D für Doppelwortgröße.



HINWEIS

Adressierte und direkt dargestellte Variablen werden an der deklarierten logischen Adresse gespeichert. Der Anwendungsprogrammierer muss sicherstellen, dass keine Speicheradresse zweifach vergeben wird.

5.9 Globale und lokale Variablen

Der Gültigkeitsbereich jeder Variablen, der durch das Schlüsselwort der Variablen festgelegt wird, definiert die Verwendung der Variablen innerhalb einer POE oder innerhalb des gesamten Projekts. Daher werden zwei Typen unterschieden:

- * Lokale Variablen
- * Globale Variablen

Kann eine Variable nur innerhalb einer POE verwendet werden, wird sie als lokale Variable bezeichnet. In diesem Fall müssen die Variablen-Schlüsselwörter VAR, VAR_INPUT und VAR_OUTPUT verwendet werden.

Variablen, die innerhalb des gesamten Projekts verwendet werden können, werden als globale Variable bezeichnet. Sie muss in der globalen Deklaration als VAR_GLOBAL und in jeder POE, in der sie verwendet wird, als VAR_EXTERNAL deklariert werden.



HINWEIS

Es kann sich als sinnvoll erweisen, alle Ein- und Ausgänge als globale Variablen zu deklarieren. In der globalen Variablen-Deklaration sollten sie dann als adressierte Variablen und in der VAR_EXTERNAL-Deklaration der POE als symbolische Variablen deklariert werden. Auf diese Weise wird der Schreibaufwand verringert, wenn Speicherorte geändert werden sollen.

5.10 Remanente Variablen

Remanente Variablen sind Variablen, deren Werte auch gespeichert bleiben, wenn die Stromversorgung abgeschaltet wird. Bei einem Warmstart wird der letzte Wert der Variablen verwendet.

Zum Deklarieren remanenter Variablen wird das Schlüsselwort RETAIN verwendet, wie im folgenden Beispiel dargestellt:

```
VAR RETAIN
    var1      :      BOOL      := TRUE;
END_VAR
```

Abbildung 58: Beispiel für die Deklaration einer remanenten Variablen

In diesem Beispiel hat die Variable den Anfangswert 'TRUE', der den Anfangswert für einen Kaltstart angibt. Bei einem Warmstart wird der aktuelle Wert der Variablen verwendet.

Das Schlüsselwort RETAIN kann in Kombination mit den Schlüsselwörtern VAR, VAR_OUTPUT und VAR_GLOBAL verwendet werden. Remanente Variablen können nicht mit den Schlüsselwörtern VAR_INPUT und VAR_EXTERNAL deklariert werden.

5.11 Initialisieren von Variablen

Gemäß IEC 61131-3 können Variablen Anfangswerte zugewiesen werden. Anfangswerte können allen Variablen zugewiesen werden, außer in VAR_EXTERNAL-Deklarationen. Wenn die SPS gestartet wird, wird die Variable mit diesem Anfangswert verarbeitet.

Anfangswerte müssen am Ende der Deklarationszeile der Variablen mit ':=' eingefügt werden, wie in der folgenden Abbildung dargestellt:

5.12 Schlüsselwörter zur Variablen-Deklaration

```
VAR
Name                               :   Datentyp   := Anfangswert;
Name AT %Speicherort               :   Datentyp   := Anfangswert;
      AT %Speicherort               :   Datentyp   := Anfangswert;
END_VAR
```

Abbildung 59: Deklaration und Initialisierung einer symbolischen, einer adressierten und einer direkt dargestellten Variablen

Der Anfangswert muss zu dem Datentyp passen. Es ist beispielsweise nicht möglich, einer Variable vom Datentyp BOOL den Anfangswert '5' zuzuweisen. In diesem Fall zeigt das System eine Fehlermeldung an.

Der Anfangswert kann nach Belieben vergeben werden. Wenn kein Anfangswert verwendet wird, wird die Variable mit dem Standard-Anfangswert des Datentyps oder bei remanenten Variablen mit dem gepufferten Wert initialisiert.

5.12 Schlüsselwörter zur Variablen-Deklaration

In der IEC 61131-3 werden verschiedene Typen von Variablen-Deklarationen beschrieben. Wie Sie der folgenden Tabelle entnehmen können, wird für jeden Typ ein anderes Schlüsselwort verwendet:

Schlüsselwort	Variablentyp / Verwendung
VAR	für interne Variablen, die nur innerhalb einer POE verwendet werden können. zum Deklarieren der Instanzen von Funktionsbausteinen. kann zur Deklaration von direkt dargestellten und adressierten Variablen in Programmen verwendet werden. kann zur Deklaration von symbolischen Variablen verwendet werden. kann zusammen mit dem Schlüsselwort 'RETAIN' zur Deklaration remanenter Variablen verwendet werden.
VAR_INPUT	für Variablen, die Eingänge von Funktionen, Funktionsbausteinen und Programmen sind. um einen Wert an eine POE zu geben, der beispielsweise von einer anderen POE kommt. der Wert wird nur innerhalb der POE gelesen. kann nur zur Deklaration von symbolischen Variablen verwendet werden.

Schlüsselwort	Variablentyp / Verwendung
VAR_OUTPUT	für Variablen, die Ausgänge von Funktionen, Funktionsbausteinen und Programmen sind. liefert einen Ausgangswert, z. B. für andere POEs. der Wert wird innerhalb der POE geschrieben. der Wert kann auch gelesen werden. kann nur zur Deklaration von symbolischen Variablen verwendet werden. kann zusammen mit dem Schlüsselwort 'RETAIN' zur Deklaration remanenter Variablen verwendet werden.
VAR_IN_OUT	Adresse der Variablen wird übergeben (Referenzparameter). die Variable kann gelesen oder geschrieben werden. wird typischerweise für komplexe Datentypen wie Folgen, Felder und Strukturen verwendet.
VAR_EXTERNAL	für globale Variablen in der POE. der Wert wird durch die Deklaration von VAR_GLOBAL geliefert. kann nicht initialisiert werden. der Wert kann innerhalb der POE geändert werden. kann nur zur Deklaration von symbolischen Variablen verwendet werden.
VAR_EXTERNAL_PG	für globale Variablen im Programm. der Wert wird durch die Deklaration von VAR_GLOBAL_PG geliefert. kann nicht initialisiert werden. der Wert kann innerhalb des Programms geändert werden. kann nur zur Deklaration von symbolischen Variablen verwendet werden.
VAR_EXTERNAL_FB	für globale Variablen im Funktionsblock. der Wert wird durch die Deklaration von VAR_GLOBAL_FB geliefert. kann nicht initialisiert werden. der Wert kann im Funktionsbaustein geändert werden. kann nur zur Deklaration von symbolischen Variablen verwendet werden.
VAR_GLOBAL	für globale Variablen, die in allen Programmen und Funktionsbausteinen des Projekts verwendet werden können. kann zur Deklaration von direkt dargestellten, adressierten und symbolischen Variablen verwendet werden. kann zusammen mit dem Schlüsselwort 'RETAIN' zur Deklaration remanenter Variablen verwendet werden.
VAR_GLOBAL_PG	für globale Variablen, die in allen Programmen des Projektes verwendet werden können. kann zur Deklaration von direkt dargestellten, adressierten und symbolischen Variablen verwendet werden. kann zusammen mit dem Schlüsselwort 'RETAIN' zur Deklaration remanenter Variablen verwendet werden.

5.13 Deklarieren von Variablen

Schlüsselwort	Variablentyp / Verwendung
VAR_GLOBAL_FB	für globale Variablen, die in allen Funktionsbausteinen des Projekts verwendet werden können. kann zur Deklaration von direkt dargestellten, adressierten und symbolischen Variablen verwendet werden. kann zusammen mit dem Schlüsselwort 'RETAIN' zur Deklaration permanenter Variablen verwendet werden.
END_VAR	um einen Variablen-Deklarationsblock zu beenden.

Abbildung 60: Schlüsselwörter für Variablen-Deklarationen



HINWEIS

Globale Variablen müssen in der globalen Variablen-Deklaration der Ressource als VAR_GLOBAL und in der Variablen-Deklaration der POE als VAR_EXTERNAL deklariert werden.



HINWEIS

Die Schlüsselwörter VAR_GLOBAL_PG, VAR_GLOBAL_FB, VAR_EXTERNAL_PG und VAR_EXTERNAL_FB sind IEC-Erweiterungen.

5.13 Deklarieren von Variablen

Sie haben zwei Möglichkeiten Variablen zu deklarieren:

- * Deklarieren von Variablen beim Editieren eines Code.
- * Deklarieren von Variablen im Variablen-Arbeitsblatt mit dem Texteditor.

Beim ersten Verfahren werden Variablen in ein Code-Arbeitsblatt eingefügt, die zuvor nicht deklariert wurden. In diesem Fall wird die Variable automatisch über die Dialoge 'Variable' und 'Automatische Variablen-Deklaration' deklariert. Dabei wird die Variablen-Deklaration automatisch in das Variablen-Arbeitsblatt und die neue Variable in das Code-Arbeitsblatt eingefügt. Dieses Verfahren wird in den entsprechenden Kapiteln zu den Programmiersprachen und der AS beschrieben.



HINWEIS

Das entsprechende Variablen-Arbeitsblatt, in das die neue Variablen-Deklaration automatisch eingefügt werden soll, wird geschlossen, wenn der Dialog 'Variable' aufgerufen wird.

Beim zweiten Verfahren werden die Variablen einfach über die Tastatur im Variablen-Arbeitsblatt deklariert. Mit dem Editor-Assistenten können dabei die Schlüsselwörter für die Variablen-Deklaration (z. B. VAR, VAR_INPUT, usw.) in das Arbeitsblatt eingefügt werden.

- Sie rufen den Editor-Assistenten für das Variablen-Arbeitsblatt auf, indem Sie <UMSCHALT> + <F2> drücken. Das Listenfeld 'Gruppe' enthält nur den Eintrag 'Variablen-Deklaration'.

Die verfügbaren Schlüsselwörter sind im Auswahlbereich aufgeführt.

Sie können diese Schlüsselwörter in Ihrem Variablen-Arbeitsblatt einfügen. Mit Hilfe des Editor-Assistenten werden vorbereitete Konstrukte im Arbeitsblatt eingefügt, in denen Sie nur noch die grün dargestellten Kommentare durch die tatsächlichen Variablen-Deklarationen ersetzen müssen.



Abbildung 61: Editor-Assistent in einem Variablen-Arbeitsblatt

Die folgende Abbildung zeigt eine VAR-Deklaration, die mit dem Editor-Assistenten eingefügt wurde. Beachten Sie, dass für jedes Schlüsselwort ein neuer Deklarationsblock eingefügt wird.

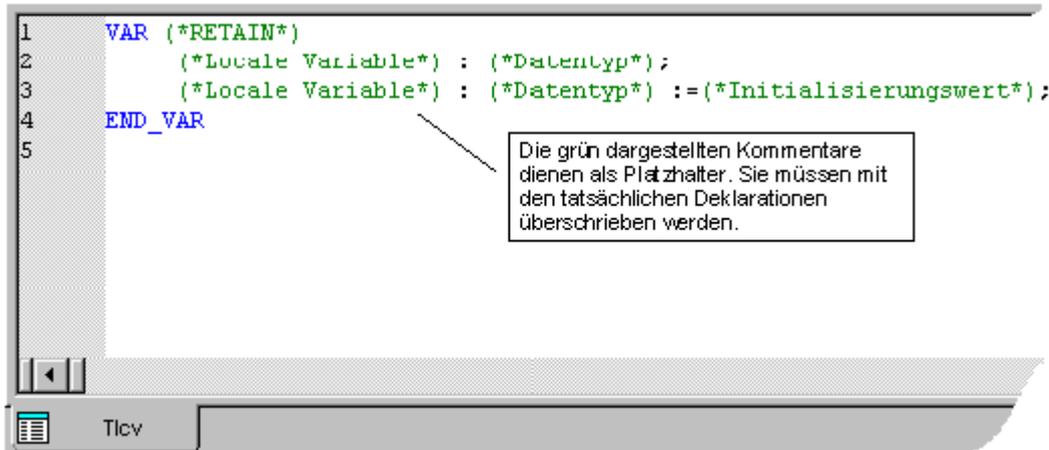


Abbildung 62: Variablen-Arbeitsblatt mit vorbereitetem Schlüsselwort 'VAR', eingefügt mit dem Editor-Assistenten

HINWEIS



Allgemeine Informationen zum Editor-Assistenten finden Sie im Kapitel 'Allgemeines zum Programm'.



Einfügen einer VAR-Deklaration über den Editor-Assistenten mit der Maus

- Legen Sie fest, an welcher Stelle im Variablen-Arbeitsblatt die VAR-Deklaration eingefügt werden soll. Beachten Sie, dass die Deklaration nicht innerhalb eines anderen Deklarationsblocks eingefügt werden darf.
Klicken Sie mit der linken Maustaste, um den Textcursor an der gewünschten Stelle zu positionieren.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Doppelklicken Sie im Editor-Assistenten in der Liste der Schlüsselwörter auf das Schlüsselwort 'VAR'. Das Schlüsselwort wird automatisch im Arbeitsblatt an der Cursorposition eingefügt. Kommentare dienen dabei als Platzhalter für die aktuellen Variablen-Deklarationen. Sie erscheinen grün und sind durch runde Klammern und Sternchen gekennzeichnet, wie in ►Abbildung 62◄ auf Seite 99 dargestellt.
- Überschreiben Sie die Kommentare durch die aktuellen Deklarationen.



Einfügen einer VAR-Deklaration über den Editor-Assistenten mit der Tastatur

- Drücken Sie <↓> oder <↑>, um den Textcursor an der Stelle im Variablen-Arbeitsblatt zu positionieren, an der die VAR-Deklaration eingefügt werden soll. Beachten Sie, dass die Deklaration nicht innerhalb eines anderen Deklarationsblocks eingefügt werden darf.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Drücken Sie <ALT> + <3>, um den Editor-Assistenten zu aktivieren.
- Drücken Sie <↓> oder <↑>, um das Schlüsselwort 'VAR' zu markieren.
- Drücken Sie <↵>, um das Schlüsselwort einzufügen. Das Schlüsselwort wird automatisch im Arbeitsblatt an der Cursorposition eingefügt. Kommentare dienen dabei als Platzhalter für die aktuellen Variablen-Deklarationen. Sie erscheinen grün und sind durch runde Klammern und Sternchen gekennzeichnet, wie in ►Abbildung 62◄ auf Seite 99 dargestellt.
- Überschreiben Sie die Kommentare durch die aktuellen Deklarationen.

5.14 Instanzieren

Die IEC 61131-3 bietet die Möglichkeit des Instanzierens. Instanzieren bedeutet, dass ein Funktionsbaustein einmal definiert wird und dann mehrfach verwendet werden kann. Da Funktionsbausteine immer einen internen Speicher haben, müssen die Werte für jeden Aufruf des Funktionsbausteins in einem anderen Speicherbereich gespeichert werden. Hierzu werden Instanznamen verwendet. Der Instanzname wird in der Variablen-Deklaration der POE deklariert, in der der Funktionsbaustein verwendet wird. In der folgenden Abbildung ist ein Beispiel einer Variablen-Deklaration für den Funktionsbaustein 'FB_exam' mit zwei Instanzen dargestellt:

```
VAR
    drive1      :      FB_exam;
    drive2      :      FB_exam;
END_VAR
```

Abbildung 63: Instanziierung eines Funktionsbausteins

Der Funktionsbaustein 'FB_exam', dessen Code an irgendeiner Stelle im Programm definiert worden ist, besitzt zwei Instanzen. Der Instanzname der ersten Instanz ist 'drive1', der der zweiten Instanz 'drive2'. In dem entsprechenden Code-Arbeitsblatt können Sie den Funktionsbaustein 'FB_exam' zweimal verwenden. Sie müssen dann jeweils den richtigen Instanznamen eingeben.

HINWEIS



Instanznamen werden immer automatisch erzeugt, wenn ein Funktionsbaustein mit dem Editor-Assistenten eingefügt wird.

Funktionsbausteine können innerhalb von Programmen oder von Funktionsbausteinen instanziiert werden. Funktionen können verwendet werden, ohne dass sie instanziiert werden müssen, da sie über keinen internen Speicher verfügen.

Der Instanzbaum zeigt alle in Ihrem Projekt verwendeten Instanzen an, wie im folgenden Beispiel dargestellt:

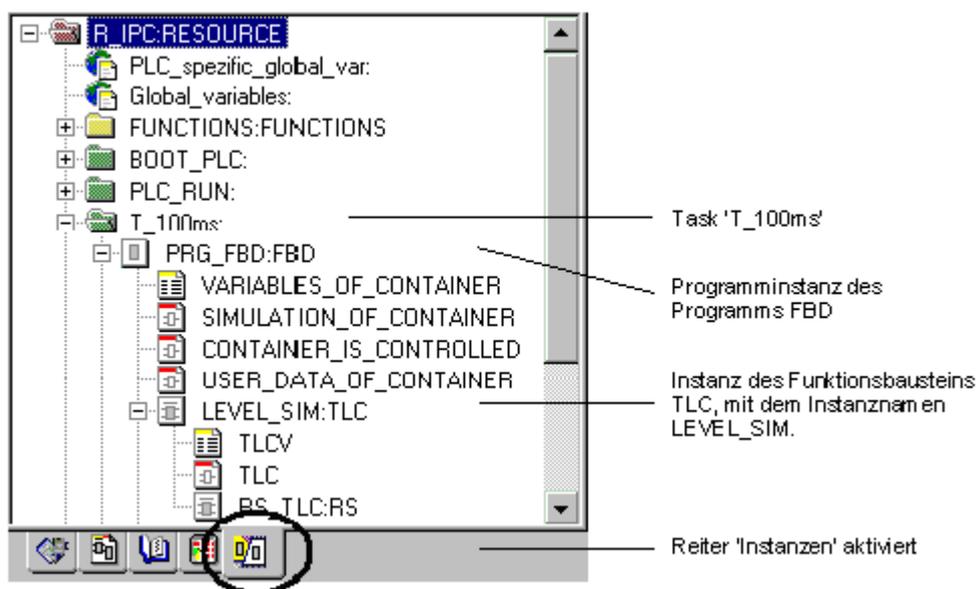


Abbildung 64: Beispiel für einen Instanzbaum

In diesem Beispiel gibt es zum Programm 'FBD' eine Programminstanz, die in der Task 'T_100ms' verwendet wird. Der Funktionsbaustein 'TLC' wurde im Programm 'FBD' instanziiert.

Sie zeigen den Instanzbaum an, indem Sie auf den Reiter 'Instanzen' unterhalb des Projektbaums klicken, wie in [▶Abbildung 64◀](#) auf Seite 101 dargestellt.

HINWEIS



Programminstanzen werden erzeugt, indem ein Programm einer Task zugewiesen und im entsprechenden Dialog ein Instanzname eingegeben wird.



Das Zuweisen von Programmen zu Tasks wird im Kapitel 'Kompilieren, Senden und Debuggen' dieses Handbuchs beschrieben.

6

EDITIEREN IN ST

6.1 Aufrufen des Texteditors mit einem ST-Arbeitsblatt

Zum Editieren eines Code-Arbeitsblattes in ST müssen Sie zunächst aus dem Projektbaum den Texteditor mit dem ST-Arbeitsblatt aufrufen.



Eine allgemeine Beschreibung des Projektbaums sowie Informationen, wie Sie im Projektbaum durch die POEs blättern, finden Sie in den Kapiteln 'Allgemeines zum Programm' und 'Arbeiten mit Projekten' in diesem Handbuch.

Im folgenden Beispiel nehmen wir an, dass Sie das ST-Arbeitsblatt eines Funktionsbausteins mit dem Namen ST_FB editieren wollen. Hierzu müssen Sie zunächst einen Funktionsbaustein mit diesem Namen einfügen, wie im Kapitel 'Arbeiten mit Projekten' beschrieben. Beachten Sie dabei, dass die Programmiersprache eines Arbeitsblattes durch die POE-Sprache festgelegt wird. Diese wird beim Einfügen der POE beziehungsweise des Arbeitsblattes definiert.



Aufrufen des Texteditors für das ST-Arbeitsblatt mit der Maus

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den Unterbaum 'ST_FB' mit den POE-Arbeitsblättern durch Doppelklicken auf den POE-Namen.
- Doppelklicken Sie auf das Symbol 'Arbeitsblatt in ST' des Funktionsbausteins 'ST_FB'.

Es wird der Texteditor mit dem ST-Arbeitsblatt aufgerufen.





Aufrufen des Texteditors für das ST-Arbeitsblatt mit der Tastatur

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den entsprechenden Unterbaum mit den POE-Arbeitsblättern wie folgt: Drücken Sie <↓> oder <↑>, um den Funktionsbaustein 'ST_FB' zu markieren. Drücken Sie <→>, um den Unterbaum des Funktionsbausteins zu öffnen.
- Drücken Sie <↓> oder <↑>, um das Symbol 'Arbeitsblatt in ST' des Funktionsbausteins 'ST_FB' zu markieren.
- Drücken Sie <↵>.
- Es wird der Texteditor mit dem ST-Arbeitsblatt aufgerufen.

6.2 Einführung in ST

Ein Code in der Textsprache ST setzt sich aus Anweisungen und Ausdrücken zusammen.

Beim Editieren können verschiedene Anweisungstypen verwendet werden.

Ein Ausdruck ist Teil einer Anweisung und liefert den notwendigen Wert zur Ausführung der Anweisung. Ausdrücke bestehen aus Operatoren und Operanden. Die Operatoren müssen auf die Operanden so angewendet werden, dass dem Operator mit der höchsten Rangfolge der Operator mit der nächstniederen Rangfolge folgt.

Anweisungen und Ausdrücke können entweder direkt in das Arbeitsblatt eingegeben werden oder mit dem Editor-Assistenten eingefügt werden. Der Editor-Assistent erleichtert das Arbeiten mit dem Texteditor. In ST können mit dem Editor-Assistenten Standard-Schlüsselwörter, Funktionen und Funktionsbausteine in den entsprechenden Arbeitsblättern eingefügt werden. Wenn Sie Anweisungen direkt in das Arbeitsblatt eingeben, sollten Sie jede Anweisung in einer neuen Zeile beginnen und Anweisungen sowie Schleifen einrücken, um einen besseren Überblick zu haben. Am Ende jeder Anweisung muss ein Semikolon stehen. Beim Arbeiten mit dem Editor-Assistenten erfolgt dies automatisch. Kommentare können unter Verwendung von Sternchen und runden Klammern eingefügt werden. Jede Zeile beginnt mit einer Zeilennummer. Die Syntaxhervorhebung gibt die einzelnen Elemente in unterschiedlichen Farben aus, d. h. Schlüsselwörter erscheinen blau, Variablen und Instanznamen schwarz und Kommentare grün.



Eine Beschreibung zum Editor-Assistenten finden Sie im Abschnitt 'Editieren von Anweisungen mit dem Editor-Assistenten' in diesem Kapitel.

6.3 Einfügen und Editieren von Zuweisungen

Eine Zuweisung ist ein spezieller Anweisungstyp. Sie bildet den Wert eines Ausdrucks von der rechten Seite auf die Variable auf der linken Seite ab, wie in der folgenden Abbildung dargestellt:

```

Variablenname      :=      Ausdruck;
```

Abbildung 65: Anweisung vom Typ 'Zuweisung' in ST

Zuweisungen können entweder mit dem Editor-Assistenten eingefügt oder direkt eingegeben werden. In Zuweisungen ist zu beachten, dass die Variable auf der linken und der Wert des Ausdrucks auf der rechten Seite vom gleichen Datentyp sind. Ist dies nicht der Fall, ist eine Typumwandlung erforderlich.

Der Ausdruck auf der rechten Seite setzt sich aus Operanden und Operatoren zusammen. Operanden können Literale, Variablen, Funktionsaufrufe oder andere Ausdrücke sein. Die IEC 61131-3 gibt eine Liste möglicher Operatoren vor, mit denen die Operanden verarbeitet werden können. Diese Operatoren haben bestimmte Prioritäten. Der Operator mit der höchsten Priorität wird als erster ausgewertet. Die möglichen Operatoren sind in der folgenden Tabelle aufgelistet. Der Operator mit der höchsten Priorität steht in der ersten Reihe der Tabelle, der mit der niedrigsten Priorität in der letzten Reihe.

Operator	Beispiel	Wert des Beispiels	Bedeutung
()	(2+3) * (4+5)	45	Klammerung
**	3**4	81	Potenzierung
-	-10	-10	Negation
NOT	NOT TRUE	FALSE	Komplement
*	10*3	30	Multiplikation
/	6/2	3	Division
MOD	17 MOD 10	7	Modulo
+	2+3	5	Addition
-	4-2	2	Subtraktion
<, >, <=, >=	4 > 12	FALSE	Vergleich
=	T#26h = T#1d2h	TRUE	Gleichheit
<>	8 <>16	TRUE	Ungleichheit
&, AND	TRUE & FALSE	FALSE	Boolesches UND
XOR	TRUE XOR FALSE	TRUE	Boolesches Exklusiv-ODER
OR	TRUE OR FALSE	TRUE	Boolesches ODER

Abbildung 66: Operatoren in ST

6.4 Einfügen und Editieren weiterer Anweisungen

In ST können außer dem Anweisungstyp 'Zuweisung' noch weitere Anweisungen verwendet werden. Dazu gehören z. B. Auswahlanweisungen, Wiederholungsanweisungen oder Rücksprunganweisungen. Die Schlüsselwörter der verschiedenen Anweisungen sowie Beispiele für ihre Verwendung und ihre Bedeutung sind in der folgenden Tabelle dargestellt:



HINWEIS

Sie können mit dem Editor-Assistenten vorbereitete Anweisungen in ST einfügen. Um mit dem Editor-Assistenten eine Anweisung einzufügen, führen Sie die Schritte im folgenden Abschnitt 'Einfügen von Anweisungen mit dem Editor-Assistenten' durch.

Schlüsselwort	Beispiel	Bedeutung
IF	<pre>IF a < b THEN c:=1; ELSIF a=b THEN c:=2; ELSE c:=3; END_IF;</pre>	<p>Auswahlanweisung: Eine Gruppe von Anweisungen wird nur ausgeführt, wenn der zugeordnete Ausdruck 'a<b' TRUE ist. Wenn die Bedingung FALSE ist, wird entweder keine Anweisung ausgeführt oder die Anweisungen, die auf ELSE folgen.</p>
CASE	<pre>CASE f OF 1: a:=3; 2..5: a:=4; 6: a:=2; b:=1; ELSE a:=0; END_CASE;</pre>	<p>Auswahlanweisung: Eine Gruppe von Anweisungen wird entsprechend dem Wert des Ausdrucks hinter dem Schlüsselwort CASE ausgeführt. Die Variable oder der Ausdruck 'f' muss vom Datentyp INT sein.</p>
FOR	<pre>FOR a:=1 TO 10 BY 3 DO f[a] :=b; END_FOR;</pre>	<p>Wiederholungsanweisung: Eine Gruppe von Anweisungen wird wiederholt ausgeführt, wobei die Variable 'a' jeweils um '3' vergrößert wird. Der Vorgang beginnt bei '1' und endet bei '10'. Der Anfangswert wird mit der Steuervariablen 'a' eingestellt. Der Endwert folgt nach 'TO' und die Inkremente folgen nach 'BY'. Alle Werte müssen vom Datentyp ANY_INT sein.</p> <p>Hinweis: Wird 'BY' nicht verwendet, wird der Standardwert '1' verwendet. In diesem Fall müssen alle Werte vom Datentyp INT sein.</p>
WHILE	<pre>WHILE b > 1 DO b:= b/2 END_WHILE;</pre>	<p>Wiederholungsanweisung: Eine Gruppe von Anweisungen wird wiederholt ausgeführt, bis der zugeordnete Ausdruck 'b>1' FALSE ist. Die Bedingung für die Anweisung wird am Anfang der Schleife ausgeführt. Wenn die Bedingung FALSE ist, wird die Schleife nicht ausgeführt.</p>

Schlüsselwort	Beispiel	Bedeutung
REPEAT	<pre>REPEAT a := a*b; UNTIL a < 10000 END_REPEAT;</pre>	Wiederholungsanweisung: Eine Gruppe von Anweisungen wird wiederholt ausgeführt, bis der zugeordnete Ausdruck TRUE ist. Die Bedingung für die Anweisung wird am Ende der Schleife ausgeführt. Wenn die Bedingung FALSE ist, wird die Schleife mindestens einmal ausgeführt.
RETURN	<pre>RETURN;</pre>	Rücksprung-Anweisung: Die Rücksprung-Anweisung verlässt das aufgerufene Element (Funktion, Funktionsbaustein oder Programm) und springt zurück in die aufrufende POE.
EXIT	<pre>FOR a:=1 TO 2 DO IF flag THEN EXIT; END_IF; SUM := SUM + a; END_FOR</pre>	EXIT-Anweisung: Die EXIT-Anweisung kann verwendet werden, um die Ausführung einer Wiederholungsanweisung abzubrechen.

Abbildung 67: Anweisungen in ST

6.5 Einfügen von Anweisungen mit dem Editor-Assistenten

Wie bereits im vorigen Abschnitt erwähnt, können mit dem Editor-Assistenten sehr einfach Codes editiert werden, d. h. Anweisungen eingefügt werden und dabei Fehleingaben vermieden werden.



HINWEIS

Allgemeine Informationen zum Editor-Assistenten finden Sie im Abschnitt 'Der Editor-Assistent' im Kapitel 'Allgemeines zum Programm' in diesem Handbuch.

Der Editor-Assistent in ST-Arbeitsblättern enthält Schlüsselwörter, Funktionen und Funktionsbausteine, die sehr einfach in den Code eingefügt werden können. Der Editor-Assistent bietet die folgenden Vorteile:

- * Syntaktische Fehler wie z. B. fehlende Strichpunkte, Auswahl- oder Wiederholungsanweisungen ohne END-Schlüsselwort, usw. werden vermieden, indem vorbereitete Anweisungen, Funktionen und Funktionsbausteine eingefügt werden. Die für die jeweilige Struktur spezifischen Variablen und Werte erscheinen als Kommentare, die anschließend einfach überschrieben werden müssen.
- * Es ist nicht notwendig, dass der Anwender die genaue Syntax der verschiedenen Anweisungen, wie z. B. von Funktionen oder Funktionsbausteinen kennt.

Wenn der Editor-Assistent nicht geöffnet ist, führen Sie die folgenden Schritte durch:



Aufrufen des Editor-Assistenten mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Editor-Assistent'. Es erscheint der Editor-Assistent.



Aufrufen des Editor-Assistenten mit der Tastatur

- Drücken Sie <UMSCHALT> + <F2>. Es erscheint der Editor-Assistent.

In der folgenden Anweisung wird beschrieben, wie eine CASE-Anweisung mit dem Editor-Assistenten in das Code-Arbeitsblatt eingefügt wird. Dabei wird vorausgesetzt, dass der Editor-Assistent bereits geöffnet ist. Ist dies nicht der Fall, rufen Sie den Editor-Assistenten entsprechend der oben beschriebenen Vorgehensweise auf.



Einfügen einer CASE-Anweisung über den Editor-Assistenten mit der Maus

- Legen Sie fest, an welcher Stelle im Code-Arbeitsblatt die neue Anweisung eingefügt werden soll. Klicken Sie mit der linken Maustaste, um den Textcursor an der gewünschten Stelle zu positionieren.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Wählen Sie im Editor-Assistenten aus dem Listenfeld 'Gruppe' die Gruppe 'Schlüsselwörter'. Die verfügbaren Schlüsselwörter werden im Auswahlbereich angezeigt.
- Suchen Sie das Schlüsselwort 'CASE'.
- Doppelklicken Sie auf das Schlüsselwort 'CASE'. Das Schlüsselwort wird automatisch im Arbeitsblatt an der Cursorposition eingefügt, wie in der folgenden Abbildung gezeigt. Kommentare dienen dabei als Platzhalter für die aktuellen Variablen und Werte. Sie erscheinen grün und sind durch runde Klammern und Sternchen gekennzeichnet.

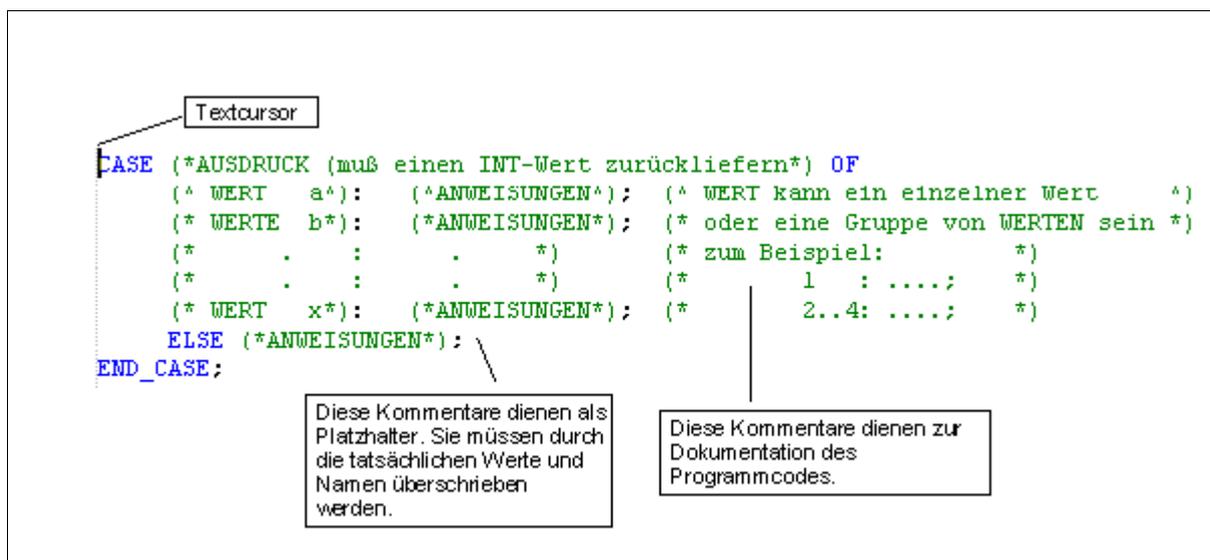


Abbildung 68: Vorbereitete CASE-Anweisung, mit dem Editor-Assistenten eingefügt

Im nächsten Schritt müssen die Kommentare, die als Platzhalter dienen, mit den benötigten Variablen und Werten überschrieben werden.



Editieren der neu eingefügten Anweisung



- Markieren Sie die grünen Platzhalter und überschreiben Sie diese durch die entsprechenden Anweisungen und Ausdrücke (Variablen und Werte).

6.6 Einfügen von Variablen

Beim Editieren in ST gibt es drei Möglichkeiten, eine Variable einzufügen:

- * Einfügen einer Variablen, die bereits im Variablen-Arbeitsblatt der POE deklariert wurde.
- * Einfügen einer Variablen, die zuvor nicht im Variablen-Arbeitsblatt der POE deklariert wurde und deren nachträgliches Deklarieren im Variablen-Arbeitsblatt der POE.
- * Einfügen einer Variablen und Deklarieren beim Editieren.

In den ersten beiden Fällen wird die Variable in das Code-Arbeitsblatt eingefügt, indem lediglich der Variablenname per Tastatur eingegeben wird.

Im folgenden Abschnitt wird der letzte Fall beschrieben. Nehmen wir einmal an, dass Sie die Variable 'T_value' in Ihr Code-Arbeitsblatt einfügen wollen.



Einfügen einer Variablen mit der Maus

- Geben Sie die Variable an der gewünschten Stelle im Code-Arbeitsblatt per Tastatur ein.
- Doppelklicken Sie auf den Variablenname, um diesen zu markieren.
- Klicken Sie in der Symbolleiste auf das Symbol 'Variable'. Es erscheint der Dialog 'Variable'. Der Variablenname wird im Feld 'Variablenliste für Ressource: POE-Name' angezeigt.

```
T_value := 100;
```

```
T_value := 100;
```



Einfügen einer Variablen mit der Tastatur

- Geben Sie die Variable an der gewünschten Stelle im Code-Arbeitsblatt per Tastatur ein.
- Halten Sie die <UMSCHALT>-Taste gedrückt und markieren Sie die Variable mit <←→>.
- Drücken Sie <F5>. Es erscheint der Dialog 'Variable'. Der Variablenname wird im Feld 'Variablenliste für Ressource: POE-Name' angezeigt.

```
T_value := 100;
```

```
T_value := 100;
```



HINWEIS

Sie können den Dialog 'Variable' auch öffnen, ohne zuvor eine zu deklarierende Variable markiert zu haben. In diesem Fall ist das Feld 'Variablenliste für Ressource: POE-Name' leer.



Abbildung 69: Dialog 'Variable'



Verwenden des Dialogs 'Variable'



- Geben Sie bei Bedarf einen Variablennamen ein (siehe obiger Hinweis).
- Bestätigen Sie den Dialog 'Variable'.
Es erscheint der Dialog 'Automatische Variablen-Deklaration'.

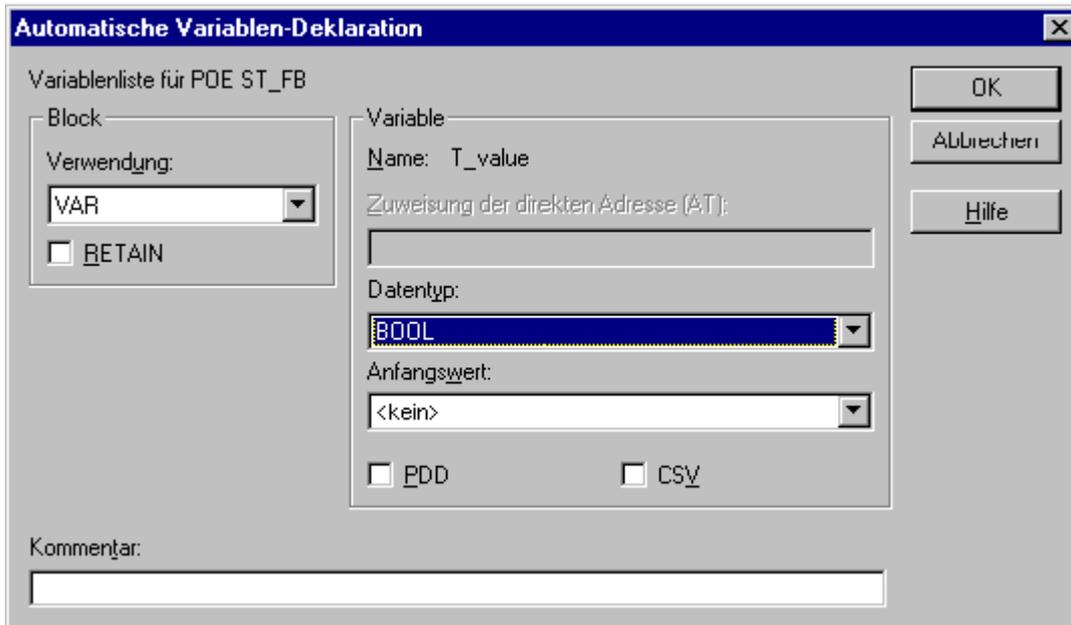


Abbildung 70: Dialog 'Automatische Variablen-Deklaration'



Verwenden des Dialogs 'Automatische Variablen-Deklaration'



- Wählen Sie aus dem Listenfeld 'Verwendung' ein Variablen-Schlüsselwort.
- Geben Sie einen Speicherort im Feld 'Zuweisung der direkten Adresse (AT)' ein, wenn Sie eine adressierte Variable deklarieren wollen (dies ist nur in Programmen oder für globale Variablen möglich).
- Wählen Sie den richtigen Datentyp aus dem Feld 'Datentyp'.
- Geben Sie bei Bedarf einen Anfangswert ein.
- Aktivieren Sie das Kontrollkästchen 'PDD', wenn die Variable im OmegaOS PDD (Process Data Directory) gespeichert werden soll.
- Aktivieren Sie das Kontrollkästchen 'CSV', wenn die Variable in der CSV-Datei gespeichert werden soll, d. h. im OPC-Server verwendet werden soll. Der OPC-Server verarbeitet nur Variablen, die in der CSV-Datei eingetragen sind. Diese Variablen können in einem OPC-Client-Prozess verwendet werden (z. B. in einer Visualisierung).
- Geben Sie bei Bedarf einen Kommentar ein.
- Bestätigen Sie den Dialog.

Die neue Variable wird in das Code-Arbeitsblatt eingefügt. Die Deklaration der Variablen wird automatisch in der Variablen-Deklaration der POE vorgenommen.



HINWEIS

Wenn Sie bereits zuvor deklarierte Variablen einfügen, erscheint der Name der Variablen im Listenfeld des Dialogs 'Variable'. Beim Bestätigen des Dialogs wird die Variable direkt in das Code-Arbeitsblatt eingefügt. Der Dialog 'Automatische Variablen-Deklaration' erscheint dann nicht.



Detaillierte Informationen zum OPC-Server finden Sie im Handbuch zum OPC-Server.

6.7 Aufruf von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten

Funktionen können in jedem Ausdruck verwendet werden, wie in der folgenden Abbildung für eine Zuweisung dargestellt:

```
Variablenname      :=      Funktionsname(invar1, invar2);
```

Abbildung 71: Funktionsaufruf in ST

In dem Beispiel wird der Wert der Funktion von der rechten Seite auf die Variable auf der linken Seite abgebildet. Anstelle der Variablen 'invar1' und 'invar2' können auch Konstanten verwendet werden. Auf diese Weise können sowohl Standardfunktionen als auch anwenderdefinierte Funktionen aufgerufen werden.

Funktionsbausteine können als einzelne Anweisung aufgerufen werden, indem der Instanzname verwendet wird, wie in der folgenden Abbildung dargestellt:

```
instanz(invar1:=1, invar2:=2);  
a:= instanz.outvar1;
```

Abbildung 72: Aufruf eines Funktionsbausteins in ST

In welcher Reihenfolge die Formalparameter übergeben werden ist unbedeutend. Wenn Parameter fehlen, wird der Anfangswert oder der Wert des letzten Aufrufs verwendet. Es gibt keinen Unterschied zwischen dem Aufruf eines Standardfunktionsbausteins und eines anwenderdefinierten Funktionsbausteins.



HINWEIS

Denken Sie daran, die Instanz für den Funktionsbaustein in der Variablen-Deklaration der POE zu deklarieren, wie im Abschnitt 'Instanzieren' im Kapitel 'Literele, Datentypen und Variablen' beschrieben.

Mit dem Editor-Assistenten können sehr einfach und bequem Funktionen oder Funktionsbausteine in das Code-Arbeitsblatt eingefügt werden und somit Fehleingaben vermieden werden.

Wenn der Editor-Assistent nicht geöffnet ist, führen Sie die folgenden Schritte durch:



Aufrufen des Editor-Assistenten mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Editor-Assistent'. Es erscheint der Editor-Assistent.



Aufrufen des Editor-Assistenten mit der Tastatur

- Drücken Sie <UMSCHALT> + <F2>. Es erscheint der Editor-Assistent.

In den folgenden Anweisungen wird beschrieben, wie die Funktion MAX und der Funktionsbaustein CTU mit dem Editor-Assistenten in das Code-Arbeitsblatt eingefügt werden. Dabei wird vorausgesetzt, dass der Editor-Assistent bereits geöffnet ist. Ist dies nicht der Fall, rufen Sie den Editor-Assistenten entsprechend der oben beschriebenen Vorgehensweise auf.



Einfügen einer MAX-Funktion über den Editor-Assistenten mit der Maus

- Legen Sie fest, an welcher Stelle im Code-Arbeitsblatt die neue Funktion eingefügt werden soll. Klicken Sie mit der linken Maustaste, um den Textcursor an der gewünschten Stelle zu positionieren.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Wählen Sie im Editor-Assistenten aus dem Listenfeld 'Gruppe' die Gruppe 'Funktionen'. Die verfügbaren Funktionen werden im Auswahlbereich angezeigt.
- Suchen Sie die Funktion 'MAX'.
- Doppelklicken Sie auf die Funktion 'MAX'. Die Funktion wird automatisch im Arbeitsblatt an der Cursorposition eingefügt, wie in der folgenden Abbildung gezeigt. Überschreiben Sie die grünen Kommentare (diese sind durch runde Klammern und Sternchen gekennzeichnet) mit den benötigten Elementen.



Einfügen eines CTU-Funktionsbausteins über den Editor-Assistenten mit der Maus

- Legen Sie fest, an welcher Stelle im Code-Arbeitsblatt der neue Funktionsbaustein eingefügt werden soll. Klicken Sie mit der linken Maustaste, um den Textcursor an der gewünschten Stelle zu positionieren.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Wählen Sie im Editor-Assistenten aus dem Listenfeld 'Gruppe' die Gruppe 'Funktionsbausteine'. Die verfügbaren Funktionsbausteine werden im Auswahlbereich angezeigt.

- Doppelklicken Sie auf den Funktionsbaustein 'CTU'. Der Dialog 'FB-Instanzen' erscheint. Das Feld 'FB-Instanzen' zeigt den Standard-Instanznamen (z. B. ist für einen CTU der Name 'CTU_n' eingetragen, wobei *n* die erste verfügbare Zahl ist, die für die jeweilige Instanz vergeben werden kann).
Sie haben die folgenden Möglichkeiten, um den Instanznamen für den neuen Funktionsbaustein zu definieren:
 - Geben Sie einen neuen Namen im Feld 'FB-Instanzen' ein.
 - Übernehmen Sie den vorgeschlagenen Namen.
 - Wählen Sie aus dem Listenfeld 'FB-Instanzen' einen bereits vorhandenen Namen.
- Drücken Sie 'OK', um den Dialog zu bestätigen. Wenn Sie einen neuen Instanznamen eingegeben haben, erscheint der Dialog 'Automatische FB-Deklaration'.
- Geben Sie bei Bedarf einen Kommentar ein und bestätigen Sie den Dialog mit 'OK'. Der Funktionsbaustein 'CTU' wird automatisch im Arbeitsblatt an der Cursorposition eingefügt, wie in der folgenden Abbildung gezeigt. Überschreiben Sie die grünen Kommentare (diese sind durch runde Klammern und Sternchen gekennzeichnet) mit den benötigten Elementen.

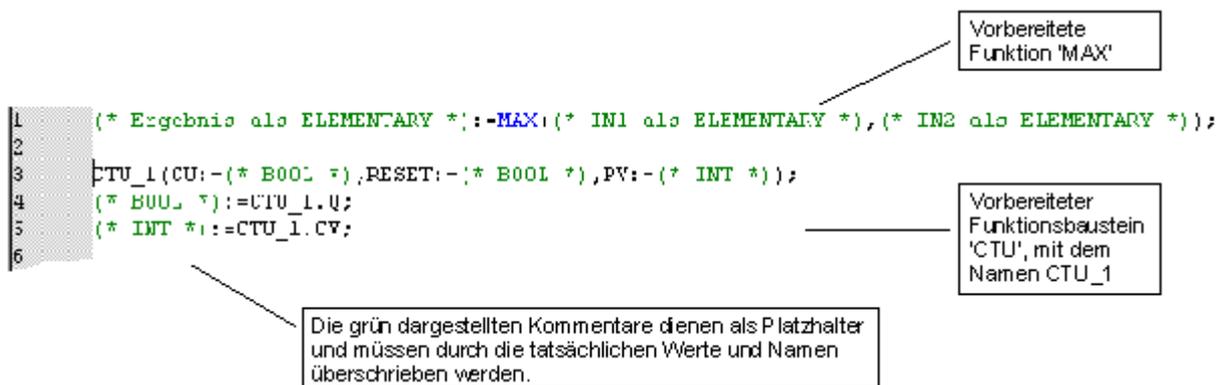


Abbildung 73: Vorbereitete Funktion MAX und Funktionsbaustein CTU, eingefügt mit dem Editor-Assistenten

EDITIEREN IN AWL

7.1 Aufrufen des Texteditors mit einem AWL-Arbeitsblatt

Zum Editieren eines Code-Arbeitsblattes in AWL müssen Sie zunächst aus dem Projektbaum den Texteditor mit dem AWL-Arbeitsblatt aufrufen.



Eine allgemeine Beschreibung des Projektbaums sowie Informationen, wie Sie im Projektbaum durch die POEs blättern, finden Sie in den Kapiteln 'Allgemeines zum Programm' und 'Arbeiten mit Projekten' in diesem Handbuch.

Im folgenden Beispiel nehmen wir an, dass Sie das AWL-Arbeitsblatt eines Funktionsbausteins mit dem Namen IL_FB editieren wollen. Hierzu müssen Sie zunächst einen Funktionsbaustein mit diesem Namen einfügen, wie im Kapitel 'Arbeiten mit Projekten' beschrieben. Beachten Sie dabei, dass die Programmiersprache eines Arbeitsblattes durch die POE-Sprache festgelegt wird. Diese wird beim Einfügen der POE beziehungsweise des Arbeitsblattes definiert.



Aufrufen des Texteditors für das AWL-Arbeitsblatt mit der Maus

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den Unterbaum 'IL_FB' mit den POE-Arbeitsblättern durch Doppelklicken auf den POE-Namen.
- Doppelklicken Sie auf das Symbol 'Arbeitsblatt in AWL' des Funktionsbausteins 'IL_FB'.
Es wird der Texteditor mit dem AWL-Arbeitsblatt aufgerufen.



Aufrufen des Texteditors für das AWL-Arbeitsblatt mit der Tastatur

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den Unterbaum 'IL_FB' mit den POE-Arbeitsblättern wie folgt: Drücken Sie <↓> oder <↑>, um den Funktionsbaustein 'IL_FB' zu markieren. Drücken Sie <→>, um den Unterbaum des Funktionsbausteins zu öffnen.



7.2 Anweisungen, Operatoren, Modifier und Operanden

- Drücken Sie <↓> oder <↑>, um das Symbol 'Arbeitsblatt in AWL' des Funktionsbausteins 'IL_FB' zu markieren.
- Drücken Sie <↵>. Es wird der Texteditor mit dem AWL-Arbeitsblatt aufgerufen.

7.2 Anweisungen, Operatoren, Modifier und Operanden

Eine Anweisungsliste besteht aus mehreren Anweisungen. Jede Anweisung beginnt in einer neuen Zeile und hat einen Operator. Modifier und Operand sind optional. Am Ende einer Zeile kann ein Kommentar unter Verwendung von runden Klammern und Sternchen eingefügt werden. Jede Zeile beginnt mit einer Zeilennummer. Die Syntaxhervorhebung gibt die einzelnen Elemente in unterschiedlichen Farben aus, d. h. Operatoren und Modifier erscheinen blau, Variablen und Operanden schwarz und Kommentare grün. Im folgenden Beispiel wird eine Anweisungsliste mit drei Anweisungen dargestellt:

```
LD    %IX2.2      (* Eingangsparmeter *)
ADD   Wert        (* Additionswert *)
ST    %QX2.2      (* Additionsergebnis *)
```

Abbildung 74: Beispiel einer Anweisungsliste

In diesem Beispiel besteht jede Anweisung aus einem Operator, einem Operanden und einem Kommentar.

Beim Editieren in AWL können Sie folgende Operatoren, Modifier und Operanden verwenden:

Operator	Modifier	Operand	Beschreibung
LD	N	ANY	Setzt aktuelles Ergebnis dem Operanden gleich.
ST	N	ANY	Speichert aktuelles Ergebnis auf die Operanden-Adresse.
S		BOOL	Setzt booleschen Operand auf 1, wenn aktuelles Ergebnis 1 ist.
R		BOOL	Setzt booleschen Operand auf 0, wenn aktuelles Ergebnis 1 ist.
AND	N,(ANY_BIT	Boolesches UND
OR	N,(ANY_BIT	Boolesches ODER
XOR	N,(ANY_BIT	Boolesches Exklusiv-ODER
ADD	(ANY_NUM	Addition
SUB	(ANY_NUM	Subtraktion

Operator	Modifizier	Operand	Beschreibung
MUL	(ANY_NUM	Multiplikation
DIV	(ANY_NUM	Division
GT	(ANY_NUM + ANY_BIT	Vergleich: >
GE	(ANY_NUM + ANY_BIT	Vergleich: >=
EQ	(ANY_NUM + ANY_BIT	Vergleich: =
NE	(ANY_NUM + ANY_BIT	Vergleich: <>
LE	(ANY_NUM + ANY_BIT	Vergleich: <=
LT	(ANY_NUM + ANY_BIT	Vergleich: <
JMP	C, N	LABEL	Sprung zur Marke
CAL	C, N	NAME	Aufruf Funktionsbaustein
RET	C, N		Rücksprung zu aufrufender Funktion oder aufrufendem Programm.
)			Bearbeitung zurückgestellter Operation.

Abbildung 75: Operatoren, Modifizier und Operanden in AWL

Die Bedeutung der Modifizier ist in folgender Tabelle beschrieben:

Modifizier	Beschreibung
N	Negation
(verarbeitet den eingeschlossenen Ausdruck zuerst.
C	bedingt

Abbildung 76: Modifizier in AWL und ihre Bedeutung

7.3 Einfügen von Anweisungen mit dem Editor-Assistenten

Anweisungen können entweder direkt in das Arbeitsblatt eingegeben oder mit dem Editor-Assistenten eingefügt werden. Der Editor-Assistent vereinfacht das Editieren von Arbeitsblättern mit dem Texteditor. In AWL können mit dem Editor-Assistenten Standard-Operatoren, Funktionen und Funktionsbausteine in die entsprechenden Arbeitsblätter eingefügt werden.



Allgemeine Informationen zum Editor-Assistenten finden Sie im Abschnitt 'Der Editor-Assistent' im Kapitel 'Allgemeines zum Programm' in diesem Handbuch.

Der Editor-Assistent bietet die folgenden Vorteile:

- * Syntaktische Fehler wie z. B. falsche Anweisungsfolgen werden vermieden, indem vorbereitete Operatoren, Funktionen und Funktionsbausteine eingefügt werden. Die spezifischen Variablen und Werte erscheinen als Kommentare, die anschließend einfach überschrieben werden müssen.
- * Es ist nicht notwendig, dass der Anwender die genaue Syntax der verschiedenen Operationen, wie z. B. von Funktionen oder Funktionsbausteinen kennt.

Beispiel: Das folgende Beispiel zeigt den Operator 'ADD' und den Funktionsbaustein 'RS', die jeweils mit dem Editor-Assistenten in das Code-Arbeitsblatt eingefügt wurden.

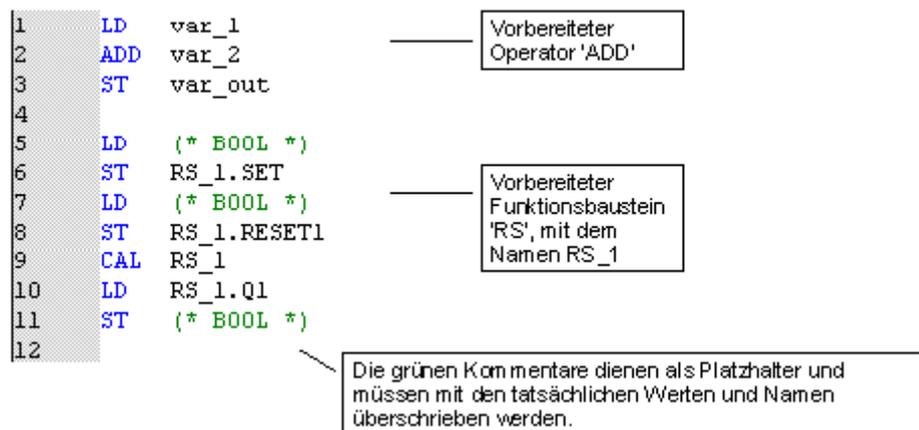


Abbildung 77: Operator 'ADD' und Funktionsbaustein 'RS', beide eingefügt mit dem Editor-Assistenten

Wenn der Editor-Assistent nicht geöffnet ist, führen Sie die folgenden Schritte durch:



Aufrufen des Editor-Assistenten mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Editor-Assistent'. Es erscheint der Editor-Assistent.



Aufrufen des Editor-Assistenten mit der Tastatur

- Drücken Sie <UMSCHALT> + <F2>. Es erscheint der Editor-Assistent.

In der folgenden Anweisung wird beschrieben, wie der Operator ADD mit dem Editor-Assistenten in das Code-Arbeitsblatt eingefügt wird. Dabei wird vorausgesetzt, dass der Editor-Assistent bereits geöffnet ist. Ist dies nicht der Fall, rufen Sie den Editor-Assistenten entsprechend der oben beschriebenen Vorgehensweise auf.



Die Vorgehensweise beim Einfügen von Funktionen und Funktionsbausteinen in das Code-Arbeitsblatt ist identisch mit der Ausnahme, dass im Editor-Assistenten jeweils die entsprechende Funktion bzw. der Funktionsbaustein ausgewählt werden muss. Dieser Vorgang ist detailliert im Abschnitt 'Aufrufen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten' in diesem Kapitel beschrieben.



Einfügen eines ADD-Operators mit dem Editor-Assistenten

- Legen Sie fest, an welcher Stelle im Code-Arbeitsblatt die neuen Anweisungen eingefügt werden sollen. Klicken Sie mit der linken Maustaste, um den Textcursor an der gewünschten Stelle zu positionieren.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Wählen Sie im Editor-Assistenten aus dem Listenfeld 'Gruppe' die Gruppe 'Operatoren'. Die verfügbaren Operatoren werden im Auswahlbereich angezeigt.
- Suchen Sie den Operator 'ADD'.
- Doppelklicken Sie auf den Operator. Der Operator wird automatisch im Arbeitsblatt an der Cursorposition eingefügt, wie in [▶Abbildung 77◀](#) auf Seite 118 gezeigt. Kommentare dienen dabei als Platzhalter für die aktuellen Variablen und Werte. Sie erscheinen grün und sind durch runde Klammern und Sternchen gekennzeichnet.

Im nächsten Schritt müssen die Kommentare, die als Platzhalter dienen, mit den benötigten Variablen und Werten überschrieben werden.



Editieren der neu eingefügten Anweisungen



- Markieren Sie die grünen Platzhalter und überschreiben Sie diese durch die aktuellen Operanden (Variablen und Werte).
- Anschließend müssen Sie die mit dem Operator eingefügten Variablen deklarieren. Doppelklicken Sie dazu auf die Variable, um diese zu markieren. Drücken Sie jetzt <F5>, um den Dialog 'Variable' zu öffnen. In diesem Dialog können Sie bei Bedarf auch den Namen einer Variablen ändern.

7.4 Einfügen von Variablen

Wenn Sie AWL-Anweisungen durch manuelle Eingabe per Tastatur editieren, haben Sie zwei Möglichkeiten, Variablen einzufügen. Die erste Möglichkeit ist, den Variablennamen per Tastatur im Code-Arbeitsblatt einzugeben und die Variable im Texteditor oder im Variablen-Dialog zu editieren.

Sie können Variablen auch über den Dialog 'Variable' einfügen. Dieser erscheint, wenn Sie in der Symbolleiste auf das Symbol 'Variable' klicken. Im zweiten Beispiel am Ende dieses Abschnitts nehmen wir an, dass Sie eine Variable einfügen wollen, die Sie bereits in der globalen Variablen-Deklaration Ihres Projekts editiert haben.

Im ersten Beispiel gehen wir davon aus, dass Sie den Operator 'ADD' bereits mit dem Editor-Assistenten eingefügt haben. Die benötigte Variable 'invar' wurde noch nicht im Variablen-Arbeitsblatt 'IL_FBV' deklariert.



Deklariere eine Variable mit der Maus

- Doppelklicken Sie auf den Namen der Variable, die deklariert werden soll.
- Klicken Sie in der Symbolleiste auf das Symbol 'Variable'. Es erscheint der Dialog 'Variable'. Der aktuelle Variablenname ist im Feld 'Variablenliste für POE-Name' eingetragen.

ADD

invar



Deklariere eine Variable mit der Tastatur

- Halten Sie die <UMSCHALT>-Taste gedrückt und markieren Sie die Variable mit <->>.
- Drücken Sie <F5>. Es erscheint der Dialog 'Variable'. Der aktuelle Variablenname ist im Feld 'Variablenliste für POE-Name' eingetragen.

ADD

invar



HINWEIS

Sie können den Dialog 'Variable' auch öffnen, ohne zuvor eine zu deklarierende Variable markiert zu haben. In diesem Fall ist das Feld 'Variablenliste für POE-Name' leer.



Abbildung 78: Dialog 'Variable' für eine zuvor markierte Variable



Verwenden des Dialogs 'Variable'

- Geben Sie bei Bedarf einen Variablennamen ein (siehe obiger Hinweis).
- Bestätigen Sie den Dialog 'Variable'. Es erscheint der Dialog 'Automatische Variablen-Deklaration'.



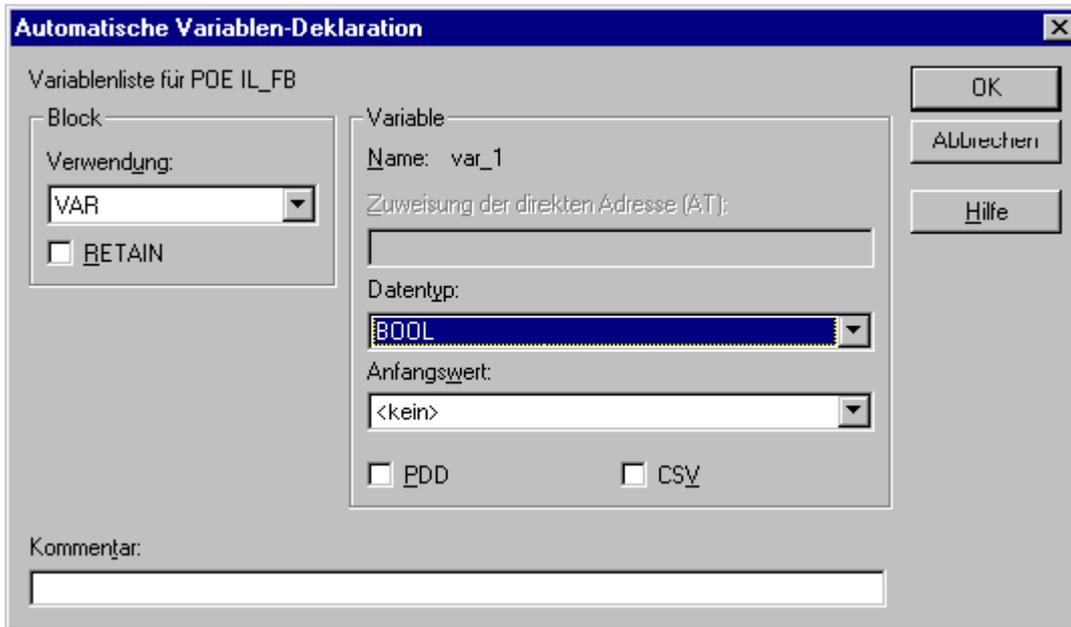


Abbildung 79: Dialog 'Automatische Variablen-Deklaration'



Verwenden des Dialogs 'Automatische Variablen-Deklaration'



- Wählen Sie aus dem Listefeld 'Verwendung' ein Variablen-Schlüsselwort.
- Geben Sie einen Speicherort im Feld 'Zuweisung der direkten Adresse (AT)' ein, wenn Sie eine adressierte Variable deklarieren wollen (dies ist nur in Programmen oder für globale Variablen möglich).
- Wählen Sie den richtigen Datentyp aus dem Feld 'Datentyp'.
- Geben Sie bei Bedarf einen Anfangswert ein.
- Aktivieren Sie das Kontrollkästchen 'PDD', wenn die Variable im OmegaOS PDD (Process Data Directory) gespeichert werden soll.
- Aktivieren Sie das Kontrollkästchen 'CSV', wenn die Variable in der CSV-Datei gespeichert werden soll, d. h. im OPC-Server verwendet werden soll. Der OPC-Server verarbeitet nur Variablen, die in der CSV-Datei eingetragen sind. Diese Variablen können in einem OPC-Client-Prozess verwendet werden (z. B. in einer Visualisierung).
- Geben Sie bei Bedarf einen Kommentar ein.
- Bestätigen Sie den Dialog.
Die neue Variable wird in das Code-Arbeitsblatt eingefügt. Die Deklaration der Variablen wird automatisch in der Variablen-Deklaration der POE vorgenommen.



HINWEIS

Wenn Sie bereits deklarierte Variablen einfügen, erscheint der Name der Variablen im Listefeld des Dialogs 'Variable'. Beim Bestätigen des Dialogs wird die Variable direkt in das Code-Arbeitsblatt eingefügt. Der Dialog 'Automatische Variablen-Deklaration' erscheint dann nicht.



Detaillierte Informationen zum OPC-Server finden Sie im Handbuch zum OPC-Server.

Im zweiten Beispiel nehmen wir an, dass Sie eine Variable einfügen wollen, die Sie bereits zuvor in der globalen Variablen-Deklaration Ihres Projekts deklariert haben.



Öffnen des Dialogs 'Variable' mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Variable'.
Es erscheint der Dialog 'Variable'. Im Feld 'Variablenliste für Ressource: *Ressource-Name*' ist kein Name eingetragen.



Öffnen des Dialogs 'Variable' mit der Tastatur

- Drücken Sie <F5>.
Es erscheint der Dialog 'Variable'. Im Feld 'Variablenliste für Ressource: *Ressource-Name*' ist kein Name eingetragen.



Abbildung 80: Dialog 'Variable'



Verwenden des Dialogs 'Variable' beim Einfügen einer Variable, die bereits in der globalen Variablen-Deklaration des Projekts deklariert wurde



- Aktivieren Sie die Option 'Global'.
- Überprüfen Sie, ob die richtige Ressource verwendet wird.
- Wählen Sie aus dem Listenfeld 'Variablenliste für Ressource: *Ressource-Name*' die gewünschte Variable.
- Bestätigen Sie den Dialog.
Die Variable wird in das Code-Arbeitsblatt und in die Variablen-Deklaration der POE als VAR_EXTERNAL eingefügt, da es sich um eine globale Variable handelt.

7.5 Verwenden von Sprüngen und Marken

Sprünge können verwendet werden, um in eine Zeile der Anweisungsliste zu springen. Hierzu werden der Operator 'JMP' und eine Marke vor der Zeile verwendet, in die gesprungen werden soll, wie im folgenden Beispiel dargestellt:

```
LD      Wert1
EQ      INT#100
JMPC   Marke
LD      Wert2
ADD     Wert3
ST      Wert4
Marke:  LD      %IX2.2
```

Abbildung 81: Beispiel für einen Sprung

Im Beispiel erfolgt ein Sprung zur Marke, wenn 'Wert1' 100 beträgt. Beträgt er nicht 100, erfolgt kein Sprung und 'Wert2' wird geladen.

7.6 Aufrufen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten

Funktionen können in einer Anweisungsliste aufgerufen werden, indem der Funktionsname in das Operatorfeld eingetragen wird. In der folgenden Abbildung ist ein Beispiel für den Aufruf einer Funktion mit drei Eingangsparametern und einem Ausgangsparameter dargestellt:

```
LD      inpar1
Funktionsname  inpar2, inpar3
ST      var1
```

Abbildung 82: Beispiel für das Aufrufen einer Funktion

Der erste deklarierte Eingangsparameter muss in der ersten Zeile des vorhergehenden Funktionsaufrufes geladen werden. Im Beispiel ist der erste Eingangsparameter 'inpar1'. Alle weiteren Eingangsparameter müssen in der zweiten Zeile als Operanden verwendet und durch Kommata getrennt werden. Das Ergebnis wird in der Variablen 'var1' gespeichert, wie in der letzten Zeile der Abbildung gezeigt.

HINWEIS



Auf diese Weise können sowohl Standardfunktionen als auch anwenderdefinierte Funktionen aufgerufen werden.

Funktionsbausteine können aufgerufen werden, indem der Operator CAL mit dem Namen des Funktionsbausteins als Operand verwendet wird. Beim Aufruf von Funktionsbausteinen ist die Eingabe aufwendiger als für den Aufruf einer Funktion. Im nächsten Beispiel wird ein Funktionsbaustein mit zwei Eingangsparametern 'inpar1' und 'inpar2' und zwei Ausgangsparametern 'outpar1' und 'outpar2' beschrieben. Der Name des Funktionsbausteins ist 'FB_exam'. Der Instanzname ist 'instance'.

In diesem Beispiel soll der Aufruf des Funktionsbausteins aussehen, wie in folgender Abbildung:

```
LD    var1
ST    instance.inpar1
LD    var2
ST    instance.inpar2
CAL   instance
LD    instance.outpar1
ST    var3
LD    instance.outpar2
ST    var4
```

Abbildung 83: Beispiel für den Aufruf eines Funktionsbausteins

Der Aufruf eines Funktionsbausteins besteht aus drei Teilen: Dem Einbringen der Eingangsparameter, dem eigentlichen Aufruf mit dem Operator CAL und dem Speichern der Ausgangsparameter.

HINWEIS



Denken Sie daran, die Instanz für den Funktionsbaustein in der Variablen-Deklaration der POE zu deklarieren, wie im Kapitel 'Literele, Datentypen und Variablen' beschrieben.

Mit dem Editor-Assistenten können sehr einfach und bequem Funktionen oder Funktionsbausteine in das Code-Arbeitsblatt eingefügt werden und somit Fehleingaben vermieden werden.

Wenn der Editor-Assistent nicht geöffnet ist, führen Sie die folgenden Schritte durch:

Aufrufen des Editor-Assistenten mit der Maus



- Klicken Sie in der Symbolleiste auf das Symbol 'Editor-Assistent'. Es erscheint der Editor-Assistent.



Aufrufen des Editor-Assistenten mit der Tastatur



- Drücken Sie <UMSCHALT> + <F2>. Es erscheint der Editor-Assistent.

In den folgenden Anweisungen wird beschrieben, wie die Funktion MAX und der Funktionsbaustein CTU mit dem Editor-Assistenten in das Code-Arbeitsblatt eingefügt werden. Dabei wird vorausgesetzt, dass der Editor-Assistent bereits geöffnet ist. Ist dies nicht der Fall, rufen Sie den Editor-Assistenten entsprechend der oben beschriebenen Vorgehensweise auf.



Einfügen einer MAX-Funktion über den Editor-Assistenten mit der Maus

- Legen Sie fest, an welcher Stelle im Code-Arbeitsblatt die neue Funktion eingefügt werden soll. Klicken Sie mit der linken Maustaste, um den Textcursor an der gewünschten Stelle zu positionieren.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Wählen Sie im Editor-Assistenten aus dem Listenfeld 'Gruppe' die Gruppe 'Funktionen'. Die verfügbaren Funktionen werden im Auswahlbereich angezeigt.
- Suchen Sie die Funktion 'MAX'.
- Doppelklicken Sie auf die Funktion 'MAX'. Die Funktion wird automatisch im Arbeitsblatt an der Cursorposition eingefügt, wie in der folgenden Abbildung gezeigt. Überschreiben Sie die grünen Kommentare (diese sind durch runde Klammern und Sternchen gekennzeichnet) mit den benötigten Elementen.



Einfügen eines CTU-Funktionsbausteins über den Editor-Assistenten mit der Maus

- Legen Sie fest, an welcher Stelle im Code-Arbeitsblatt der neue Funktionsbaustein eingefügt werden soll. Klicken Sie mit der linken Maustaste, um den Textcursor an der gewünschten Stelle zu positionieren.
- Drücken Sie <↵>, um eine neue Zeile einzufügen.
- Wählen Sie im Editor-Assistenten aus dem Listenfeld 'Gruppe' die Gruppe 'Funktionsbausteine'. Die verfügbaren Funktionsbausteine werden im Auswahlbereich angezeigt.
- Doppelklicken Sie auf den Funktionsbaustein 'CTU'. Der Dialog 'FB-Instanzen' erscheint. Das Feld 'FB-Instanzen' zeigt den Standard-Instanznamen (z. B. ist für einen CTU der Name 'CTU_n' eingetragen, wobei n die erste verfügbare Zahl ist, die für die jeweilige Instanz vergeben werden kann).
Sie haben die folgenden Möglichkeiten, um den Instanznamen für den neuen Funktionsbaustein zu definieren:
 - Geben Sie einen neuen Namen im Feld 'FB-Instanzen' ein.
 - Übernehmen Sie den vorgeschlagenen Namen.
 - Wählen Sie aus dem Listenfeld 'FB-Instanzen' einen bereits vorhandenen Namen.
- Drücken Sie 'OK', um den Dialog zu bestätigen. Wenn Sie einen neuen Instanznamen eingegeben haben, erscheint der Dialog 'Automatische FB-Deklaration'.
- Geben Sie bei Bedarf einen Kommentar ein und bestätigen Sie den Dialog mit 'OK'. Der Funktionsbaustein 'CTU' wird automatisch im Arbeitsblatt an der Cursorposition eingefügt, wie in der folgenden Abbildung gezeigt. Überschreiben Sie die grünen Kommentare (diese sind durch runde Klammern und Sternchen gekennzeichnet) mit den benötigten Elementen.

7.6 Aufrufen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten

```
1 LD (* IN1 als ELEMENTARY *)
2 MAX (* IN2 als ELEMENTARY *)
3 ST (* Ergebnis als ELEMENTARY *)
4
5 LD (* BOOL *)
6 ST CTU_1.CU
7 LD (* BOOL *)
8 ST CTU_1.RESET
9 LD (* INT *)
10 ST CTU_1.PV
11 CAL CTU_1
12 LD CTU_1.Q
13 ST (* BOOL *)
14 LD CTU_1.CV
15 ST (* INT *)
16
```

Vorbereitete Funktion 'MAX'

Vorbereiteter Funktionsbaustein 'CTU' mit dem Namen CTU_1

Die grünen Kommentare dienen als Platzhalter und müssen mit den tatsächlichen Werten und Namen überschrieben werden.

Abbildung 84: Vorbereitete Funktion MAX und Funktionsbaustein CTU, eingefügt mit dem Editor-Assistenten

EDITIEREN IN FBS

8.1 Aufrufen des graphischen Editors mit einem FBS-Arbeitsblatt

Zum Editieren eines Code-Arbeitsblattes in FBS müssen Sie zunächst aus dem Projektbaum den graphischen Editor mit dem FBS-Arbeitsblatt aufrufen.



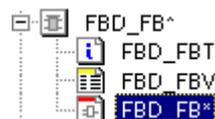
Eine allgemeine Beschreibung des Projektbaums sowie Informationen, wie Sie im Projektbaum durch die POEs blättern, finden Sie in den Kapiteln 'Allgemeines zum Programm' und 'Arbeiten mit Projekten' in diesem Handbuch.

Im folgenden Beispiel nehmen wir an, dass Sie das FBS-Arbeitsblatt eines Funktionsbausteins mit dem Namen FBD_FB editieren wollen. Hierzu müssen Sie zunächst einen Funktionsbaustein mit diesem Namen einfügen, wie im Kapitel 'Arbeiten mit Projekten' beschrieben. Beachten Sie dabei, dass die Programmiersprache eines Arbeitsblattes durch die POE-Sprache festgelegt wird. Diese wird beim Einfügen der POE beziehungsweise des Arbeitsblattes definiert.



Aufrufen des graphischen Editors für das FBS-Arbeitsblatt mit der Maus

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den Unterbaum 'FBD_FB' mit den POE-Arbeitsblättern durch Doppelklicken auf den POE-Namen.
- Doppelklicken Sie auf das Symbol 'Arbeitsblatt in FBS' des Funktionsbausteins 'FBD_FB'. Es wird der graphische Editor mit dem FBS-Arbeitsblatt aufgerufen.





Aufrufen des graphischen Editors für das FBS-Arbeitsblatt mit der Tastatur

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den entsprechenden Unterbaum mit den POE-Arbeitsblättern wie folgt: Drücken Sie <↓> oder <↑>, um den Funktionsbaustein 'FBD_FB' zu markieren. Drücken Sie <→>, um den Unterbaum des Funktionsbausteins zu öffnen.
- Drücken Sie <↓> oder <↑>, um das Symbol 'Arbeitsblatt in FBS' des Funktionsbausteins 'FBD_FB' zu markieren.
- Drücken Sie <↵>. Es wird der graphische Editor mit dem FBS-Arbeitsblatt aufgerufen.



8.2 Einführung in FBS

Der Code in der graphischen Sprache FBS (Funktionsbaustein-Sprache) setzt sich aus Funktionen, Funktionsbausteinen und Variablen zusammen, die durch Linien miteinander verbunden sind. Zusätzlich kann eine Linie mit einer anderen Linie verbunden werden. Die verbundenen Objekte zusammen bilden ein FBS-Netzwerk. In FBS-Netzwerken ist es nicht möglich, Ausgänge mit Ausgängen zu verbinden.

In FBS-Codes können Kommentare unter Verwendung des Kontextmenüs Menüpunkt 'Text (Anmerkung)...' eingefügt werden.

Funktionen und Funktionsbausteine können mit dem Editor-Assistenten eingefügt werden und im Dialog 'Funktion/Funktionsbaustein' editiert werden. Beide Vorgänge sind in diesem Kapitel beschrieben. Der Editor-Assistent vereinfacht das Editieren von Arbeitsblättern mit dem graphischen Editor. Er enthält alle Funktionen und Funktionsbausteine, die in den FBS-Arbeitsblättern eingefügt werden können.

Die graphische Programmiersprache FBS bietet viele zusätzliche Leistungsmerkmale, die das Erstellen des Code für Programme, Funktionsbausteine oder Funktionen erleichtert. Dies sind z. B.:

- * Einfache Bedienung mit der Tastatur, d. h. Sie können Elemente einfügen und den sichtbaren Bereich des Arbeitsblattes verschieben. Objekte können mit den Cursor-Tasten in Verbindung mit der <STRG>-Taste bearbeitet werden. Der Mauszeiger wird mit den Cursor-Tasten in Verbindung mit der <UMSCHALT>-Taste und <STRG>-Taste bewegt.
- * Duplizieren von Eingängen direkt mit der Tastatur oder über die Symbolleiste oder Menüpunkte.
- * Negieren von Eingängen, Ausgängen, Kontakten und Spulen direkt mit der Tastatur oder über die Symbolleiste oder Menüpunkte.
- * Einfaches Autorouting.
- * Elemente können direkt auf eine Linie gesetzt werden oder mit den Eingängen oder Ausgängen von vorhandenen Elementen verbunden werden.
- * Sie können Fenster splitten und Übersichtsfenster öffnen.
- * Elemente können in beliebiger Form angeordnet werden.
- * Durch Doppelklicken auf anwenderdefinierte Funktionen und Funktionsbausteine wird automatisch das zugehörige POE-Arbeitsblatt geöffnet.

8.3 Einfügen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten



Allgemeine Informationen zum Editor-Assistenten finden Sie im Abschnitt 'Der Editor-Assistent' im Kapitel 'Allgemeines zum Programm' in diesem Handbuch.

Der Editor-Assistent in FBS-Arbeitsblättern enthält Funktionen und Funktionsbausteine, die sehr einfach in das Code-Arbeitsblatt eingefügt werden können.

Wenn der Editor-Assistent nicht geöffnet ist, führen Sie die folgenden Schritte durch:



Aufrufen des Editor-Assistenten mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Editor-Assistent'. Es erscheint der Editor-Assistent.



Aufrufen des Editor-Assistenten mit der Tastatur

- Drücken Sie <UMSCHALT> + <F2>. Es erscheint der Editor-Assistent.

In der folgenden Anweisung wird beschrieben, wie der Funktionsbaustein CTU mit dem Editor-Assistenten in das FBS-Arbeitsblatt eingefügt wird. Dabei wird vorausgesetzt, dass der Editor-Assistent bereits geöffnet ist. Ist dies nicht der Fall, rufen Sie den Editor-Assistenten entsprechend der oben beschriebenen Vorgehensweise auf.



Einfügen des CTU-Funktionsbausteins über den Editor-Assistenten mit der Maus

- Blenden Sie bei Bedarf das Gitter im Arbeitsbereich ein. Wählen Sie dazu aus dem Untermenü 'Layout' den Menüpunkt 'Gitter'.
- Klicken Sie in das Arbeitsblatt, um eine Einfügemarke zu setzen.
- Wählen Sie im Editor-Assistenten aus dem Listenfeld 'Gruppe' die Gruppe 'Funktionsbausteine'. Die verfügbaren Funktionsbausteine werden im Auswahlbereich angezeigt.
- Doppelklicken Sie auf den Funktionsbaustein 'CTU'. Der Dialog 'FB-Instanzen' erscheint. Das Feld 'FB-Instanzen' zeigt den Standard-Instanznamen (z. B. ist für einen CTU der Name 'CTU_n' eingetragen, wobei n die erste verfügbare Zahl ist, die für die jeweilige Instanz vergeben werden kann).
Sie haben die folgenden Möglichkeiten, um den Instanznamen für den neuen Funktionsbaustein zu definieren:
 - Geben Sie einen neuen Namen im Feld 'FB-Instanzen' ein.
 - Übernehmen Sie den vorgeschlagenen Namen.
 - Wählen Sie aus dem Listenfeld 'FB-Instanzen' einen bereits vorhandenen Namen.
- Geben Sie bei Bedarf einen Kommentar ein und bestätigen Sie den Dialog mit 'OK'. Der Funktionsbaustein 'CTU' wird automatisch in das Arbeitsblatt eingefügt und die Deklaration der Funktionsbaustein-Instanz wird automatisch in der Variablen-Deklaration der POE vorgenommen.

8.4 Ändern der Eigenschaften von Funktionen und Funktionsbausteinen

In den folgenden Schritten werden die Eigenschaften der/des eingefügten Funktion/Funktionsbausteins editiert und das Element mit einem FBS-Netzwerk verbunden.



Das Ändern der Eigenschaften von Funktionen und Funktionsbausteinen wird im folgenden Abschnitt 'Ändern der Eigenschaften von Funktionen und Funktionsbausteinen' beschrieben. Wie Objekte verbunden werden, wird im Abschnitt 'Verbinden von Objekten' in diesem Kapitel beschrieben.

8.4 Ändern der Eigenschaften von Funktionen und Funktionsbausteinen

Die Eigenschaften von Funktionen und Funktionsbausteinen im FBS-Arbeitsblatt können Sie im Dialog 'Funktion/Funktionsbaustein' ändern.



Öffnen des Dialogs 'Funktion/Funktionsbaustein' für eine Funktion bzw. einen Funktionsbaustein

- Klicken Sie im FBS-Arbeitsblatt auf die zu editierende Funktion oder den Funktionsbaustein. Das markierte Objekt wird hervorgehoben.
- Klicken Sie im FBS-Arbeitsblatt mit der rechten Maustaste auf die zu editierende Funktion oder den Funktionsbaustein, um das Kontextmenü zu öffnen.
- Wählen Sie den Menüpunkt 'Objekteigenschaften...'. Es erscheint der Dialog 'Funktion/Funktionsbaustein'.



Abbildung 85: Dialog 'Funktion/Funktionsbaustein'



Verwenden des Dialogs 'Funktion/Funktionsbaustein'



- Die Listenfelder 'Gruppe' und 'Name' sind inaktiv, da das Objekt bereits erzeugt wurde.
- Ändern Sie bei Bedarf den Instanznamen.
- Bestätigen Sie den Dialog.

Wenn Sie den Instanznamen eines Funktionsbausteins geändert haben, erscheint der Dialog 'Automatische FB-Deklaration'. Geben Sie bei Bedarf einen Kommentar ein und bestätigen Sie den Dialog mit 'OK'. Die Deklaration der Funktionsbaustein-Instanz wird automatisch in der Variablen-Deklaration der POE vorgenommen.

8.5 Ersetzen von Funktionen und Funktionsbausteinen

Sie können eine bestimmte Funktion oder einen Funktionsbaustein durch ein anderes Objekt mit Hilfe des Editor-Assistenten oder über den Dialog 'Funktion/Funktionsbaustein' ersetzen.



Ersetzen eines Objektes mit dem Editor-Assistenten

- Klicken Sie auf das zu ersetzende Objekt. Das markierte Objekt wird hervorgehoben.
- Ersetzen Sie das markierte Objekt, indem Sie im Editor-Assistenten im Auswahlbereich auf die neue Funktion oder den Funktionsbaustein doppelklicken. Detaillierte Informationen zum Einfügen von Funktionen und Funktionsbausteinen finden Sie im Abschnitt 'Einfügen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten'.
- Editieren Sie bei Bedarf das neue Objekt, wie im Abschnitt 'Ändern der Eigenschaften von Funktionen und Funktionsbausteinen' beschrieben.



Ersetzen eines Objektes über den Dialog 'Funktion/Funktionsbaustein'

- Klicken Sie auf das zu ersetzende Objekt. Das markierte Objekt wird hervorgehoben.
- Klicken Sie mit der rechten Maustaste auf das markierte Objekt, um das Kontextmenü zu öffnen. Wählen Sie den Menüpunkt 'Objekteigenschaften...'. Es erscheint der Dialog 'Funktion/Funktionsbaustein'. Die Listenfelder 'Gruppe' und 'Name' sind inaktiv.
- Aktivieren Sie das Kontrollkästchen 'FB/FU ersetzen'. Die Listenfelder 'Gruppe' und 'Name' werden aktiv.
- Wählen Sie aus den Listenfeldern 'Gruppe' und 'Name' die benötigten Einträge für das neu einzufügende Objekt. Ein Instanzname wird vom System vorgeschlagen.
- Editieren Sie das neue Objekt, wie im Abschnitt 'Ändern der Eigenschaften von Funktionen und Funktionsbausteinen' beschrieben.

8.6 Einfügen von Variablen

Sie haben zwei Möglichkeiten, Variablen in ein FBS-Arbeitsblatt einzufügen:

- * Einfügen einer Variablen, die bereits zuvor deklariert wurde. Siehe hierzu Abschnitt 'Deklariieren von Variablen' im Kapitel 'Literale, Datentypen und Variablen'.
- * Einfügen und deklarieren einer neuen Variablen.

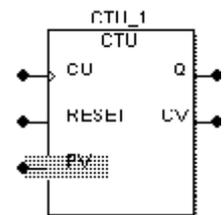
Variablen können an jeder beliebigen Stelle im FBS-Arbeitsblatt eingefügt werden oder direkt mit einem Eingang/Ausgang (Formalparameter) einer Funktion oder eines Funktionsbausteins verbunden werden.

In den nächsten Schritten nehmen wir an, dass Sie eine Variable einfügen wollen, die direkt mit dem Formalparameter 'PV' des Funktionsbausteins 'CTU' verbunden wird. Die Variable ist zuvor noch nicht deklariert worden.



Einfügen einer Variablen am Formalparameter 'PV' mit der Maus

- Klicken Sie auf den Formalparameter 'PV', um ihn zu markieren. Der markierte Eingang ändert die Farbe.



- Klicken Sie in der Symbolleiste auf das Symbol 'Variable'. Es erscheint der Dialog 'Variable'.



Einfügen einer Variablen am Formalparameter 'PV' mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten den Formalparameter 'PV'.
- Drücken Sie <F5>. Es erscheint der Dialog 'Variable'.



Abbildung 86: Dialog 'Variable'



Verwenden des Dialogs 'Variable'



- Geben Sie den Variablennamen 'VAR1' ein.
- Bestätigen Sie den Dialog.
Es erscheint der Dialog 'Automatische Variablen-Deklaration'.

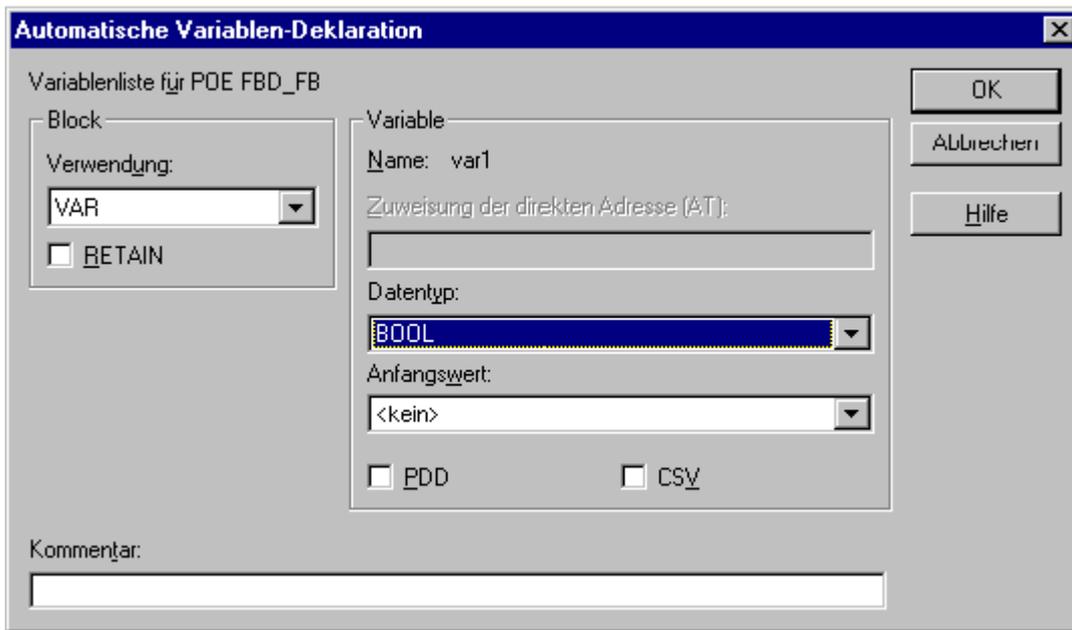


Abbildung 87: Dialog 'Automatische Variablen-Deklaration'



Verwenden des Dialogs 'Automatische Variablen-Deklaration'



- Wählen Sie aus dem Listefeld 'Verwendung' ein Variablen-Schlüsselwort.
- Geben Sie einen Speicherort im Feld 'Zuweisung der direkten Adresse (AT)' ein, wenn Sie eine adressierte Variable deklarieren wollen (dies ist nur in Programmen oder für globale Variablen möglich).
- Wählen Sie den richtigen Datentyp aus dem Feld 'Datentyp'.
- Geben Sie bei Bedarf einen Anfangswert ein.
- Aktivieren Sie das Kontrollkästchen 'PDD', wenn die Variable im OmegaOS PDD (Process Data Directory) gespeichert werden soll, d. h. in Kommunikations-Funktionsbausteinen der IEC 61131-5 verwendet werden soll.
- Aktivieren Sie das Kontrollkästchen 'CSV', wenn die Variable in der CSV-Datei gespeichert werden soll, d. h. im OPC-Server verwendet werden soll. Der OPC-Server verarbeitet nur Variablen, die in der CSV-Datei eingetragen sind. Diese Variablen können in einem OPC-Client-Prozess verwendet werden (z. B. in einer Visualisierung).
- Geben Sie bei Bedarf einen Kommentar ein.



- Bestätigen Sie den Dialog.
Die neue Variable wird in das Code-Arbeitsblatt eingefügt. Die Deklaration der Variablen wird automatisch in der Variablen-Deklaration der POE vorgenommen.



HINWEIS

Wenn Sie bereits deklarierte Variablen einfügen, erscheint der Name der Variablen im Listenfeld des Dialogs 'Variable'. Beim Bestätigen des Dialogs wird die Variable direkt in das Code-Arbeitsblatt eingefügt. Der Dialog 'Automatische Variablen-Deklaration' erscheint dann nicht.



Detaillierte Informationen zum OPC-Server finden Sie im Handbuch zum OPC-Server.

8.7 Verbinden von Objekten

Nachdem Sie Funktionen, Funktionsbausteine oder Variablen eingefügt haben, müssen Sie diese miteinander verbinden, um ein gültiges FBS-Netzwerk herzustellen. Sie haben die folgenden Möglichkeiten, Objekte in einem FBS-Netzwerk zu verbinden:

- * Verbinden von Objekten mit dem Verbindungsmodus
- * Verbinden von Objekten durch Drag & Drop mit der Maus
- * Verbinden von Objekten direkt beim Einfügen eines neuen Objektes

Im Verbindungsmodus können Sie alle Objekte mit nicht belegten Verbindungspunkten miteinander verbinden. Verbindungspunkte werden als grüne und blaue Punkte dargestellt. Im Verbindungsmodus können Sie auch eine neue Linie an einer bestehenden Linie anknüpfen.



HINWEIS

Objekte können nur mit horizontalen Linien verbunden werden. Die Linien müssen also an einem Verbindungspunkt eines Objektes beginnen und horizontal von dem Objekt weg gezogen werden.

Ein nützliches Werkzeug beim Verbinden von Eingängen und Ausgängen von Funktionen und Funktionsbausteinen ist das Autorouting. Hier wird vom System automatisch festgelegt, wie die Linie zwischen zwei nicht belegten Verbindungspunkten im Netzwerk geführt wird. In diesem Fall muss z. B. eine Linie nicht um eine Ecke geführt werden.



HINWEIS

Autorouting ist nur möglich, wenn Sie einen Eingang mit einem Ausgang oder umgekehrt verbinden. Wenn Sie eine Rückkopplung erzeugen oder die Linie an einem Verbindungspunkt beginnt, ist Autorouting nicht möglich.

In den nächsten Anweisungen nehmen wir an, dass Sie den Funktionsbaustein 'CTU' mit der Funktion 'ADD' verbinden wollen.



Verbinden eines Funktionsbaustein-Ausgangs mit einem Funktionseingang im Verbindungsmodus mit Autorouting

- Klicken Sie in der Symbolleiste auf das Symbol 'Objekte verbinden'. Unter dem Cursor erscheint eine symbolische Linie. Bei aktiviertem Verbindungsmodus erscheint das Symbol in der Symbolleiste gedrückt.
- Klicken Sie auf den Ausgang des Funktionsbausteins 'CTU', den Sie verbinden möchten.
- Bewegen Sie den Cursor zu dem Eingang der Funktion 'ADD', den Sie verbinden möchten. Die Verbindungslinie erscheint rot. Wenn das System den gewünschten Verbindungspunkt erkennt, wird die Linienführung automatisch durchgeführt. Die Farbe der Linie ändert sich dann nach grün.
- Klicken Sie auf den Eingang.
Die neue Verbindungslinie wird automatisch vom Ausgang zum gewünschten Eingang im Netzwerk geführt und eingefügt.



Verbinden von zwei Objekten im Verbindungsmodus ohne Autorouting

- Klicken Sie in der Symbolleiste auf das Symbol 'Objekte verbinden'. Unter dem Cursor erscheint eine symbolische Linie. Bei aktiviertem Verbindungsmodus erscheint das Symbol in der Symbolleiste gedrückt.
- Klicken Sie auf die Stelle, an der die Linie beginnen soll.
- Bewegen Sie den Cursor zu dem Objekt, das Sie verbinden möchten.
- Klicken Sie mit der linken Maustaste, um die Verbindung bei Bedarf um eine Ecke zu führen.
- Klicken Sie auf den gewünschten Endpunkt der Linie.



Verbinden von zwei Objekten mit Drag & Drop mit der Maus

- Klicken Sie auf die Funktion 'ADD' und halten die Maustaste gedrückt.
- Bewegen Sie den Cursor zum Funktionsbaustein 'CTU', so dass sich die Verbindungspunkte überschneiden.
- Lassen Sie die Maustaste los.
Die Verbindung ist erstellt.
- Bewegen Sie bei Bedarf die Funktion oder den Funktionsbaustein an eine freie Stelle im Netzwerk.

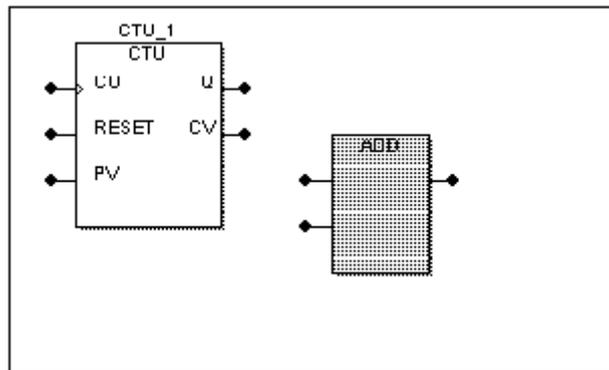


Abbildung 88: Funktion und Funktionsbaustein vor dem Verbinden

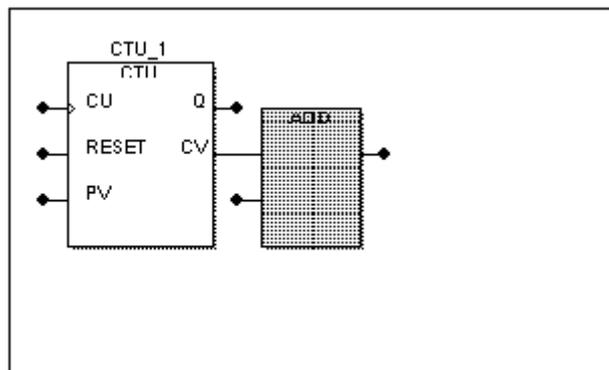


Abbildung 89: Die Verbindung ist hergestellt

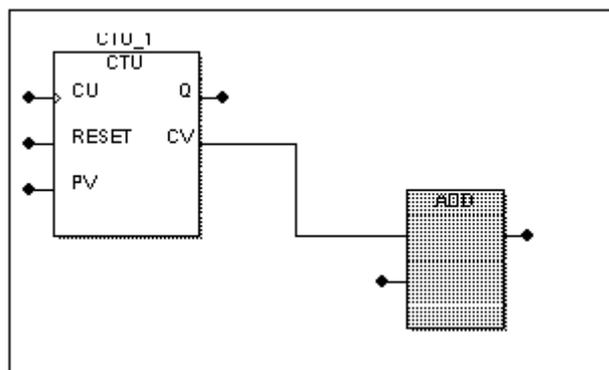


Abbildung 90: Die Funktion wurde an eine freie Stelle verschoben. Die Verbindungslinie wird automatisch geführt.

Sie können auch eine Funktion oder einen Funktionsbaustein mit einer anderen Funktion oder einem anderen Funktionsbaustein verbinden, indem Sie einen Ausgangsparameter markieren und dann das Objekt wie nachfolgend beschrieben einfügen.



Verbinden eines Objektes mit einer neuen Funktion/einem neuen Funktionsbaustein beim Einfügen

- Klicken Sie auf den Ausgang 'CV' des Funktionsbausteins 'CTU', bevor Sie die Funktion 'ADD' einfügen. Der markierte Ausgang ändert die Farbe.
- Fügen Sie mit dem Editor-Assistenten die neue Funktion 'ADD' ein, wie im Abschnitt 'Einfügen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten' beschrieben. Die Verbindung zwischen dem markierten Ausgang 'CV' und dem ersten Eingang der Funktion 'ADD' wird automatisch erstellt.
- Bewegen Sie bei Bedarf die Funktion oder den Funktionsbaustein an eine freie Stelle im Netzwerk.

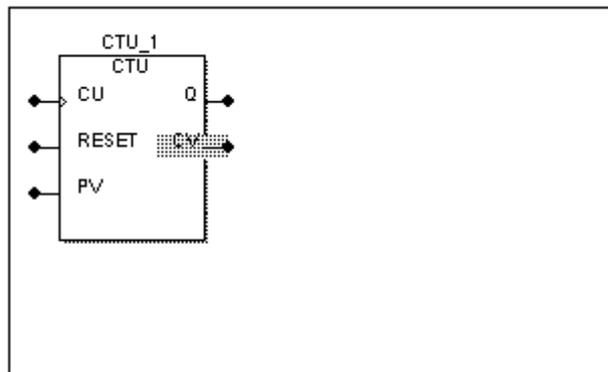


Abbildung 91: Der Ausgang 'CV' des Funktionsbausteins 'CTU' wird als gewünschter Verbindungspunkt markiert, bevor die Funktion 'ADD' eingefügt wird

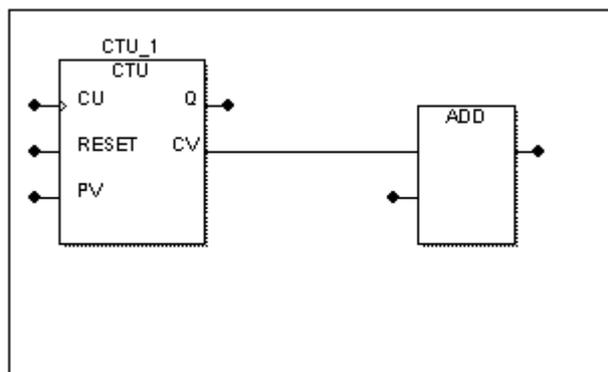


Abbildung 92: Die Verbindung wird automatisch beim Einfügen der Funktion 'ADD' erstellt

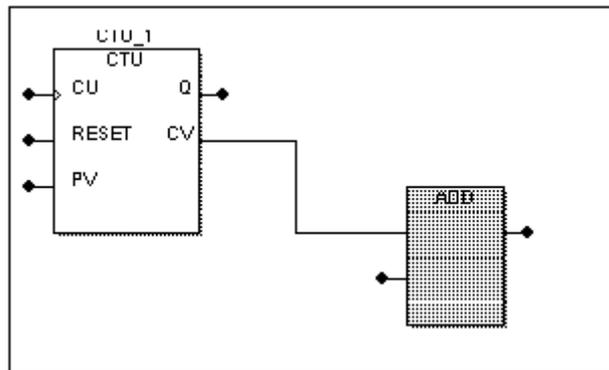


Abbildung 93: Die Funktion wurde an eine freie Stelle verschoben. Die Verbindungslinie wird automatisch geführt.

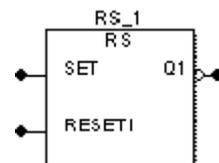
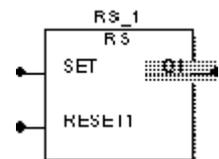
8.8 Negieren von Eingängen und Ausgängen

Eingänge und Ausgänge von Funktionen und Funktionsbausteinen können negiert werden. Sie haben mehrere Möglichkeiten einen FP (Formalparameter) zu negieren. Anhand des Ausgangs 'Q1' des Funktionsbausteins 'RS' sind diese Möglichkeiten hier beschrieben.



Negieren des Formalparameters 'Q1' mit der Maus

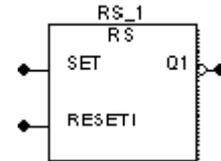
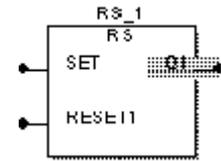
- Klicken Sie auf den Formalparameter 'Q1', um ihn zu markieren. Der markierte Ausgang ändert die Farbe.
- Klicken Sie in der Symbolleiste auf das Symbol 'FP-Negation ein/aus'. Der negierte Ausgang wird durch einen Kreis gekennzeichnet. Um die Negierung wieder aufzuheben, führen Sie einfach die hier beschriebenen Schritte für den gleichen Ausgang erneut durch.





Negieren des Formalparameters 'Q1' mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten den Formalparameter 'Q1'. Der markierte Ausgang ändert die Farbe.
- Drücken Sie <UMSCHALT> + <F5>. Der negierte Ausgang wird durch einen Kreis gekennzeichnet. Um die Negierung wieder aufzuheben, führen Sie einfach die hier beschriebenen Schritte für den gleichen Ausgang erneut durch.



8.9 Duplizieren von Funktionseingängen

Sie können mit dem System Eingänge von erweiterbaren Funktionen duplizieren. Dies ermöglicht Ihnen, so viele Eingänge von erweiterbaren Funktionen zu duplizieren, wie Sie in Ihrer Anwendung benötigen.

HINWEIS



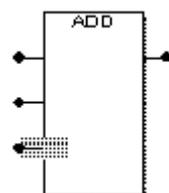
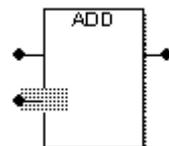
Beachten Sie, dass nur der letzte Eingang einer Funktion dupliziert werden kann.

Im folgenden Beispiel nehmen wir an, dass Sie den Eingang der erweiterbaren Funktion 'ADD' duplizieren wollen.



Duplizieren eines Eingangs der Funktion 'ADD' mit der Maus

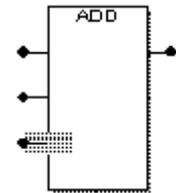
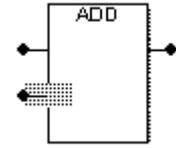
- Klicken Sie auf den letzten Eingang der Funktion. Der markierte Eingang ändert die Farbe.
- Klicken Sie in der Symbolleiste auf das Symbol 'FP duplizieren'. Der neue Eingang wird unterhalb des zuvor markierten Eingangs eingefügt.





Duplizieren eines Eingangs der Funktion 'ADD' mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten den letzten Eingang der Funktion. Der markierte Eingang ändert die Farbe.
- Drücken Sie <STRG> + <F5>.
Der neue Eingang wird unterhalb des zuvor markierten Eingangs eingefügt.



EDITIEREN IN KOP

9.1 Aufrufen des graphischen Editors mit einem KOP-Arbeitsblatt

Zum Editieren eines Code-Arbeitsblattes in KOP müssen Sie zunächst aus dem Projektbaum den graphischen Editor mit dem KOP-Arbeitsblatt aufrufen.



Eine allgemeine Beschreibung des Projektbaums sowie Informationen, wie Sie im Projektbaum durch die POEs blättern, finden Sie in den Kapiteln 'Allgemeines zum Programm' und 'Arbeiten mit Projekten' in diesem Handbuch.

Im folgenden Beispiel nehmen wir an, dass Sie das KOP-Arbeitsblatt eines Funktionsbausteins mit dem Namen 'LD_FB' editieren wollen. Hierzu müssen Sie zunächst einen Funktionsbaustein mit diesem Namen einfügen, wie im Kapitel 'Arbeiten mit Projekten' beschrieben. Beachten Sie dabei, dass die Programmiersprache eines Arbeitsblattes durch die POE-Sprache festgelegt wird. Diese wird beim Einfügen der POE beziehungsweise des Arbeitsblattes definiert.



Aufrufen des graphischen Editors für das KOP-Arbeitsblatt mit der Maus

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den Unterbaum 'LD_FB' mit den POE-Arbeitsblättern durch Doppelklicken auf den POE-Namen.
- Doppelklicken Sie auf das Symbol 'Arbeitsblatt in KOP' des Funktionsbausteins 'LD_FB'.

Es wird der graphische Editor mit dem KOP-Arbeitsblatt aufgerufen.





Aufrufen des graphischen Editors für das KOP-Arbeitsblatt mit der Tastatur

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den entsprechenden Unterbaum mit den POE-Arbeitsblättern wie folgt: Drücken Sie <↓> oder <↑>, um den Funktionsbaustein 'LD_FB' zu markieren. Drücken Sie <→>, um den Unterbaum des Funktionsbausteins zu öffnen.
- Drücken Sie <↓> oder <↑>, um das Symbol 'Arbeitsblatt in KOP' des Funktionsbausteins 'LD_FB' zu markieren.
- Drücken Sie <↵>. Es wird der graphische Editor mit dem KOP-Arbeitsblatt aufgerufen.

9.2 KOP-Netzwerke, Kontakte, Spulen und Stromschienen

Code in der graphischen Sprache KOP besteht aus Spulen und Kontakten. Gemäß IEC 61131-3 können verschiedene Typen von Kontakten und Spulen verwendet werden:

Symbol	Name	Beschreibung
-- --	Schließer	Der boolesche Wert wird von links nach rechts kopiert, wenn der Zustand der zugehörigen Variablen EIN ist.
-- / --	Öffner	Der boolesche Wert wird von links nach rechts kopiert, wenn der Zustand der zugehörigen Variablen AUS ist.
-- () --	Spule	Der boolesche Wert wird von links nach rechts und auf die zugehörige Variable kopiert.
-- (/) --	Negative Spule	Der boolesche Wert wird von links nach rechts kopiert. Der negierte boolesche Wert wird auf die zugehörige Variable kopiert.
-- (S) --	SETZE-Spule	Der boolesche Wert wird von links nach rechts kopiert. Die zugehörige Variable wird gesetzt, wenn die linke Verbindung TRUE ist.
-- (R) --	RÜCKSETZE-Spule	Der boolesche Wert wird von links nach rechts kopiert. Die zugehörige Variable wird rückgesetzt, wenn die linke Verbindung TRUE ist.

Abbildung 94: Kontakte und Spulen in KOP

Kontakte und Spulen sind durch Linien verbunden und links und rechts durch Stromschienen begrenzt. Der Zustand der linken Stromschiene wird immer als EIN betrachtet. Die rechte Stromschiene ist optional.

Zusätzlich zu den seriellen Verbindungen von Kontakten und Spulen können parallele Zweige erstellt werden. Parallele Zweige werden auch verdrahtetes ODER genannt.

Die verbundenen Objekte zusammen bilden ein KOP-Netzwerk. Jedes KOP-Netzwerk muss mindestens einen Kontakt, eine Spule und eine linke Stromschiene enthalten.

Variablen, die mit Kontakten und Spulen in KOP verwendet werden, sind immer boolesche Variablen. Beim Einfügen von Kontakten oder Spulen kann der Variablenname eingegeben werden. Der Variablenname wird über dem Kontakt oder der Spule im Arbeitsblatt angezeigt.

In KOP-Codes können Kommentare unter Verwendung des Kontextmenüs Menüpunkt 'Text (Anmerkung)...' eingefügt werden.

Das folgende Beispiel zeigt ein einfaches KOP-Netzwerk:

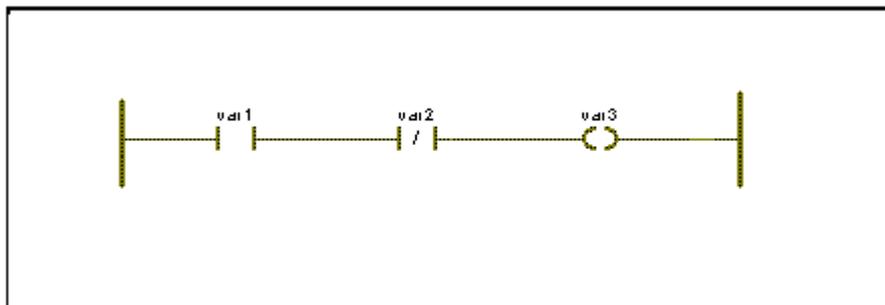


Abbildung 95: Beispiel eines KOP-Netzwerks

Das Beispiel-Netzwerk enthält eine linke und eine rechte Stromschiene, die das KOP-Netzwerk nach links und nach rechts begrenzen. Dazwischen liegen zwei Kontakte und eine Spule, die durch horizontale Linien miteinander verbunden sind. Der erste dieser Kontakte, der als Schließer bezeichnet wird, leitet den Eingangswert von links nach rechts, wenn der Wert der Variablen 'var1' TRUE (wahr) ist. Der zweite Kontakt leitet den Eingangswert weiter, wenn der Wert der Variablen 'var2' FALSE (falsch) ist. Die Spule speichert den Eingangswert in der mit ihr verbundenen Variablen 'var3'.



HINWEIS

Um das Arbeitsblatt kompilieren zu können, müssen Sie die Variablen deklarieren, wie im Kapitel 'Literele, Datentypen und Variablen' beschrieben.

Sie können in KOP-Arbeitsblättern auch FBS-Elemente entweder manuell oder mit Hilfe des Editor-Assistenten einfügen. Dieser Vorgang ist im Abschnitt 'Aufrufen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten' in diesem Kapitel beschrieben.

9.3 Einfügen von Kontakten und Spulen

Für die folgenden Schritte setzen wir voraus, dass Sie bereits einen KOP-Funktionsbaustein mit dem Namen 'LD_FB' eingefügt haben, wie im Kapitel 'Arbeiten mit Projekten' beschrieben. Des Weiteren nehmen wir an, dass Sie zunächst ein KOP-Netzwerk mit vier Kontakten einfügen wollen.



Einfügen eines KOP-Anfangsnetzwerks mit der Maus

- Blenden Sie bei Bedarf das Gitter im Arbeitsbereich ein. Wählen Sie dazu aus dem Untermenü 'Layout' den Menüpunkt 'Gitter'.
- Klicken Sie in das Arbeitsblatt, um eine Einfügemarke zu setzen.
- Klicken Sie in der Symbolleiste auf das Symbol 'Kontaktnetzwerk'. Es wird ein KOP-Anfangsnetzwerk mit einem Kontakt und einer Spule eingefügt.



Einfügen eines KOP-Anfangsnetzwerks mit der Tastatur

- Blenden Sie bei Bedarf das Gitter im Arbeitsbereich ein. Drücken Sie dazu die Tastenkombination <STRG> + <.,>.
- Drücken Sie die <LEERTASTE>, um eine Einfügemarke zu setzen.
- Drücken Sie <F6>. Es wird ein KOP-Anfangsnetzwerk mit einem Kontakt und einer Spule eingefügt.

Ihr Bildschirm müsste nun wie in der folgenden Abbildung aussehen:

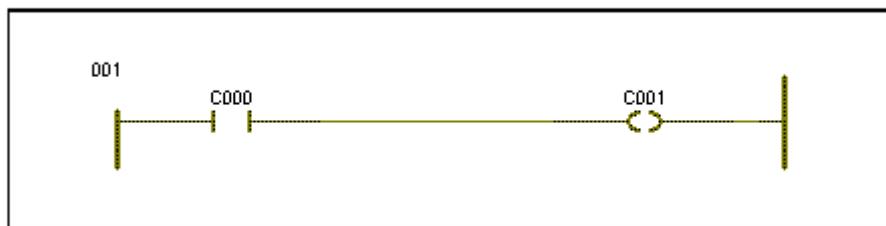


Abbildung 96: KOP-Anfangsnetzwerk

9.4 Einfügen von seriellen Kontakten und Spulen

Weitere serielle Kontakte und Spulen können Sie mit den Symbolen 'Kontakt links einfügen', 'Kontakt rechts einfügen' und 'Spule rechts einfügen' in der Symbolleiste oder den entsprechenden Menüpunkten bzw. den definierten Shortcuts in Ihr KOP-Netzwerk einfügen. Für die folgenden Schritte nehmen wir an, dass Sie einen weiteren seriellen Kontakt in Ihr KOP-Netzwerk einfügen wollen.



Einfügen weiterer serieller Kontakte mit der Maus

- Klicken Sie auf den Kontakt 'C000', um diesen zu markieren.
- Klicken Sie in der Symbolleiste auf das Symbol 'Kontakt rechts einfügen'. Der neue Kontakt 'C002' wird rechts vom markierten Kontakt eingefügt.





Einfügen weiterer serieller Kontakte mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten den Kontakt 'C000'.
- Drücken Sie <F7>.
Der neue Kontakt 'C002' wird rechts vom markierten Kontakt eingefügt.

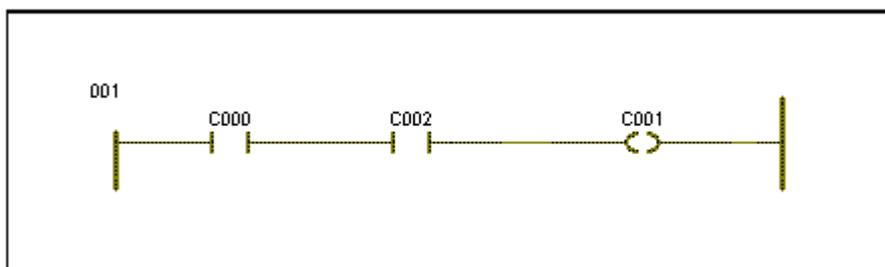


Abbildung 97: KOP-Anfangsnetzwerk mit eingefügtem Kontakt 'C002'

9.5 Einfügen von parallelen Kontakten oder Spulen

Bis jetzt haben wir nur serielle KOP-Netzwerke betrachtet. In KOP können Sie außerdem parallele Zweige erstellen. Diese werden auch verdrahtete ODER genannt. In den nächsten Schritten nehmen wir an, dass Sie am Kontakt 'C002' des KOP-Netzwerks im vorhergehenden Abschnitt einen parallelen Zweig einfügen wollen.



Einfügen eines parallelen Zweigs mit der Maus

- Klicken Sie auf den Kontakt 'C002', um diesen zu markieren.
- Klicken Sie in der Symbolleiste auf das Symbol 'Kontakt/Spule unterhalb Markierung einfügen', um einen parallelen Zweig unterhalb von Kontakt 'C002' einzufügen.



Oder:

- Klicken Sie in der Symbolleiste auf das Symbol 'Kontakt/Spule über Markierung einfügen', um einen parallelen Zweig über Kontakt 'C002' einzufügen.



In beiden Fällen wird der neue Kontakt 'C003' eingefügt.



Einfügen eines parallelen Zweigs mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten den Kontakt 'C002'.
- Drücken Sie <STRG> + <F7>, um einen Kontakt unterhalb von Kontakt 'C002' einzufügen.
Der neue Kontakt 'C003' wird eingefügt.

9.6 Verwenden des Editiermodus für KOP-Zweige

Ihr Bildschirm müsste nun wie in der folgenden Abbildung aussehen:

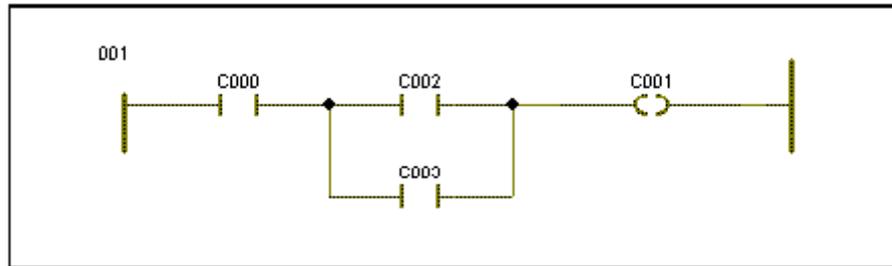


Abbildung 98: KOP-Netzwerk mit einem parallelen Zweig

9.6 Verwenden des Editiermodus für KOP-Zweige

Neue Zweige können Sie im Editiermodus für KOP-Zweige in Ihrem Netzwerk einfügen. Dies ermöglicht Ihnen, komplexe Zweigstrukturen sehr schnell zu erzeugen. Der Editiermodus kann in bestehenden KOP-Netzwerken verwendet werden. Bevor Sie den Editiermodus verwenden können, müssen Sie zuerst ein Netzwerk mit mindestens einem Kontakt und einer Spule erstellt haben.

In der folgenden Beschreibung nehmen wir an, dass Sie einen weiteren parallelen Kontakt einfügen wollen.



Einfügen eines parallelen Zweigs mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'KOP-Zweig einfügen'. Unter dem Cursor erscheint ein Symbol für einen KOP-Zweig. Bei aktiviertem Editiermodus erscheint das Symbol in der Symbolleiste gedrückt.
- Klicken Sie auf die Stelle einer Verbindung zwischen zwei Objekten, an der Sie Ihren parallelen Zweig beginnen wollen (siehe Schritt 1. in der folgenden Abbildung).
- Bewegen Sie die Maus nach oben oder nach unten an eine freie Position (siehe Schritt 2. in der folgenden Abbildung).
- Klicken Sie mit der linken Maustaste auf die Stelle, an der das neue Objekt positioniert werden soll.
- Bewegen Sie die Maus zu dem gewünschten Ende des parallelen Zweigs (siehe Schritt 3. in der folgenden Abbildung).
- Klicken Sie mit der linken Maustaste, um die Verbindung zu erstellen.



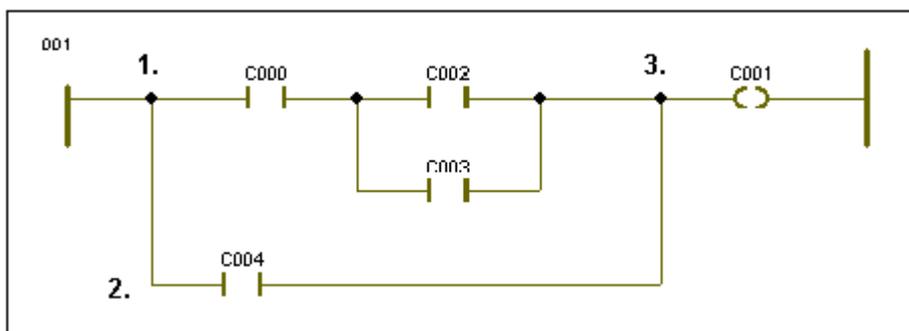


Abbildung 99: KOP-Netzwerk mit einem parallelen Zweig, eingefügt im Editiermodus für KOP-Zweige

9.7 Ändern der Eigenschaften von Kontakten und Spulen

Sie können die Eigenschaften aller Objekte ändern, die Sie in Ihr KOP-Arbeitsblatt eingefügt haben. Vielleicht wollen Sie einen Schließer in einen Öffner oder eine Spule in eine SETZE-Spule umwandeln. Zu diesem Zweck befinden sich mehrere Symbole in der Symbolleiste. Zu jedem Symbol und dessen Funktion bietet Ihnen das System eine Kurzinformation. Diese wird eingeblendet, wenn Sie den Cursor auf ein bestimmtes Symbol setzen. Zusätzlich erscheint in der Statuszeile eine kurze Funktionsbeschreibung des Symbols.

In den nächsten Schritten nehmen wir an, dass Sie einen Kontakt in einen Öffner umwandeln wollen.



Ändern der Eigenschaften eines Kontakts mit der Maus

- Klicken Sie auf den zu ändernden Kontakt, um diesen zu markieren.
- Klicken Sie in der Symbolleiste auf das Symbol 'Öffner'.
Der Typ des Kontakts wird geändert.



Ändern der Eigenschaften eines Kontakts mit der Tastatur

- Markieren Sie mit Hilfe der Cursorstasten einen Kontakt.
- Drücken Sie <UMSCHALT> + <F7>, um die Eigenschaften zu ändern.
Der Typ des Kontakts wird geändert.



HINWEIS

Beim Ändern der Eigenschaften einer Spule, können Sie diese in eine negative Spule, SETZE-Spule und RÜCKSETZE-Spule umwandeln.



Ändern der Eigenschaften eines Kontakts im Dialog 'Kontakt/Spule'

- Doppelklicken Sie auf den Kontakt, der geändert werden soll.
Es erscheint der Dialog 'Kontakt/Spule'.

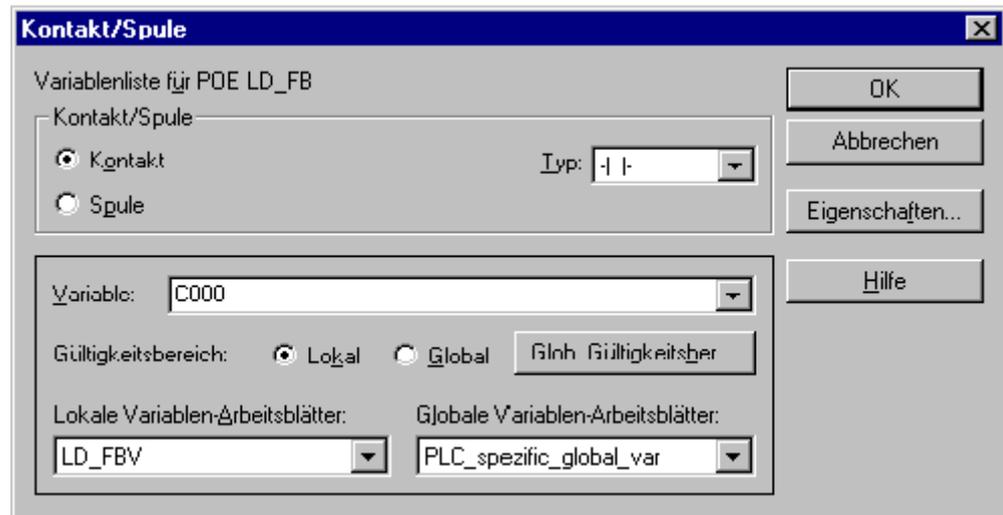


Abbildung 100: Dialog 'Kontakt/Spule'



Verwenden des Dialogs 'Kontakt/Spule'



- Wählen Sie aus dem Listenfeld 'Typ' den benötigten Kontaktyp.
- Geben Sie bei Bedarf einen Namen für die Variable ein.
- Bestätigen Sie den Dialog.
Es erscheint der Dialog 'Automatische Variablen-Deklaration'.

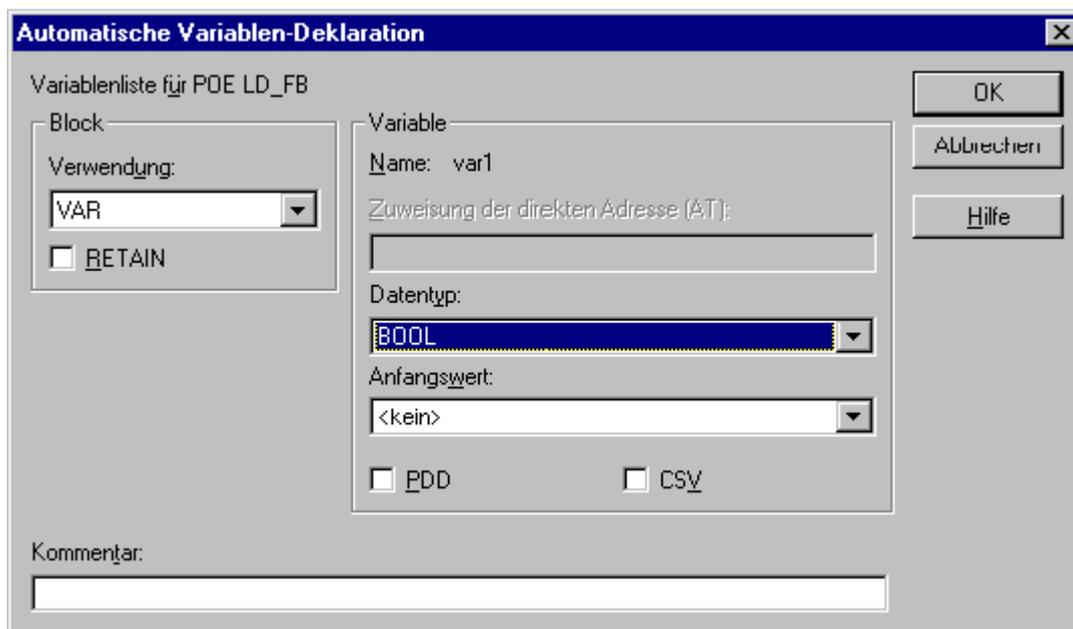


Abbildung 101: Dialog 'Automatische Variablen-Deklaration'

Verwenden des Dialogs 'Automatische Variablen-Deklaration'



- Wählen Sie aus dem Listenfeld 'Verwendung' ein Variablen-Schlüsselwort.
- Geben Sie einen Speicherort im Feld 'Zuweisung der direkten Adresse (AT)' ein, wenn Sie eine adressierte Variable deklarieren wollen (dies ist nur in Programmen oder für globale Variablen möglich).
- Wählen Sie den richtigen Datentyp aus dem Feld 'Datentyp'.
- Geben Sie bei Bedarf einen Anfangswert ein.
- Aktivieren Sie das Kontrollkästchen 'PDD', wenn die Variable im OmegaOS PDD (Process Data Directory) gespeichert werden soll, d. h. in Kommunikations-Funktionsbausteinen der IEC 61131-5 verwendet werden soll.
- Aktivieren Sie das Kontrollkästchen 'CSV', wenn die Variable in der CSV-Datei gespeichert werden soll, d. h. im OPC-Server verwendet werden soll. Der OPC-Server verarbeitet nur Variablen, die in der CSV-Datei eingetragen sind. Diese Variablen können in einem OPC-Client-Prozess verwendet werden (z. B. in einer Visualisierung).
- Geben Sie bei Bedarf einen Kommentar ein.
- Bestätigen Sie den Dialog.
Die neue Variable wird in das Code-Arbeitsblatt eingefügt. Die Deklaration der Variablen wird automatisch in der Variablen-Deklaration der POE vorgenommen.

HINWEIS



Wenn Sie bereits deklarierte Variablen einfügen, erscheint der Name der Variablen im Listenfeld des Dialogs 'Variable'. Beim Bestätigen des Dialogs wird die Variable direkt in das Code-Arbeitsblatt eingefügt. Der Dialog 'Automatische Variablen-Deklaration' erscheint dann nicht.

9.8 Einfügen von Variablen



Detaillierte Informationen zum OPC-Server finden Sie im Handbuch zum OPC-Server.

Wenn Sie alle Schritte des Programmierbeispiels durchgeführt haben, müsste Ihr Bildschirm wie in der folgenden Abbildung aussehen:

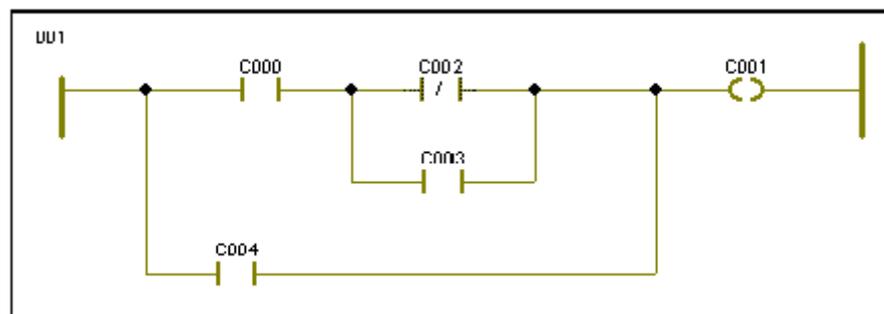


Abbildung 102: KOP-Netzwerk mit geänderten Eigenschaften

9.8 Einfügen von Variablen

Beim Einfügen von Kontakten und Spulen verwendet der graphische Editor einen Platzhalter für den Namen der Variablen. Dieser Platzhalter beginnt mit dem Buchstaben 'C' gefolgt von Ziffern. In den meisten Fällen wollen Sie anstelle dieser Platzhalter wahrscheinlich Ihre eigenen Variablennamen verwenden. Dazu müssen Sie die entsprechenden Schritte durchführen, die im Abschnitt 'Ändern der Eigenschaften von Kontakten und Spulen' in diesem Kapitel beschrieben werden.

Im ersten Schritt muss der Dialog 'Kontakt/Spule' aufgerufen werden, indem Sie auf den Kontakt oder die Spule doppelklicken. In diesem Dialog müssen Sie zunächst festlegen, ob Sie eine lokale oder globale Variable verwenden wollen, indem Sie die entsprechende Option aktivieren. Wenn die Variable bereits deklariert ist, können Sie diese aus dem Listenfeld auswählen. Ist dies nicht der Fall, können Sie einen Namen für die neue Variable eingeben. In diesem Fall erscheint anschließend der Dialog 'Automatische Variablen-Deklaration'. In beiden Fällen wird der Variablenname im KOP-Arbeitsblatt angezeigt, nachdem Sie den Dialog bestätigt haben. Falls erforderlich, wird die Variablen-Deklaration automatisch in das Variablen-Arbeitsblatt der POE eingefügt.

9.9 Aufrufen von Funktionen und Funktionsbausteinen mit dem Editor-Assistenten

In einem KOP-Netzwerk können auch Funktionen und Funktionsbausteine aufgerufen werden.



HINWEIS

Funktionen und Funktionsbausteine werden als FBS-Elemente abgebildet. Dies ermöglicht, KOP- und FBS-Elemente in KOP-Arbeitsblättern zu mischen.

Sie können eine Funktion oder einen Funktionsbaustein an jeder beliebigen Stelle im Arbeitsblatt einfügen und diese anschließend mit dem Netzwerk verbinden oder Funktionen und Funktionsbausteine direkt in ein bestehendes KOP-Netzwerk einfügen.

Mit dem Editor-Assistenten können sehr einfach und bequem FBS-Funktionen oder -Funktionsbausteine in das KOP-Arbeitsblatt eingefügt werden.



Allgemeine Informationen zum Editor-Assistenten finden Sie im Abschnitt 'Der Editor-Assistent' im Kapitel 'Allgemeines zum Programm' in diesem Handbuch.

Wenn der Editor-Assistent nicht geöffnet ist, führen Sie die folgenden Schritte durch:

Aufrufen des Editor-Assistenten mit der Maus



- Klicken Sie in der Symbolleiste auf das Symbol 'Editor-Assistent'. Es erscheint der Editor-Assistent.



Aufrufen des Editor-Assistenten mit der Tastatur



- Drücken Sie <UMSCHALT> + <F2>. Es erscheint der Editor-Assistent.

Nehmen wir im folgenden Beispiel an, dass Sie den Funktionsbaustein 'CTU' in ein vorhandenes KOP-Netzwerk bestehend aus einer linken und rechten Stromschiene, einem Kontakt und einer Spule einfügen wollen.

Einfügen des Funktionsbausteins 'CTU' in ein bestehendes KOP-Netzwerk mit dem Editor-Assistenten



- Klicken Sie im KOP-Netzwerk auf einen Kontakt, um diesen zu markieren.
- Wählen Sie im Editor-Assistenten aus dem Listenfeld 'Gruppe' die Gruppe 'Funktionsbausteine'. Die verfügbaren Funktionsbausteine werden im Auswahlbereich angezeigt.

- Doppelklicken Sie auf den Funktionsbaustein 'CTU'. Der Dialog 'FB-Instanzen' erscheint. Das Feld 'FB-Instanzen' zeigt den Standard-Instanznamen (z. B. ist für einen CTU der Name 'CTU_n' eingetragen, wobei *n* die erste verfügbare Zahl ist, die für die jeweilige Instanz vergeben werden kann).
Sie haben die folgenden Möglichkeiten, um den Instanznamen für den neuen Funktionsbaustein zu definieren:
 - Geben Sie einen neuen Namen im Feld 'FB-Instanzen' ein.
 - Übernehmen Sie den vorgeschlagenen Namen.
 - Wählen Sie aus dem Listenfeld 'FB-Instanzen' einen bereits vorhandenen Namen.
- Drücken Sie 'OK', um den Dialog zu bestätigen. Wenn Sie einen neuen Instanznamen eingegeben haben, erscheint der Dialog 'Automatische FB-Deklaration'.
- Geben Sie bei Bedarf einen Kommentar ein und bestätigen Sie den Dialog mit 'OK'. Der Funktionsbaustein 'CTU' wird automatisch in das KOP-Netzwerk eingefügt.



Einfügen des Funktionsbausteins 'CTU' in ein bestehendes KOP-Netzwerk mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten einen Kontakt im KOP-Netzwerk.
- Drücken Sie <ALT> + <3>, um den Editor-Assistenten zu aktivieren.
- Drücken Sie die <TAB>-Taste um im Editor-Assistenten das Listenfeld 'Gruppe' zu aktivieren. Suchen Sie mit den Cursor-Tasten die Gruppe 'Funktionsbausteine'.
- Drücken Sie die <TAB>-Taste, um den Auswahlbereich zu aktivieren und markieren Sie anschließend mit den Cursor-Tasten den Funktionsbaustein 'CTU'.
- Drücken Sie <↵>. Der Dialog 'FB-Instanzen' erscheint. Das Feld 'FB-Instanzen' zeigt den Standard-Instanznamen (z. B. ist für einen CTU der Name 'CTU_n' eingetragen, wobei *n* die erste verfügbare Zahl ist, die für die jeweilige Instanz vergeben werden kann).
Sie haben die folgenden Möglichkeiten, um den Instanznamen für den neuen Funktionsbaustein zu definieren:
 - Geben Sie einen neuen Namen im Feld 'FB-Instanzen' ein.
 - Übernehmen Sie den vorgeschlagenen Namen.
 - Wählen Sie aus dem Listenfeld 'FB-Instanzen' einen bereits vorhandenen Namen.
- Drücken Sie <↵>, um den Dialog zu bestätigen. Wenn Sie einen neuen Instanznamen eingegeben haben, erscheint der Dialog 'Automatische FB-Deklaration'.
- Geben Sie bei Bedarf einen Kommentar ein und bestätigen Sie den Dialog mit <↵>. Der Funktionsbaustein 'CTU' wird automatisch in das KOP-Netzwerk eingefügt.

Ihr Bildschirm müsste nun wie in der folgenden Abbildung aussehen:

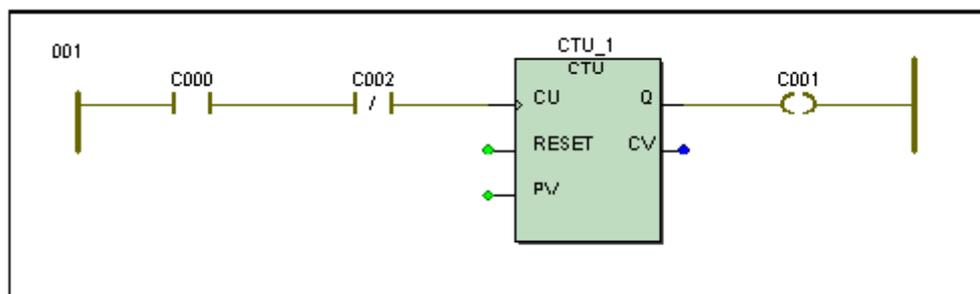


Abbildung 103: KOP-Netzwerk mit Funktionsbaustein 'CTU'

**HINWEIS**

Die nicht belegten Formalparameter des Funktionsbausteins können mit anderen Kontakten, Spulen oder Variablen verbunden werden.



Das Ändern der Eigenschaften von FBS-Funktionen und -Funktionsbausteinen ist im Abschnitt 'Ändern der Eigenschaften von Funktionen und Funktionsbausteinen' im Kapitel 'Editieren in FBS' beschrieben.

10

EDITIEREN IN AS

10.1 Aufrufen des graphischen Editors mit einem AS-Arbeitsblatt

Zum Editieren eines Code-Arbeitsblattes in AS müssen Sie zunächst aus dem Projektbaum den graphischen Editor mit dem AS-Arbeitsblatt aufrufen.



Eine allgemeine Beschreibung des Projektbaums sowie Informationen, wie Sie im Projektbaum durch die POEs blättern, finden Sie in den Kapiteln 'Allgemeines zum Programm' und 'Arbeiten mit Projekten' in diesem Handbuch.

Im folgenden Beispiel nehmen wir an, dass Sie das AS-Arbeitsblatt eines Programms mit dem Namen 'SFC_PROG' editieren wollen. Hierzu müssen Sie zunächst ein Programm mit diesem Namen einfügen, wie im Kapitel 'Arbeiten mit Projekten' beschrieben.



Aufrufen des graphischen Editors für ein AS-Arbeitsblatt mit der Maus

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den Unterbaum 'SFC_PROG' mit den POE-Arbeitsblättern durch Doppelklicken auf den POE-Namen.
- Doppelklicken Sie auf das Symbol 'Arbeitsblatt in AS' des Programms 'SFC_PROG'.
Der graphische Editor mit dem AS-Arbeitsblatt wird aufgerufen.





Aufrufen des graphischen Editors für ein AS-Arbeitsblatt mit der Tastatur

- Wenn das gewünschte Arbeitsblatt-Symbol im Projektbaum nicht sichtbar ist, öffnen Sie den entsprechenden Unterbaum mit den POE-Arbeitsblättern wie folgt: Drücken Sie <↓> oder <↑>, um das Programm 'SFC_PROG' zu markieren. Drücken Sie <→>, um den Unterbaum des Programms zu öffnen.
- Drücken Sie <↓> oder <↑>, um das Symbol 'Arbeitsblatt in AS' des Programms 'SFC_PROG' zu markieren.
- Drücken Sie <↵>.
Der graphische Editor mit dem AS-Arbeitsblatt wird aufgerufen.



10.2 Einführung in AS

Ein AS-Netzwerk besteht aus Schritten und Transitionen, die direkt miteinander verbunden sind.

Einem Schritt können ein oder mehrere Aktionsblöcke zugeordnet sein. Während der Schritt aktiv ist, wird die dazugehörige Aktion gemäß ihrem Qualifier ausgeführt. Die Aktion kann eine boolesche Variable sein. Der auszuführende Code kann auch in einem zusätzlichen Code-Arbeitsblatt editiert werden, das als Lupe bezeichnet wird. In diesem Fall muss der Name des Code-Arbeitsblatts als Name der Aktion verwendet werden. Lupen werden im Arbeitsblatt grün dargestellt. Aktionen, die als einzelne Variablen definiert sind, erscheinen rot.

Eine Transition entspricht der Bedingung, die erfüllt sein muss, um von einem Schritt zum nächsten Schritt überzugehen. Wenn eine Transition TRUE wird, wird der vorangehende Schritt noch einmal ausgeführt und dann wird der nachfolgende Schritt aktiv. Eine Transition kann entweder eine einzelne boolesche Variable sein oder ein direkt verbundener boolescher Ausdruck in FBS oder KOP. Der auszuführende Code einer Transition kann auch in einem zusätzlichen Code-Arbeitsblatt editiert werden, das als Lupe bezeichnet wird. In diesem Fall muss der Name des Code-Arbeitsblatts auch als Transitionsname verwendet werden. Transitionen, die eine direkte Verbindung repräsentieren, werden im Arbeitsblatt wie normale Transitionen mit einem zusätzlichen grünen Verbindungspunkt dargestellt.

Alle verbundenen Objekte zusammen bilden ein AS-Netzwerk. Ein AS-Netzwerk muss immer einen Anfangsschritt haben, der nach einem Kaltstart oder Warmstart als erster ausgeführt wird. Der Anfangsschritt wird im Arbeitsblatt durch ein Rechteck mit doppelter Linie dargestellt. Für alle Schritte wird die Farbe blau verwendet.

In das AS-Netzwerk können simultane oder alternative Zweige eingefügt werden.

In AS-Code-Arbeitsblättern können Kommentare unter Verwendung des Kontextmenüs Menüpunkt 'Text (Anmerkung)...' eingefügt werden. Zusätzlich können mit dem Editor-Assistenten sehr einfach FBS- oder KOP-Elemente in ein AS-Arbeitsblatt eingefügt werden. Eine Beschreibung hierzu finden Sie in den Kapiteln 'Editieren in FBS' und 'Editieren in KOP'.



HINWEIS

Ein AS-Netzwerk sollte aus Zeitgründen nie in Bypass-Tasks oder Event-Tasks verwendet werden! Der Einsatz sollte sich auf die DEFAULT-Task und die CYCLIC-Tasks beschränken.

10.3 Einfügen eines Anfangs-AS-Netzwerkes

In den folgenden Beschreibungen beginnen wir mit einem einfachen AS-Netzwerk. Im ersten Beispiel nehmen wir an, dass Sie ein AS-Netzwerk mit einem Schritt, einer Transition und einem Aktionsblock einfügen wollen.



Einfügen eines AS-Netzwerks mit der Maus

- Blenden Sie bei Bedarf das Gitter im Arbeitsbereich ein. Wählen Sie dazu aus dem Untermenü 'Layout' den Menüpunkt 'Gitter'.
 - Klicken Sie in das Arbeitsblatt, um eine Einfügemarke zu setzen.
 - Klicken Sie in der Symbolleiste auf das Symbol 'Transition-Schritt-Kette einfügen'.
- Es wird ein AS-Netzwerk mit einem Schritt und einer Transition eingefügt.



Einfügen eines AS-Netzwerks mit der Tastatur

- Blenden Sie bei Bedarf das Gitter im Arbeitsbereich ein. Drücken Sie dazu die Tastenkombination <STRG> + <.>.
 - Drücken Sie die <LEERTASTE>, um eine Einfügemarke zu setzen.
 - Drücken Sie <F8>.
- Es wird ein AS-Netzwerk mit einem Schritt und einer Transition eingefügt.

Ihr Arbeitsblatt müsste nun wie in der folgenden Abbildung aussehen:

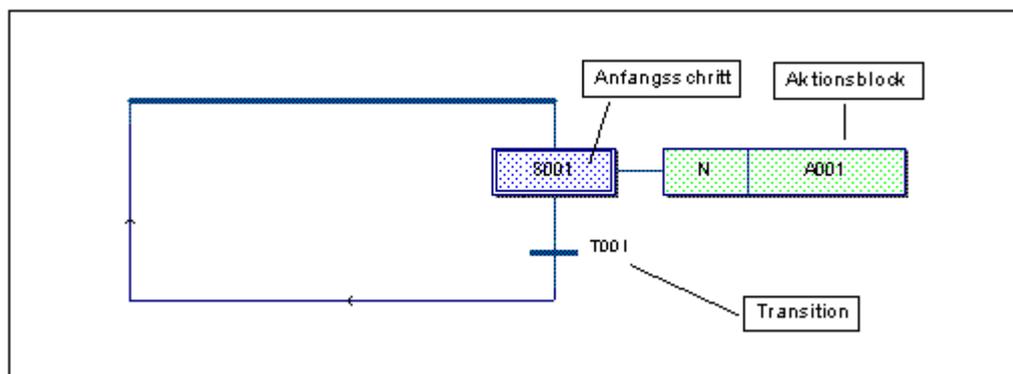


Abbildung 104: AS-Netzwerk mit einem Schritt und einer Transition

Das Beispiel zeigt ein AS-Netzwerk mit einem Anfangsschritt (S001), dem dazugehörigen Aktionsblock (A001) und einer Transition (T001).

10.4 Einfügen weiterer Schritte und Transitionen

Nehmen wir an, dass Sie in Ihr AS-Netzwerk drei weitere Schritte und Transitionen einfügen möchten.



HINWEIS

Schritte und Transitionen werden immer als Paare eingefügt, um ein gültiges Netzwerk aufrechtzuerhalten.



Einfügen weiterer Schritte und Transitionen mit der Maus

- Klicken Sie auf den Schritt, nach dem eine neue Transition und ein neuer Schritt eingefügt werden sollen.
- Klicken Sie in der Symbolleiste auf das Symbol 'Transition-Schritt-Kette einfügen'. Es wird eine Transition-Schritt-Kombination zwischen dem markierten Schritt und der nächsten Transition eingefügt.
- Wiederholen Sie diese Schritte noch zweimal, um zwei weitere Paare einzufügen.



HINWEIS

Wenn Sie anstatt eines Schritts eine **Transition** markieren und anschließend eine Schritt-Transition-Kombination einfügen, wird der neue Schritt und die neue Transition nach der markierten Transition im Netzwerk eingefügt.



Einfügen weiterer Schritte und Transitionen mit der Tastatur

- Markieren Sie mit den Cursortasten den Schritt, nach dem eine neue Transition und ein neuer Schritt eingefügt werden sollen.
- Drücken Sie <F8>. Es wird eine Transition-Schritt-Kombination nach dem markierten Schritt eingefügt.
- Wiederholen Sie diese Schritte noch zweimal, um zwei weitere Paare einzufügen.



HINWEIS

Wenn Sie anstatt eines Schritts eine **Transition** markieren und anschließend eine Schritt-Transition-Kombination einfügen, wird der neue Schritt und die neue Transition nach der markierten Transition im Netzwerk eingefügt.

Ihr Bildschirm müsste nun wie in der folgenden Abbildung aussehen:



HINWEIS

Im folgenden Netzwerk wurden drei neue Schritt-Transition-Kombinationen eingefügt, indem zunächst der Schritt S001 markiert wurde und anschließend dreimal das entsprechende Symbol oder der Shortcut aktiviert wurden. Die neuen Kombinationen wurden **zwischen** Schritt S001 und Transition T001 eingefügt.

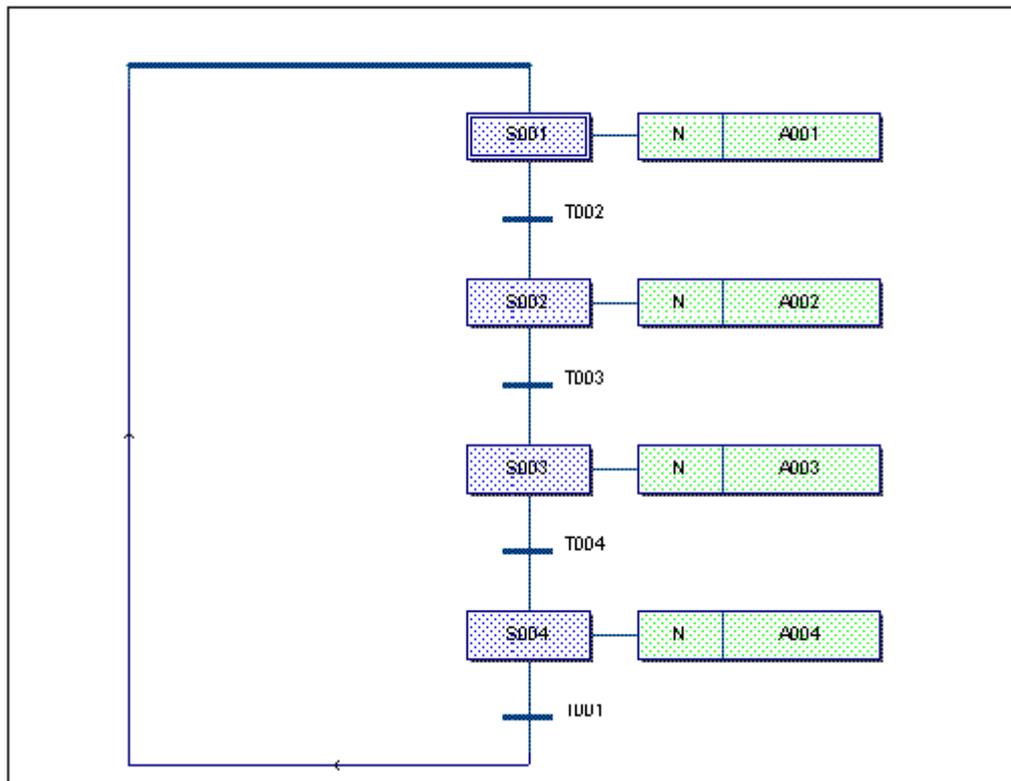


Abbildung 105:AS-Netzwerk mit vier Schritten

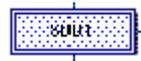
10.5 Umwandeln eines Anfangsschritts in einen normalen Schritt und umgekehrt

Der erste im AS-Netzwerk eingefügte Schritt ist ein Anfangsschritt. In den folgenden Abschnitten nehmen wir an, dass der zweite Schritt zum Anfangsschritt werden soll. Dazu müssen Sie den ersten Schritt in einen normalen Schritt umwandeln und den zweiten Schritt in einen Anfangsschritt. Im folgenden Abschnitt wird beschrieben, wie der Anfangsschritt in einen normalen Schritt geändert werden kann.



Umwandeln eines Anfangsschritts in einen normalen Schritt mit der Maus

- Doppelklicken Sie auf den Anfangsschritt.
Es erscheint der Dialog 'Schritt'.



Umwandeln eines Anfangsschritts in einen normalen Schritt mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten den Anfangsschritt.
- Drücken Sie <↵>.
Es erscheint der Dialog 'Schritt'.



Abbildung 106: Dialog 'Schritt'



Verwenden des Dialogs 'Schritt'

- Deaktivieren Sie das Kontrollkästchen 'Anfangsschritt'.
- 
- Bestätigen Sie den Dialog.

10.6 Einfügen von alternativen Zweigen

Alternative Zweige bedeuten, dass entweder die eine oder die andere Transition wahr wird und nur ein Teil des entsprechenden Zweigs ausgeführt wird. In der folgenden Beschreibung nehmen wir an, dass Sie einen alternativen Zweig nach dem Schritt S002 einfügen möchten.



Einfügen von alternativen Zweigen mit der Maus

- Klicken Sie auf den Schritt S002, um diesen zu markieren.
 - Klicken Sie in der Symbolleiste auf das Symbol 'Simultan-/Alternativverzweigung einfügen'.
- Es erscheint der Dialog 'Verzweigung'.



Einfügen von alternativen Zweigen mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten den Schritt S002.
 - Drücken Sie <STRG> + <F8>.
- Es erscheint der Dialog 'Verzweigung'.

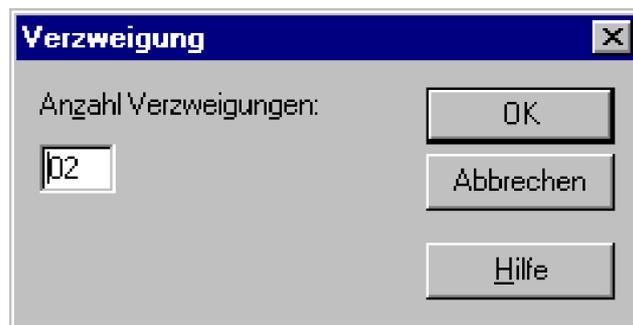


Abbildung 107: Dialog 'Verzweigung'



Verwenden des Dialogs 'Verzweigung'

- Der Wert im Feld 'Anzahl Verzweigungen' gibt an, wie viele Zweige nach dem markierten Schritt eingefügt werden. Bestätigen Sie für unser Beispiel den eingetragenen Wert.



Ihr Bildschirm müsste nun wie in der folgenden Abbildung aussehen:

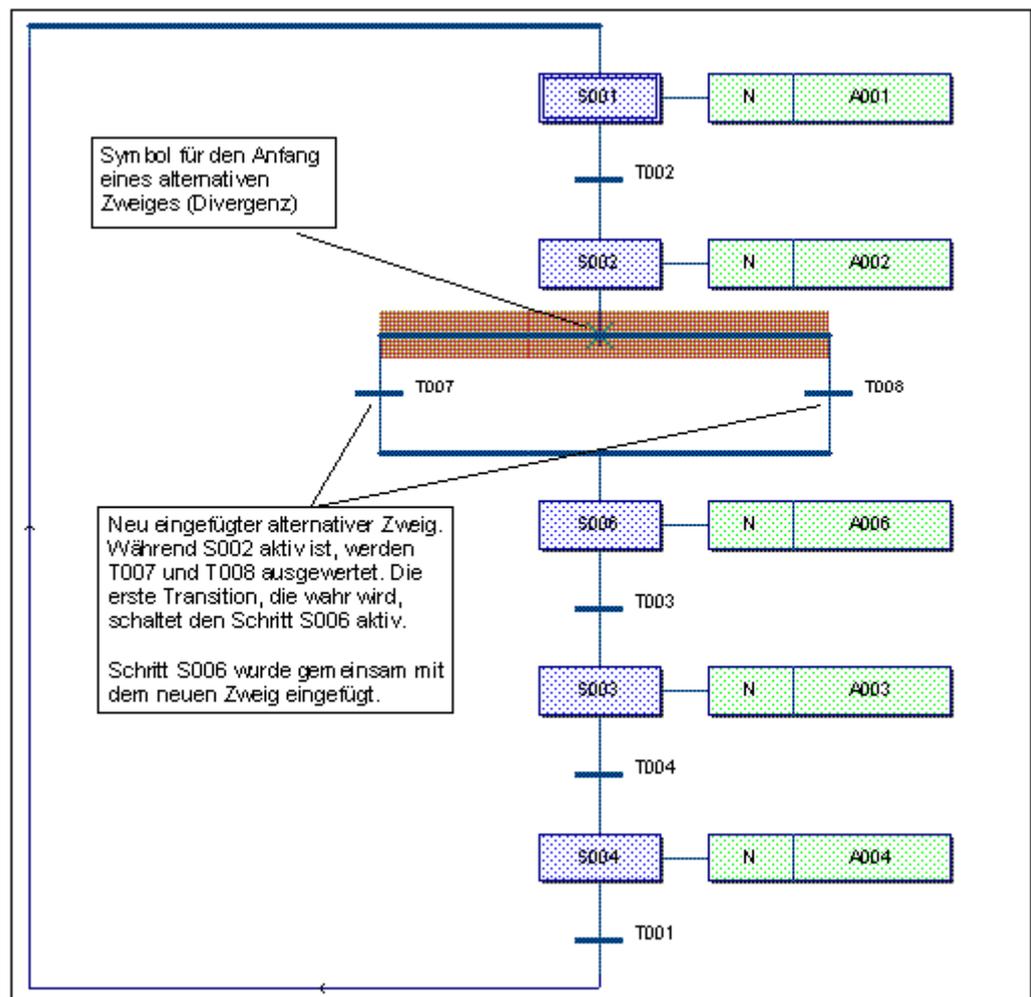


Abbildung 108: AS-Netzwerk mit zwei alternativen Zweigen

10.7 Einfügen von simultanen Zweigen

In simultanen Zweigen werden die Schritte und die dazugehörigen Aktionsblöcke in allen Zweigen gleichzeitig ausgeführt. In der folgenden Beschreibung nehmen wir an, dass Sie einen simultanen Zweig nach der Transition T004 einfügen möchten.



Einfügen von simultanen Zweigen mit der Maus

- Klicken Sie auf die Transition T004, um diese zu markieren.
 - Klicken Sie in der Symbolleiste auf das Symbol 'Simultan-/Alternativverzweigung einfügen'.
- Es erscheint der Dialog 'Verzweigung'.





Einfügen von simultanen Zweigen mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten die Transition T004.
- Drücken Sie <STRG> + <F8>.
Es erscheint der Dialog 'Verzweigung'.

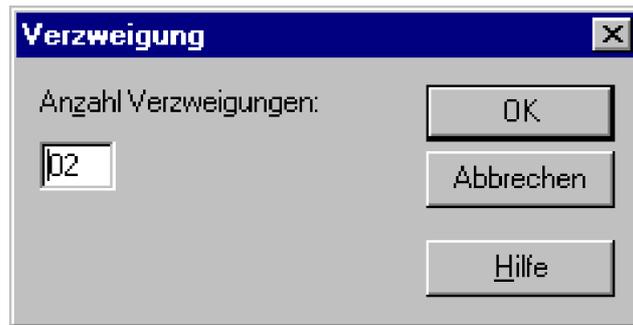


Abbildung 109: Dialog 'Verzweigung'



Verwenden des Dialogs 'Verzweigung'

- Der Wert im Feld 'Anzahl Verzweigungen' gibt an, wie viele Zweige nach der markierten Transition eingefügt werden. Bestätigen Sie für unser Beispiel den eingetragenen Wert.



10.8 Verwenden des Editiermodus für AS-Zweige

Ihr Bildschirm müsste nun wie in der folgenden Abbildung aussehen:

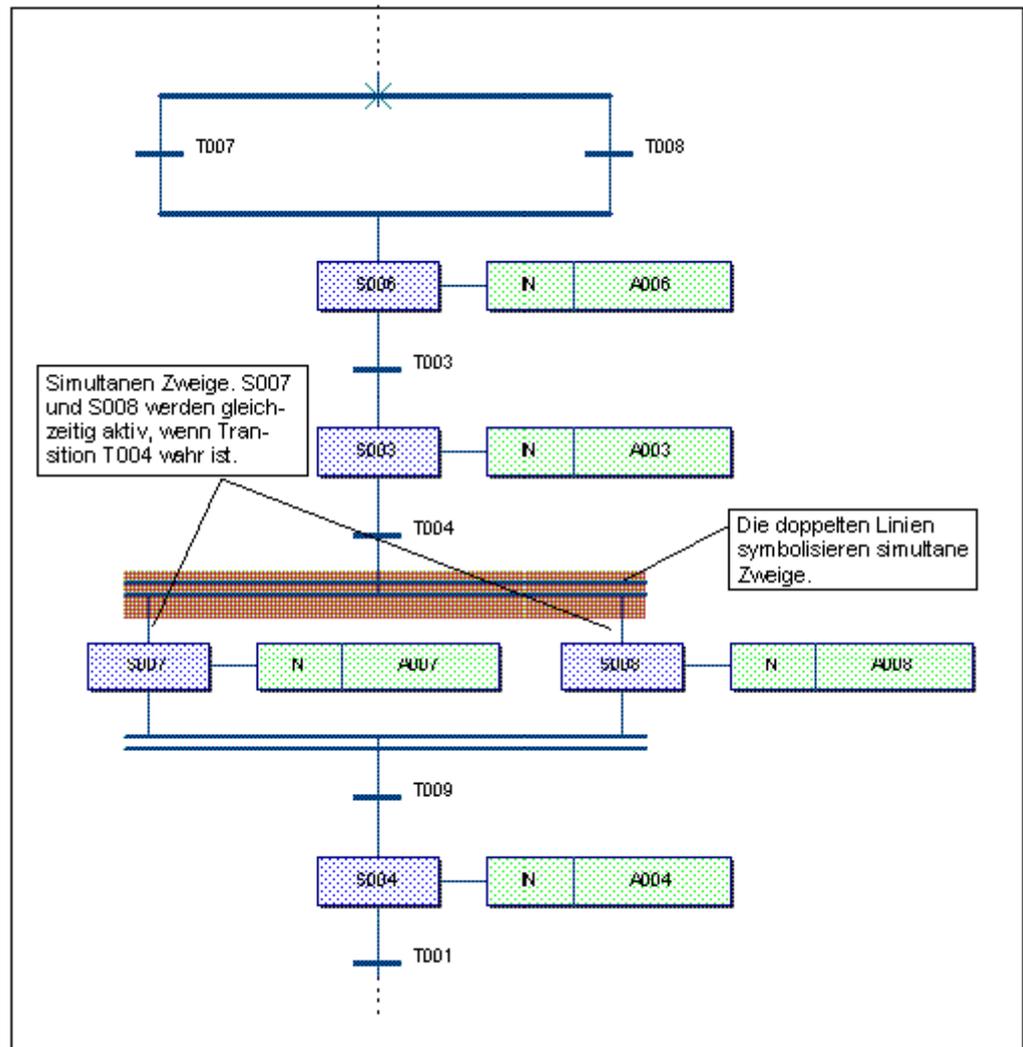


Abbildung 110: Teil eines AS-Netzwerks mit einem alternativen und einem simultanen Zweig

10.8 Verwenden des Editiermodus für AS-Zweige

Sie können Zweige mit dem Editiermodus für AS-Zweige einfügen. Der Editiermodus für AS-Zweige ist ein bequem zu bedienendes Werkzeug, um Simultan- oder Alternativzweige in das Netzwerk einzufügen.



HINWEIS

Wenn Sie den Editiermodus für AS-Zweige verwenden, klicken Sie immer auf das Objekt und nicht auf die Linie.



Einfügen eines alternativen Zweigs mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'AS-Zweig einfügen'. Unter dem Cursor erscheint ein Symbol für einen AS-Zweig. Bei aktiviertem Editiermodus erscheint das Symbol in der Symbolleiste gedrückt.
- Klicken Sie mit der linken Maustaste in die **untere** Hälfte eines Schritts, um einen alternativen Zweig einzufügen.
- Bewegen Sie den Cursor an eine freie Stelle, an der die neuen Objekte platziert werden sollen. Vermeiden Sie dabei Kollisionen mit anderen Objekten. Klicken Sie mit der linken Maustaste, um die Objekte zu platzieren.
- Bewegen Sie den Cursor zu dem Schritt, an dem der Zweig geschlossen werden soll. Klicken Sie an dieser Stelle mit der linken Maustaste. Der Editiermodus wird automatisch deaktiviert, nachdem der neue Zweig eingefügt wurde. Wenn Sie einen weiteren Zweig einfügen möchten, klicken Sie erneut auf das Symbol, um den Editiermodus zu aktivieren.



Einfügen eines simultanen Zweigs mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'AS-Zweig einfügen'. Unter dem Cursor erscheint ein Symbol für einen AS-Zweig. Bei aktiviertem Editiermodus erscheint das Symbol in der Symbolleiste gedrückt.
- Klicken Sie mit der linken Maustaste in die **obere** Hälfte eines Schritts, um einen simultanen Zweig einzufügen.
- Bewegen Sie den Cursor an eine freie Stelle, an der die neuen Objekte platziert werden sollen. Vermeiden Sie dabei Kollisionen mit anderen Objekten. Klicken Sie mit der linken Maustaste, um die Objekte zu platzieren.
- Bewegen Sie den Cursor zu der Transition, an der der Zweig geschlossen werden soll. Klicken Sie an dieser Stelle mit der linken Maustaste. Der Editiermodus wird automatisch deaktiviert, nachdem der neue Zweig eingefügt wurde. Wenn Sie einen weiteren Zweig einfügen möchten, klicken Sie erneut auf das Symbol, um den Editiermodus zu aktivieren.



10.9 Einfügen von Variablen als Aktionen

In AS können boolesche Variablen mit Aktionen verbunden werden. Der Variablenwert wird durch den Qualifier der Aktion bestimmt.



HINWEIS

Detaillierte Informationen über die verfügbaren Qualifier für Aktionen finden Sie in Ihrer SPS-Dokumentation oder in der kontextsensitiven Hilfe.



Verbinden von Variablen mit Aktionsblöcken mit der Maus

- Klicken Sie mit der rechten Maustaste auf den Aktionsblock 'A004', um das Kontextmenü zu öffnen.
- Wählen Sie den Menüpunkt 'Objekteigenschaften...'.
Es erscheint der Dialog 'Aktion'.



Verbinden von Variablen mit Aktionsblöcken mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten den Aktionsblock 'A004'.
- Drücken Sie <ALT> + <↵>.
Es erscheint der Dialog 'Aktion'.

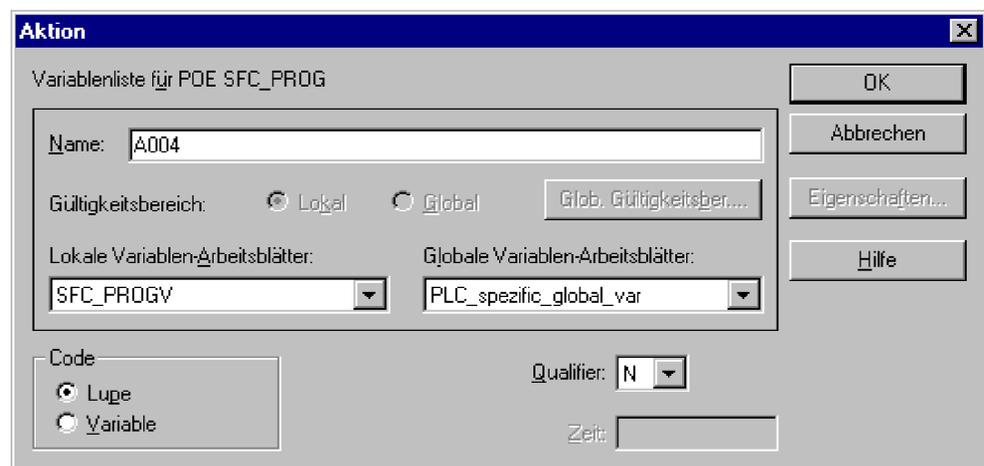


Abbildung 111: Dialog 'Aktion'



Verwenden des Dialogs 'Aktion'

- Geben Sie in das Feld 'Name' den Namen der Variablen ein. Löschen oder überschreiben Sie dabei den aktuellen Eintrag (in unserem Beispiel 'A004').
- Aktivieren Sie die Option 'Variable'.
- Bestätigen Sie den Dialog.
Es erscheint der Dialog 'Automatische Variablen-Deklaration'.



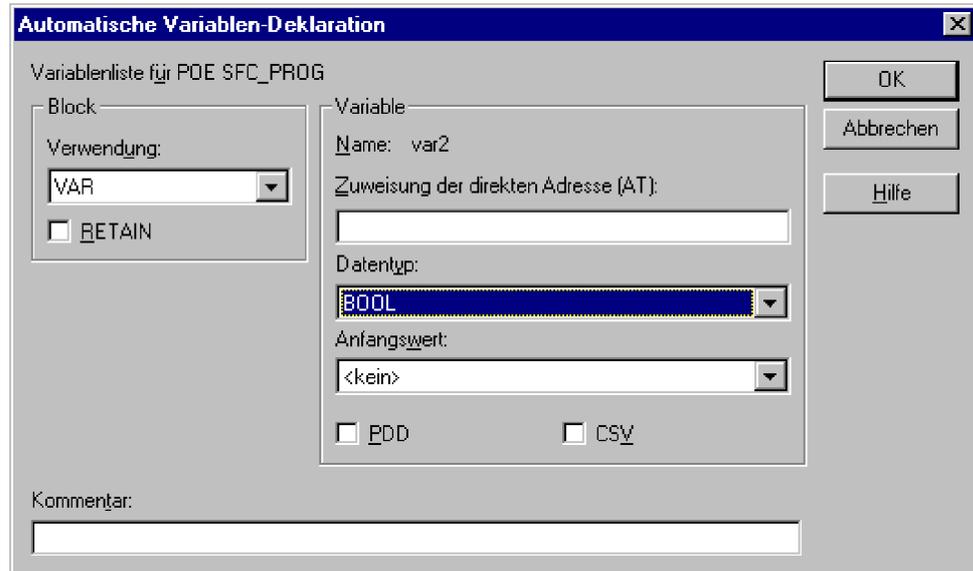


Abbildung 112: Dialog 'Automatische Variablen-Deklaration'

Verwenden des Dialogs 'Automatische Variablen-Deklaration'



- Wählen Sie aus dem Listenfeld 'Verwendung' ein Variablen-Schlüsselwort.
- Geben Sie einen Speicherort im Feld 'Zuweisung der direkten Adresse (AT)' ein, wenn Sie eine adressierte Variable deklarieren wollen (dies ist nur in Programmen oder für globale Variablen möglich).
- Wählen Sie den richtigen Datentyp aus dem Feld 'Datentyp'.
- Geben Sie bei Bedarf einen Anfangswert ein.
- Aktivieren Sie das Kontrollkästchen 'PDD', wenn die Variable im OmegaOS PDD (Process Data Directory) gespeichert werden soll, d. h. in Kommunikations-Funktionsbausteinen der IEC 61131-5 verwendet werden soll.
- Aktivieren Sie das Kontrollkästchen 'CSV', wenn die Variable in der CSV-Datei gespeichert werden soll, d. h. im OPC-Server verwendet werden soll. Der OPC-Server verarbeitet nur Variablen, die in der CSV-Datei eingetragen sind. Diese Variablen können in einem OPC-Client-Prozess verwendet werden (z. B. in einer Visualisierung).
- Geben Sie bei Bedarf einen Kommentar ein.
- Bestätigen Sie den Dialog.
Die neue Variable wird in das Code-Arbeitsblatt eingefügt. Die Deklaration der Variablen wird automatisch in der Variablen-Deklaration der POE vorgenommen.

HINWEIS



Wenn Sie bereits deklarierte Variablen einfügen, erscheint der Name der Variablen im Listenfeld des Dialogs 'Aktion'. Beim Bestätigen des Dialogs wird die Variable direkt in das Code-Arbeitsblatt eingefügt. Der Dialog 'Automatische Variablen-Deklaration' erscheint dann nicht.



Detaillierte Informationen zum OPC-Server finden Sie im Handbuch zum OPC-Server.

In der folgenden Abbildung ist der Aktionsblock mit dem Variablennamen 'var2' dargestellt:

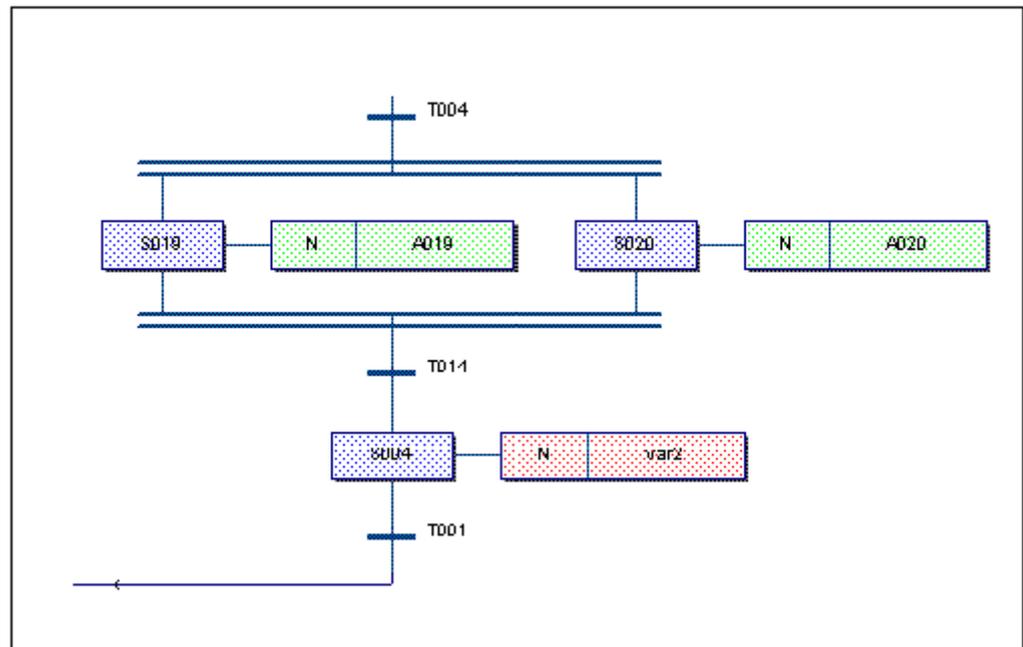


Abbildung 113: Aktionsblock mit Variablennamen

10.10 Einfügen von Variablen als Transitionen

Zum Verbinden einer Variablen mit einer Transition müssen zwei wesentliche Schritte durchgeführt werden. Zunächst müssen die Eigenschaften der Transition in eine direkte Verbindung umgewandelt und dann die Variable eingefügt und verbunden werden. In den folgenden Schritten nehmen wir an, dass Sie eine Variable mit dem Namen 'var1' mit Transition T003 verbinden möchten.



Ändern der Eigenschaften der Transition mit der Maus

- Klicken Sie mit der rechten Maustaste auf die Transition 'T003', um das Kontextmenü zu öffnen.
- Wählen Sie den Menüpunkt 'Objekteigenschaften...'. Es erscheint der Dialog 'Transition'.



Ändern der Eigenschaften der Transition mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten die Transition 'T003'.
- Drücken Sie <ALT> + <↓>.
Es erscheint der Dialog 'Transition'.

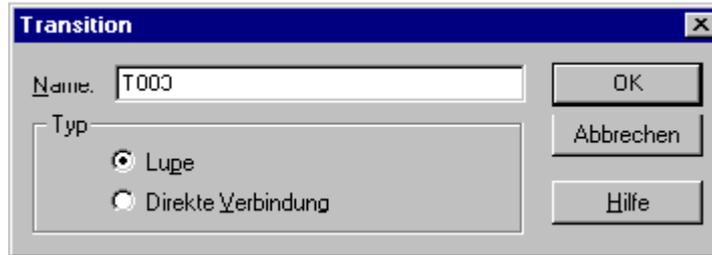


Abbildung 114: Dialog 'Transition'



Verwenden des Dialogs 'Transition'



- Aktivieren Sie die Option 'Direkte Verbindung'.
- Bestätigen Sie den Dialog.
An der Transition erscheint ein blauer Verbindungspunkt.

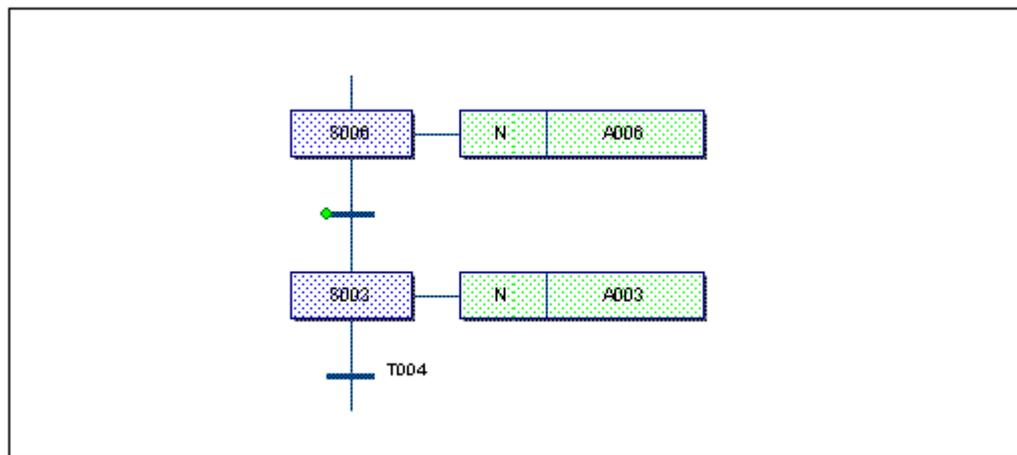


Abbildung 115: Transition als direkte Verbindung spezifiziert (mit blauem Verbindungspunkt)



Einfügen der Variablen mit der Maus

- Klicken Sie auf die Transition, um diese zu markieren.
- Klicken Sie in der Symbolleiste auf das Symbol 'Variable'.
Es erscheint der Dialog 'Variable'.



Einfügen der Variablen mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten die Transition.
- Drücken Sie <F5>.
Es erscheint der Dialog 'Variable'.



Abbildung 116: Dialog 'Variable'



Verwenden des Dialogs 'Variable'

- Geben Sie einen Variablennamen ein.
- Bestätigen Sie den Dialog.
Es erscheint der Dialog 'Automatische Variablen-Deklaration'.



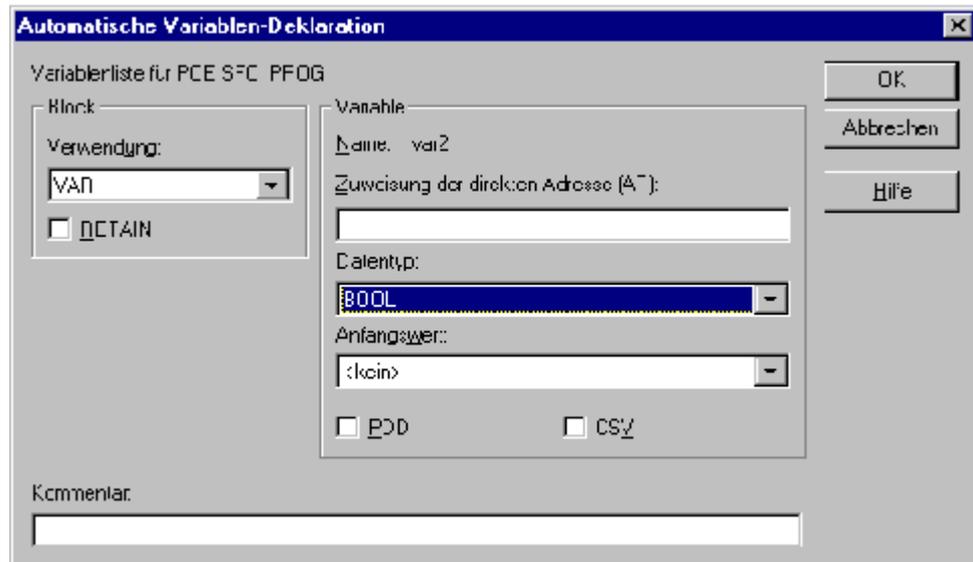


Abbildung 117: Dialog 'Automatische Variablen-Deklaration'

Verwenden des Dialogs 'Automatische Variablen-Deklaration'



- Wählen Sie aus dem Listenfeld 'Verwendung' ein Variablen-Schlüsselwort.
- Geben Sie einen Speicherort im Feld 'Zuweisung der direkten Adresse (AT)' ein, wenn Sie eine adressierte Variable deklarieren wollen (dies ist nur in Programmen oder für globale Variablen möglich).
- Wählen Sie den richtigen Datentyp aus dem Feld 'Datentyp'.
- Geben Sie bei Bedarf einen Anfangswert ein.
- Aktivieren Sie das Kontrollkästchen 'PDD', wenn die Variable im OmegaOS PDD (Process Data Directory) gespeichert werden soll.
- Aktivieren Sie das Kontrollkästchen 'CSV', wenn die Variable in der CSV-Datei gespeichert werden soll, d. h. im OPC-Server verwendet werden soll. Der OPC-Server verarbeitet nur Variablen, die in der CSV-Datei eingetragen sind. Diese Variablen können in einem OPC-Client-Prozess verwendet werden (z. B. in einer Visualisierung).
- Geben Sie bei Bedarf einen Kommentar ein.
- Bestätigen Sie den Dialog.
Die neue Variable wird in das Code-Arbeitsblatt eingefügt. Die Deklaration der Variablen wird automatisch in der Variablen-Deklaration der POE vorgenommen.

HINWEIS



Wenn Sie bereits deklarierte Variablen einfügen, erscheint der Name der Variablen im Listenfeld des Dialogs 'Variable'. Beim Bestätigen des Dialogs wird die Variable direkt in das Code-Arbeitsblatt eingefügt. Der Dialog 'Automatische Variablen-Deklaration' erscheint dann nicht.



Detaillierte Informationen zum OPC-Server finden Sie im Handbuch zum OPC-Server.

Die folgende Abbildung zeigt eine Transition, die mit einer Variablen verbunden ist.

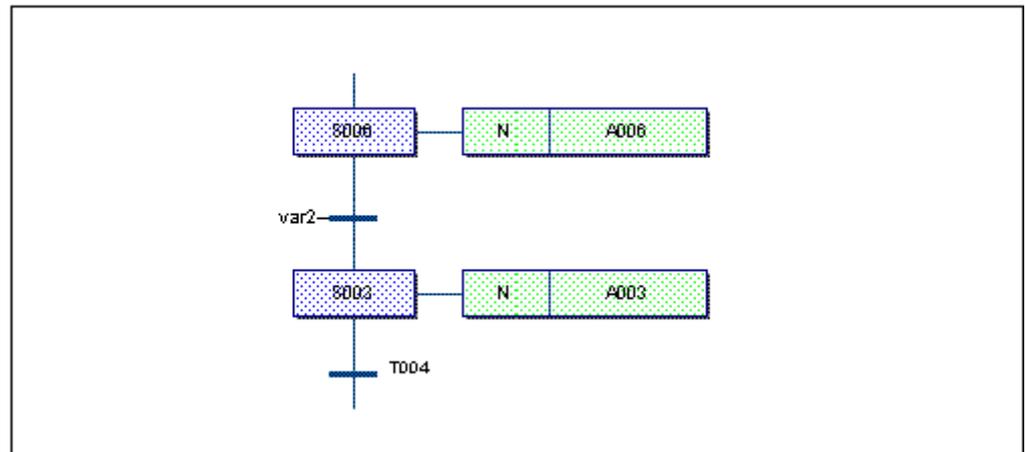


Abbildung 118: Variable verbunden mit einer Transition

10.11 Aufrufen von Funktionen

Anstelle von Variablen, wie im letzten Abschnitt beschrieben, können Sie auch Funktionen mit Transitionen verbinden, die als direkte Verbindungen spezifiziert sind. Dazu müssen Sie zuerst die Funktion einfügen und anschließend mit dem Verbindungspunkt der Transition verbinden.



Eine detaillierte Beschreibung zum Einfügen und Verbinden von Funktionen mit dem Editor-Assistenten finden Sie im Abschnitt 'Einfügen von Funktionen und Funktionsbausteine mit dem Editor-Assistenten' im Kapitel 'Editieren in FBS' in diesem Handbuch.

Sie können auch KOP-Netzwerke einfügen und mit einer Transition verbinden.



Die Schritte, wie Kontakte und Spulen eingefügt werden, werden im Kapitel 'Editieren in KOP' in diesem Handbuch beschrieben.

10.12 Lupe von Aktionen und Transitionen

Gemäß IEC 61131-3 kann für Aktionen und Transitionen eigener Code entwickelt werden, anstatt Variablen mit ihnen zu verbinden. Diese Codes werden in eigenen Arbeitsblättern editiert, die sich im Projektbaum innerhalb der POE unter den Verzeichnisknoten für Aktionen und Transitionen befinden. In den folgenden Schritten nehmen wir an, dass Sie Aktion A005 als Lupe programmieren möchten.



Erstellen einer Aktionslupe mit der Maus

- Doppelklicken Sie im AS-Netzwerk auf den Aktionsblock 'A005'.
Es erscheint der Dialog 'Einfügen'.



Erstellen einer Aktionslupe mit der Tastatur

- Markieren Sie mit Hilfe der Cursortasten den Aktionsblock 'A005' im AS-Netzwerk.
- Drücken Sie <↵>.
Es erscheint der Dialog 'Einfügen'.

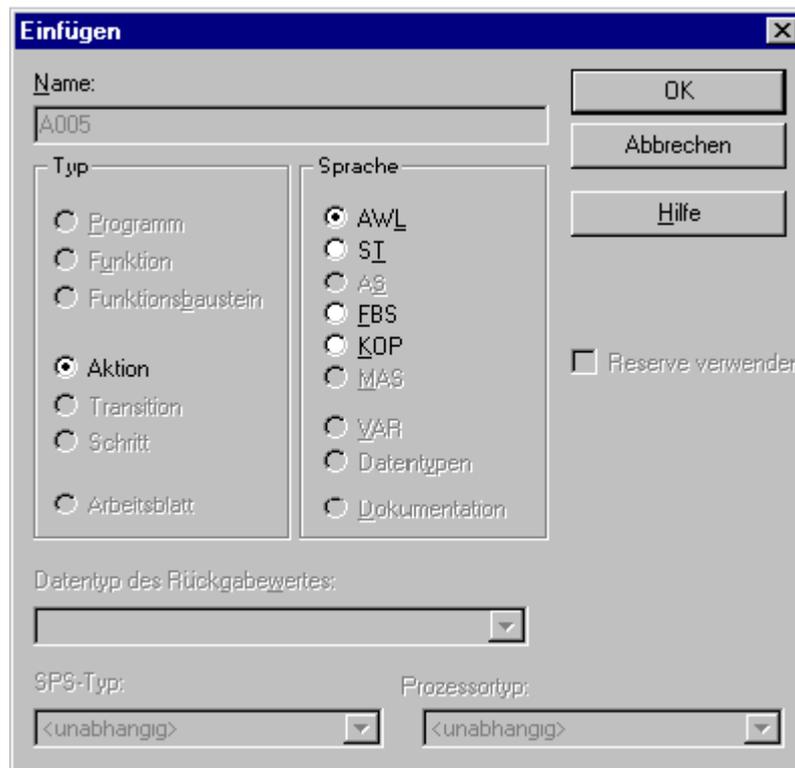


Abbildung 119: Dialog 'Einfügen'



Verwenden des Dialogs 'Einfügen'



- Wählen Sie eine Programmiersprache aus.
- Bestätigen Sie den Dialog.
Das Arbeitsblatt wird in den Projektbaum eingefügt und sofort geöffnet, damit Sie es direkt editieren können.
- Editieren Sie den Code für die Aktionslupe.



HINWEIS

Bei Transitionen gehen Sie gleich vor wie bei Aktionen.

KOMPILIEREN, SENDEN UND DEBUGGEN

11.1 Einfügen von Konfigurationen, Ressourcen und Tasks

Der Projektbaum eines neu erstellten Projektes enthält normalerweise eine Konfiguration, eine Ressource und eine Task. Sie können aber zusätzlich neue Konfigurationen, Ressourcen und Tasks in Ihr bestehendes Projekt einfügen, wobei das Einfügen der einzelnen Elemente immer nach dem selben Schema erfolgt.



HINWEIS

Die Typen der Konfiguration und der Ressource sind von der angeschlossenen SPS abhängig.

Für die folgenden Schritte nehmen wir an, dass Sie in ein bestehendes Projekt mit einer Konfiguration und einer Ressource eine neue Ressource mit dem Namen 'RES_2' für den Prozessortyp IPC einfügen möchten.



Einfügen einer neuen Ressource mit der Maus

- Klicken Sie im Projektbaum mit der rechten Maustaste auf das Symbol '*Ressource-Name*', um das Kontextmenü zu öffnen.
- Wählen Sie den Menüpunkt 'Einfügen'. Es erscheint der Dialog 'Einfügen'.





Einfügen einer neuen Ressource mit der Tastatur

- Drücken Sie <↓> oder <↑>, um das Symbol 'Ressource-Name' im Projektbaum zu markieren.
- Drücken Sie <EINFG>. Es erscheint der Dialog 'Einfügen'.

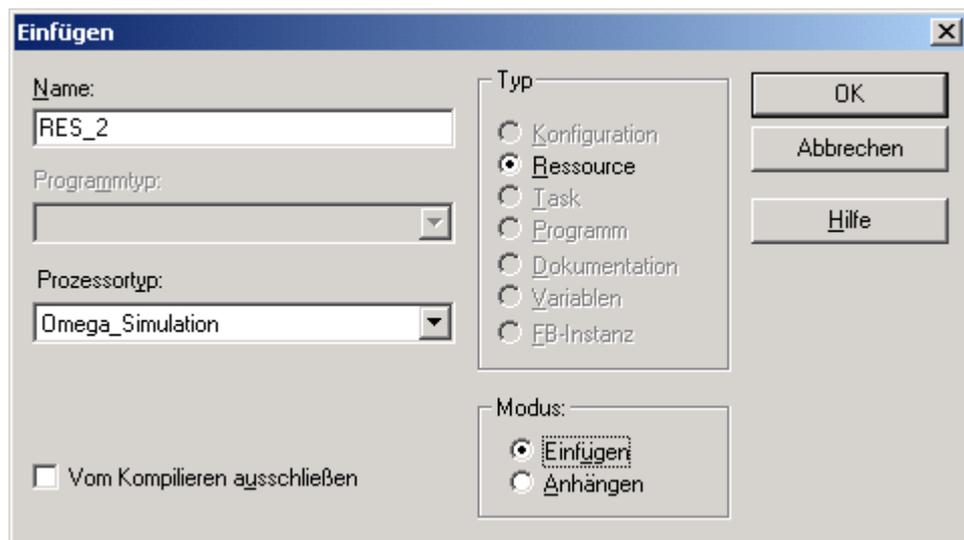
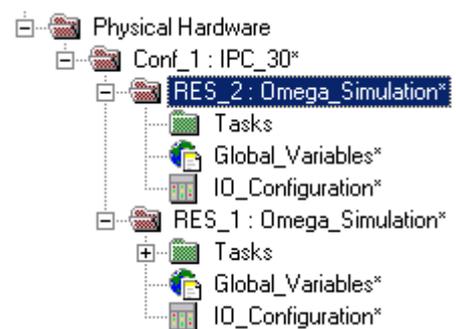


Abbildung 120: Dialog 'Einfügen'



Verwenden des Dialogs 'Einfügen'

- Geben Sie in das Feld 'Name' den Namen der neuen Ressource ein (in unserem Beispiel 'RES_2') und wählen Sie den gewünschten Prozessortyp.
- Bestätigen Sie den Dialog. Die Ressource wird in den Projektbaum eingefügt.
- Wählen Sie nun aus dem Kontextmenü der Ressource (= rechte Maustaste) den Menüpunkt 'Einstellungen...'. Es erscheint der Dialog 'Ressource-Einstellungen für ...'.



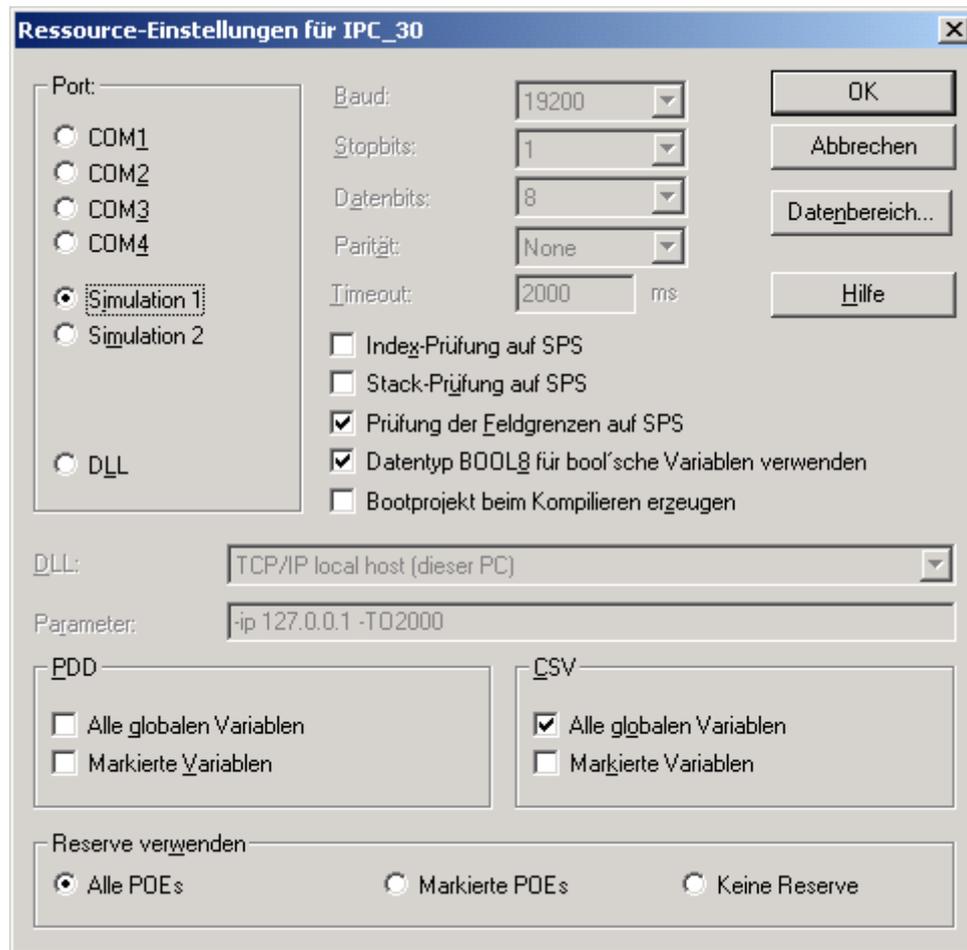


Abbildung 121: Dialog 'Ressource-Einstellungen für ...'



Verwenden des Dialogs 'Ressource-Einstellungen für ...'



- Aktivieren Sie die Option 'Simulation 1', um als Zielsystem für das SPS-Programm beim Senden die Simulation 1 einzustellen.
- Ändern Sie gegebenenfalls die Eigenschaften oder den Datenbereich der Ressource.
- Aktivieren Sie die Kontrollkästchen für die CSV-Einstellungen, wenn Sie die Ressource-Variablen im OPC-Server verwenden möchten. Wenn Sie das Kontrollkästchen 'Alle globalen Variablen' aktivieren, werden alle globalen Variablen in der CSV-Datei gespeichert. Sollen die Variablen, die explizit als CSV-Variablen markiert wurden, in der CSV-Datei gespeichert werden, aktivieren Sie das Kontrollkästchen 'Markierte Variablen'. Der OPC-Server verarbeitet diese Variablen und überträgt deren aktuellen Werte an einen OPC-Client (z. B. an eine Visualisierung).
Detaillierte Informationen zum OPC-Server finden Sie im Handbuch zum OPC-Server.
- Bestätigen Sie den Dialog.

11.1.1 Einfügen der DEFAULT-Task

Die Default-Task wird immer dann ausgeführt, wenn gerade keine andere CYCLIC- oder EVENT-Task ausgeführt werden muss.

Dabei wird die DEFAULT-Task auch immer zyklisch ausgeführt und wechselt sich mit der zyklischen Kommunikationstask, über die die Online-Kommunikation abgewickelt wird und die vom Laufzeitsystem intern automatisch aktiviert wird, ab.

Durch eine zyklische Abarbeitung ist sichergestellt, dass Kommunikationstask und DEFAULT-Task sich nicht gegenseitig beeinflussen und die zeitlichen Zusammenhänge der Programmabarbeitung des IEC-Codes unabhängig davon sind, ob der Anwender gerade Online- oder Offline ist.

Die Zykluszeit der DEFAULT-Task beträgt das doppelte des Systemticks der verwendeten Ressource.

Das Aufrufintervall des Systemticks kann Online im Info-Fenster des jeweiligen Zielsystems abgelesen werden (unter "Online/Ressourcekontrolle/Info" im Eintrag "Systemtick") und ist abhängig von der verwendeten Ressource.

Das Aufrufintervall des Systemticks beträgt z.B. für Drive Line II, b maXX PLC und Controller PLC 10 ms.

Damit wird die DEFAULT-Task alle 20 ms ausgeführt (und die Kommunikationstask ebenfalls alle 20 ms, weil sie sich mit der DEFAULT-Task mit jedem Systemtick abwechselt).

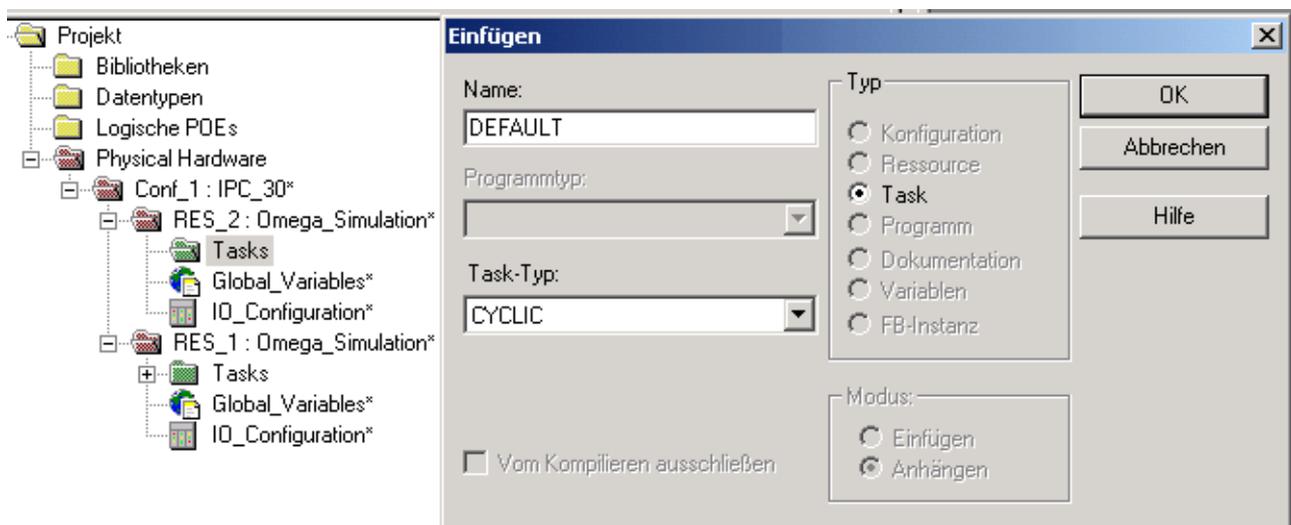


Abbildung 122: Einfügen einer DEFAULT-Task

Nach Bestätigen des Dialogs kann die voreingestellte Watchdogzeit angepasst werden. Dauert die Ausführung der Programmteile, die der DEFAULT-Task zugewiesen wurden, länger als diese Watchdogzeit, geht die Steuerung in Stopp, und eine entsprechende Fehlermeldung der Steuerung kann ausgelesen werden.

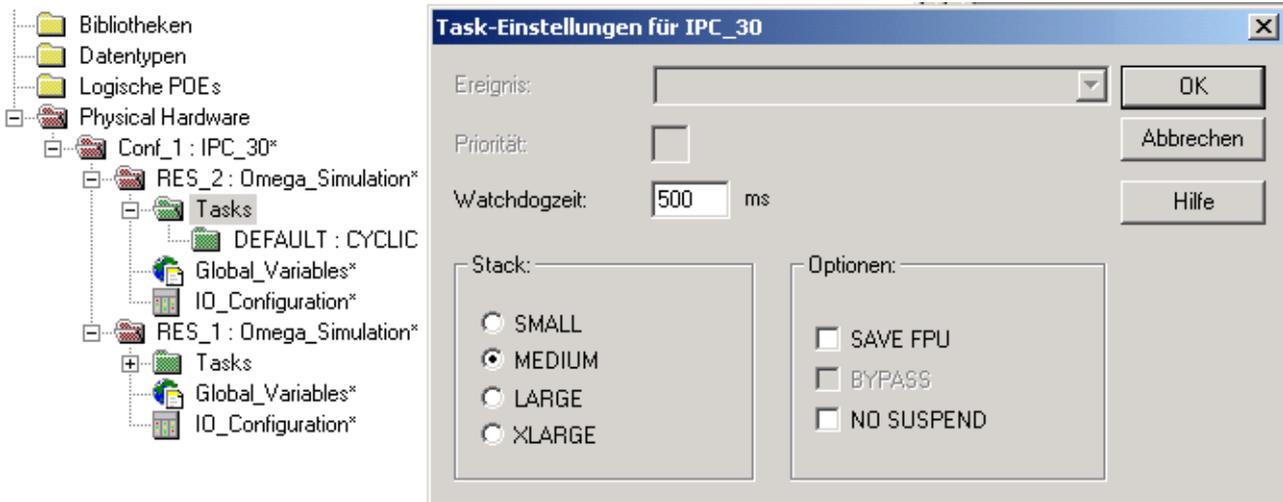


Abbildung 123: Task-Einstellungen für DEFAULT-Task

11.1.2 Einfügen einer neuen Task

Neben der Default-Task können auch andere Tasks angelegt werden.

Dabei werden drei Gruppen unterschieden:

- CYCLIC
- EVENT
- SYSTEM



Abbildung 124: Einfügen einer CYCLIC-Task

Eine Task des Typs "CYCLIC" ist eine zyklische Task, bei der Sie jeweils ein unterschiedliches Aufrufintervall eingeben können (außer der DEFAULT-Task, dessen Aufrufintervall ja bereits durch das verwendete Zielsystem vorgegeben ist). Die Bearbeitung dieser Tasktypen werden vom Laufzeitsystem verwaltet, womit in den POEs, die diesem Task-

typ zugeordnet sind, die Online/Logikanalyse und Online/Adressstatus ausgeführt werden kann.

Eine Task des Typs "EVENT" ist eine BYPASS-Task, die nicht unter Laufzeitkontrolle steht und abhängig von den verwendeten Ressourcen ist. Dazu siehe die entsprechenden Abschnitte in den einzelnen Bedienungs- und Applikationshandbüchern der jeweiligen Steuerung.

Eine Task des Typs "SYSTEM" enthält eine spezielle Systemtask wie z. B. die Kalt- und die Warmstart-Task.

Als Beispiel wollen wir nun eine zyklische Task "TaskNeu" einfügen, die alle 50 ms aufgerufen werden soll.

In [▶Abbildung 124◀](#) auf Seite 179 ist als Task-Typ der Typ "CYCLIC" auszuwählen. Nach Bestätigen des Dialogs erscheint ein Auswahlnenü, in dem die Zykluszeit (=Intervall) und die Überwachungszeit (=Watchdogzeit) eingestellt werden kann.

Auch hier meldet die Steuerung einen Fehler, wenn die Laufzeit der Task die eingestellte Watchdogzeit überschreitet.

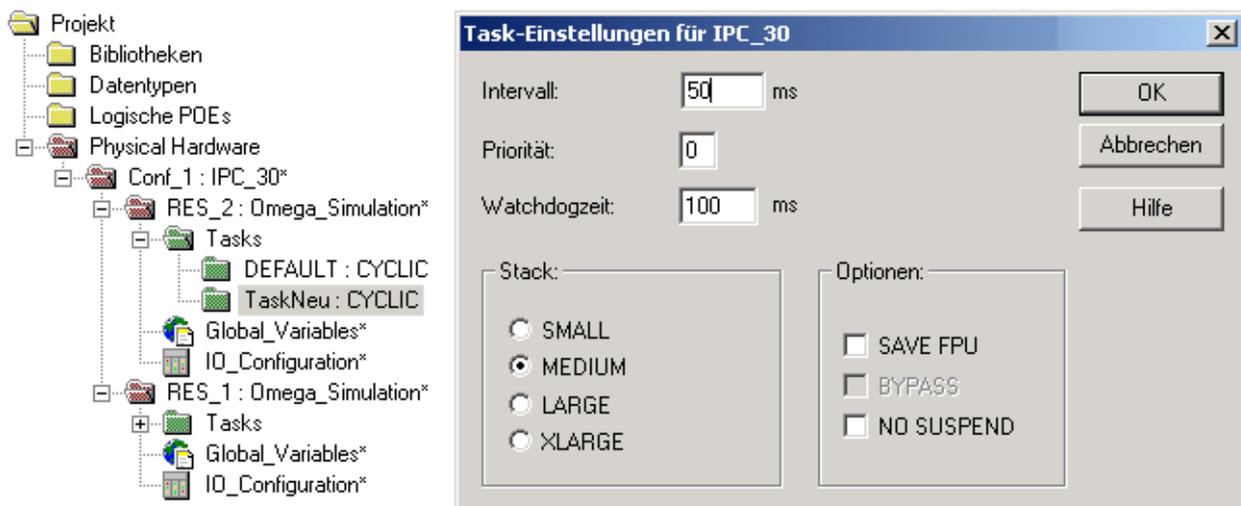


Abbildung 125: Task-Einstellungen für eine CYCLIC-Task



HINWEIS

Die Zykluszeit muss dabei ein Vielfaches der Systemtick-Zeit sein (siehe oben).

Beispiel: Bei Systemtick = 10 ms wirkt eine Eingabe von „11 ms“ oder „19 ms“ intern wie der Wert „10 ms“.

Werte, die kleiner als die Systemtick-Zeit sind, werden intern auf die Systemtick-Zeit aufgerundet (hier 10 ms).

11.1.3 Einfügen einer System-Task

Wird als Task-Typ der Typ "SYSTEM" ausgewählt (spezielle Systemtasks), kann man aus dem folgenden Menü auswählen:

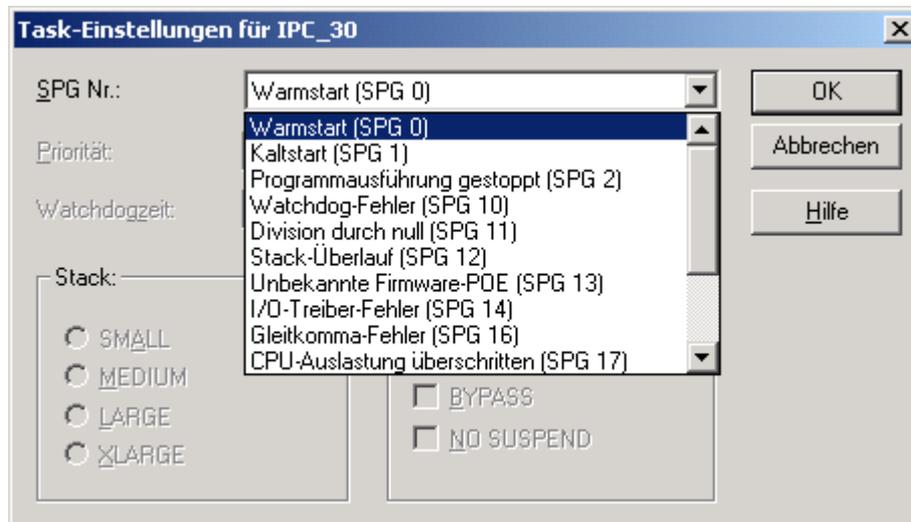


Abbildung 126: Task-Einstellungen für System-Task

Die Beschreibungen zu den einzelnen Systemtasks sind ebenfalls den entsprechenden Abschnitten der Bedienungs- und Applikationshandbüchern der jeweiligen Steuerung zu entnehmen.

11.2 Zuweisen von Programmen zu Tasks

Nachdem Sie eine neue Ressource im Projektbaum eingefügt haben, müssen Sie in dieser Ressource mindestens eine Task einfügen. Im folgenden Beispiel gehen wir davon aus, dass Sie bereits die Task 'DEFAULT' in der Ressource 'res_2' entsprechend der Beschreibung im vorhergehenden Abschnitt eingesetzt haben.

Damit das Projekt erfolgreich kompiliert werden kann, müssen den Tasks Programme zugewiesen werden, d. h. es muss festgelegt werden, in welcher Task ein Programm verarbeitet wird.



Zuweisen eines Programms zu einer Task mit der Maus

- Klicken Sie im Projektbaum mit der rechten Maustaste auf das Task-Symbol, um das Kontextmenü zu öffnen.
- Wählen Sie den Menüpunkt 'Einfügen'. Es erscheint der Dialog 'Einfügen'.





Zuweisen eines Programms zu einer Task mit der Tastatur

- Drücken Sie <↓> oder <↑>, um das Task-Symbol im Projektbaum zu markieren.
- Drücken Sie <EINFG>. Es erscheint der Dialog 'Einfügen'.

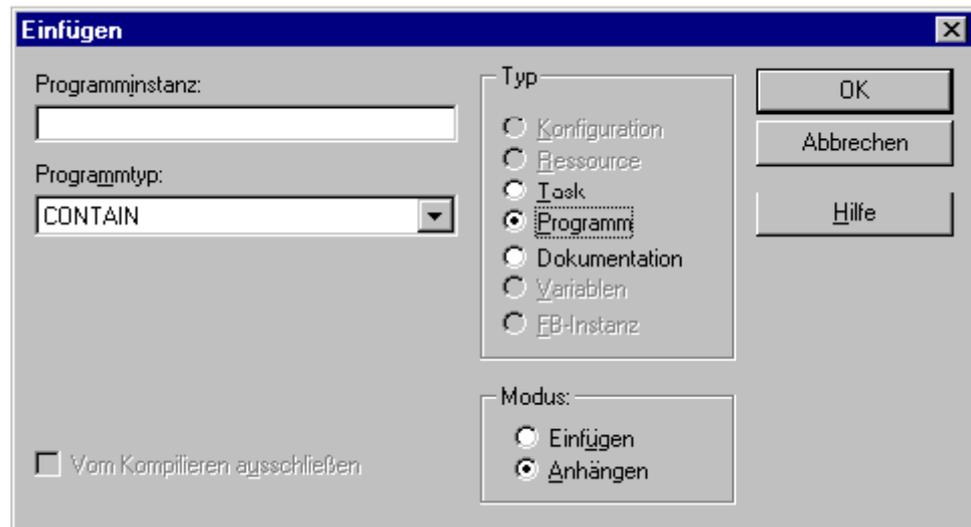


Abbildung 127: Dialog 'Einfügen'



Verwenden des Dialogs 'Einfügen'



- Aktivieren Sie die Option 'Programm' wie in ►Abbildung 127◀ auf Seite 182 gezeigt.
- Geben Sie in das Feld 'Programminstanz' den Instanznamen des Programms ein.
- Wählen Sie aus dem Listenfeld 'Programmtyp' den Programmnamen aus.
- Bestätigen Sie den Dialog. Das Programmsymbol wird unter der entsprechenden Task in den Projektbaum eingefügt.

11.3 Kompilieren eines Projekts

Beim Kompilieren eines Projekts werden die Inhalte der Arbeitsblätter in einen speziellen Code umgewandelt, der von Ihrer SPS ausgeführt werden kann. Dieser Vorgang gliedert sich in mehrere Schritte. Zuerst werden die verschiedenen Arbeitsblätter kompiliert, d. h. deren Syntax wird geprüft. Im zweiten Schritt werden die kompilierten Arbeitsblätter verbunden und es wird ein IEC-Zwischencode erzeugt. Im letzten Schritt wird der SPS-Code generiert.

Sie können mit dem System entweder das gesamte Projekt oder nur Teile davon kompilieren. Die folgende Liste zeigt die verschiedenen Möglichkeiten, die Ihnen das System

zum Kompilieren bietet. Sie können diese Funktionen mit Hilfe der Menüpunkte im Untermenü 'Code' oder den entsprechenden Symbolen in der Symbolleiste ausführen.

- * 'Make' - ist der Standardmodus zum Kompilieren, den Sie im Normalfall verwenden sollten, wenn Sie das Editieren beendet haben. Mit 'Make' kompilieren Sie alle Arbeitsblätter, die von Ihnen bearbeitet wurden. Diese werden im Projektbaum durch ein Sternchen gekennzeichnet. Wenn Sie 'Make' ausführen, wird der SPS-Code erzeugt und das Projekt kann anschließend an die SPS gesendet werden.
- * 'Online-Änderungen' - wird verwendet, um die Änderungen zu kompilieren, die von Ihnen z. B. nach dem Debuggen eines Projektes vorgenommen wurden. Die Änderungen werden automatisch an die SPS gesendet, man kann sie dann direkt im Online-Modus sehen. Um mit 'Online-Änderungen' die Änderungen zu kompilieren, muss die Programmausführung auf der SPS nicht gestoppt werden.
- * 'Arbeitsblatt kompilieren' - wird verwendet, um ein einzelnes Arbeitsblatt nach dem Editieren zu kompilieren. Der Kompiler überprüft dabei das aktuelle Arbeitsblatt und das zugehörige Variablen-Arbeitsblatt auf Syntaxfehler. Fehler und Warnungen, die beim Kompilieren auftreten, werden im Meldungsfenster protokolliert. Aus dem Meldungsfenster können Sie anschließend automatisch das Arbeitsblatt öffnen, in dem der Fehler oder die Warnung erkannt wurde. Doppelklicken Sie dazu einfach auf die entsprechende Meldung im Fenster.



HINWEIS

Beim Schließen oder Speichern eines Arbeitsblattes wird dieses automatisch kompiliert. Außerdem wird das erste Variablen-Arbeitsblatt in dieser POE ebenfalls kompiliert. Aus diesem Grund ist es wichtig, dass Sie neu eingefügte Variablen im ersten Arbeitsblatt der POE deklarieren.

Durch das automatische Kompilieren ist unmittelbar nach dem Speichern des Arbeitsblattes jede anwenderdefinierte Funktion bzw. jeder FB im Editor-Assistent verfügbar.

- * 'Projekt neu erzeugen' - wird verwendet, um das gesamte Projekt nach dem Editieren zum ersten Mal zu kompilieren. Sie sollten den Menüpunkt nur dann verwenden, wenn Sie Ihr Projekt ohne Zwischencode (Frontend) entpackt haben oder Änderungen in einer eingebundenen Anwenderbibliothek vorgenommen haben.
- * Mit 'Projekt neu erzeugen' werden alle Arbeitsblätter kompiliert und zusammengelinkt. Fehler und Warnungen, die dabei vom Kompiler entdeckt werden, werden im Meldungsfenster protokolliert. Nachdem die Syntax erfolgreich überprüft wurde, erzeugt das System automatisch den IEC-Code und den speziellen SPS-Code. Anschließend können Sie das Projekt an die SPS senden.

In den folgenden Abschnitten wird beschrieben, wie Sie Ihr Projekt bzw. die Arbeitsblätter mit den Menüpunkten 'Make' und 'Online-Änderungen' kompilieren.

11.4 Kompilieren eines Projekts mit 'Make'

Dieser Abschnitt beschreibt die Vorgehensweise beim Kompilieren von Änderungen, die Sie in den Arbeitsblättern vorgenommen haben. Die weiteren Möglichkeiten zum Kompilieren werden in der kontextsensitiven Hilfe beschrieben.

Mit dem Menüpunkt 'Make' bzw. dem Symbol 'Make' werden die geänderten Arbeitsblätter kompiliert und gelinkt, und es wird der geänderte SPS-Code erzeugt. Wenn dieser Vorgang erfolgreich abgeschlossen wird, d. h. es wurden keine Fehler gemeldet, können Sie anschließend das geänderte Projekt an die SPS senden.

HINWEIS



Bevor Sie mit dem Kompilieren beginnen, stellen Sie sicher, dass das Meldungsfenster geöffnet ist. In diesem Fenster können Sie sehen, welche Schritte der Kompiler momentan ausführt. Zusätzlich werden hier Fehler, Warnungen und zusätzliche Informationen zum Prozess protokolliert. Wenn das Fenster nicht geöffnet ist, drücken Sie <STRG> + <F2>.



Kompilieren der Änderungen mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Make'. Im Blatt 'Projekt erzeugen' des Meldungsfensters können Sie sehen, welche Schritte der Kompiler momentan ausführt. Wenn beim Kompilieren Fehler und Warnungen entdeckt wurden, werden diese in den entsprechenden Blättern des Meldungsfensters protokolliert.



Kompilieren der Änderungen mit der Tastatur

- Drücken Sie <F9>.
Im Blatt 'Projekt erzeugen' des Meldungsfensters können Sie sehen, welche Schritte der Kompiler momentan ausführt. Wenn beim Kompilieren Fehler und Warnungen entdeckt wurden, werden diese in den entsprechenden Blättern des Meldungsfensters protokolliert.

In den meisten Fällen treten beim ersten Kompilieren eines Projektes Programmierfehler auf, wie z. B. doppelt verwendete Variablennamen oder Schreibfehler. In diesen Fällen erscheint im Blatt 'Projekt erzeugen' des Meldungsfensters die Meldung, dass beim Kompilieren Fehler aufgetreten sind. Sie können der Meldung entnehmen, wie viele Fehler vom Kompiler entdeckt wurden.



Abbildung 128: Fehlermeldung nach dem Kompilieren eines Projektes

Um die beim Kompilieren entdeckten Fehlermeldungen anzuzeigen, klicken Sie auf den Reiter 'Fehler'. Die Fehlermeldungen erscheinen in der folgenden Form:

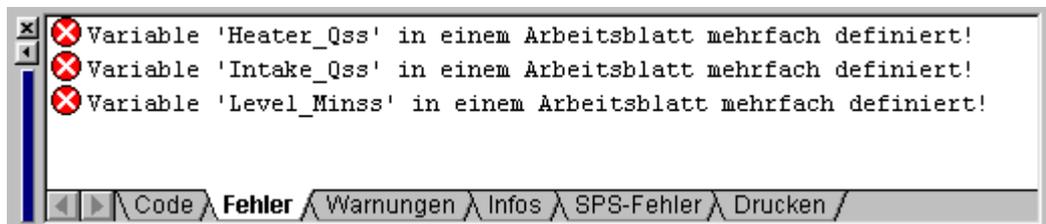


Abbildung 129: Fehlermeldungen, wie sie im Meldungsfenster ausgegeben werden

Wenn Sie die beim Kompilieren aufgetretenen Warnungen anzeigen möchten, klicken Sie einfach auf den Reiter 'Warnungen'.

In den meisten Fällen können Sie durch Doppelklicken auf einen Fehler bzw. eine Warnung direkt das Arbeitsblatt öffnen, in dem der Programmierfehler bzw. der Grund für die Warnung aufgetreten ist. Die betreffende Zeile oder das Objekt wird dann im Arbeitsblatt markiert. Das System bietet Ihnen zusätzlich eine Hilfeseite zu allen Fehlern an. Dort erfahren Sie, weshalb der Fehler aufgetreten ist und wie Sie diesen korrigieren. Um eine Hilfeseite aufzurufen, markieren Sie den Fehler im Meldungsfenster und drücken <F1>.

HINWEIS



Nachdem Sie den ersten Fehler korrigiert haben, können Sie mit <STRG> + <F12> direkt in das Arbeitsblatt mit dem nächsten Fehler gelangen.

11.5 Online-Änderungen von POEs

Das 'Online-Ändern' einer POE bedeutet, dass die Änderungen, die Sie während des Debuggens eines Projektes vorgenommen haben, kompiliert werden, der entsprechende Code erzeugt und automatisch an die laufende SPS gesendet wird. Um ein geändertes Projekt an das Zielsystem zu senden, muss die Programmausführung auf der SPS nicht gestoppt werden.

Wenn Sie im Online-Modus einen Programmierfehler entdeckt und in den Offline-Modus geschaltet haben, um den Fehler zu korrigieren, können Sie die vorgenommenen Änderungen mit 'Online-Änderungen' kompilieren. Die Änderungen werden automatisch an die SPS gesendet während die SPS in Betrieb bleibt, d. h. die Änderungen können sofort direkt im Online-Modus betrachtet werden.

In der folgenden Tabelle sind die Fälle aufgelistet, in denen Sie den Menüpunkt 'Online-Änderungen' verwenden können.

Sprache	Sie können 'Online-Änderungen' verwenden, wenn Sie ...
alle	<ul style="list-style-type: none"> * neue lokale oder globale Variablen eingefügt haben. * einen neuen Funktionsaufruf eingefügt haben, wenn die Funktion bereits zuvor in der POE aufgerufen wurde. * einen Funktionsbausteinufruf eingefügt haben, wenn der Funktionsbaustein bereits zuvor in der POE instanziiert wurde. * neue, leere Zeilen eingefügt haben. * Kommentare geändert oder hinzugefügt haben. * Variablen gelöscht haben, die nicht in der CSV-Datei (*.csv) verwendet werden.
AWL	<ul style="list-style-type: none"> * AWL-Operatoren geändert oder eingefügt haben. * die Verschachtelungstiefe geändert haben. * neue lokale Variablen deklariert haben.
ST	<ul style="list-style-type: none"> * Anweisungen oder Ausdrücke geändert haben.
FBS	<ul style="list-style-type: none"> * vorhandene Netzwerke geändert haben.
KOP	<ul style="list-style-type: none"> * vorhandene Netzwerke geändert haben.
AS	<ul style="list-style-type: none"> * FBS-Netzwerke, KOP-Netzwerke oder Variablen geändert haben, die mit Transitionen mit der Eigenschaft 'direkte Verbindung' verbunden sind. * das Zeitintervall für Qualifier für Aktionen geändert haben. * den Variablennamen in Aktionsblöcken geändert haben.

Abbildung 130: Fälle, in denen der Menüpunkt 'Online-Änderungen' verwendet werden kann



HINWEIS

Wenn Sie den Menüpunkt 'Online-Änderungen' für POEs verwenden möchten, in denen neue Variablen eingefügt wurden, sollten Sie die POE-Speicherreserve für diese Änderungen verwenden. Dazu aktivieren Sie das Kontrollkästchen 'Reserve verwenden' im Eigenschaftsdialog der POE und die entsprechenden Optionen im Dialog 'Ressource-Einstellungen für ...'. Die Reservegröße können Sie im Dialog 'Datenbereich' einstellen.

Sie können den Menüpunkt 'Online-Änderungen' nicht in den folgenden Fällen verwenden:

- * nach dem Ändern von Zeichenfolge-Konstanten, anwenderdefinierten Zeichenfolgen, Variablen, der Hardwarestruktur und der Formalparameter von Funktionen und Funktionsbausteinen (VAR_INPUT, VAR_OUTPUT und VAR_IN_OUT).
- * nach dem Einfügen neuer Zeichenfolge-Konstanten, anwenderdefinierter Zeichenfolgen, Funktionen, Funktionsbausteininstanzen, POEs, Bibliotheken und CASE-Anweisungen ohne die Reserve zu verwenden
- * nach dem Löschen von POEs oder Bibliotheken.



HINWEIS

Der Menüpunkt 'Online-Änderungen' ist nur im Offline-Modus verfügbar. Sie können mit dem Symbol 'Debug ein/aus' in der Symbolleiste zwischen Online- und Offline-Modus umschalten.



Online-Änderungen vornehmen mit der Maus

- Vergewissern Sie sich, dass das Arbeitsblatt das aktive Fenster ist.
- Vergewissern Sie sich, dass das Arbeitsblatt im Offline-Modus ist. Im Offline-Modus erscheint das Symbol 'Debug ein/aus' nicht gedrückt.
- Editieren Sie Ihr Arbeitsblatt und korrigieren Sie Programmierfehler.
- Wählen Sie aus dem Untermenü 'Code' den Menüpunkt 'Online-Änderungen'.



Der Compiler wird gestartet. Das Meldungsfenster zeigt Ihnen die vom Compiler ausgeführten Schritte.

Wenn beim Kompilieren Fehler und Warnungen entdeckt wurden, werden diese in den entsprechenden Blättern des Meldungsfensters protokolliert.



Eine Beschreibung, wie Sie Fehler und Warnungen anzeigen und die betreffenden Arbeitsblätter öffnen, finden Sie im vorhergehenden Abschnitt 'Kompilieren eines Projekts mit 'Make''.

11.6 Senden an das Zielsystem

Nach dem Kompilieren Ihres Projekts mit 'Make' oder 'Projekt neu erzeugen' muss es an die Simulation oder SPS gesendet werden. Für den Fall, dass Sie Änderungen in Ihrem Projekt mit 'Online-Änderungen' kompiliert haben, wird der neu erzeugte Code automatisch an das Zielsystem gesendet. In diesem Fall muss die Programmausführung auf der SPS nicht gestoppt werden.



HINWEIS

Das Zielsystem für den Sendevorgang stellen Sie im Dialog 'Ressource-Einstellungen für ...' ein. Um den Dialog aufzurufen, wählen Sie den Menüpunkt 'Einstellungen...' aus dem Kontextmenü der Ressource. Aktivieren Sie in diesem Dialog die betreffende Simulation oder das Zielsystem, an welche bzw. an welches das SPS-Programm gesendet werden soll.



Senden des Projekts mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Kontrolldialog anzeigen'. Wenn in Ihrem Projekt nur eine Ressource vorhanden ist, erscheint direkt der Kontrolldialog.



Wenn der Projektbaum mehrere Ressourcen enthält, erscheint der Dialog 'Ressource auswählen'. Wählen Sie hier die gewünschte Ressource und bestätigen Sie den Dialog, um den Kontrolldialog aufzurufen.



Senden des Projekts mit der Tastatur

- Drücken Sie <STRG> + <F10>.

Wenn in Ihrem Projekt nur eine Ressource vorhanden ist, erscheint direkt der Kontrolldialog.

Wenn der Projektbaum mehrere Ressourcen enthält, erscheint der Dialog 'Ressource auswählen'. Wählen Sie hier die gewünschte Ressource und bestätigen Sie den Dialog mit <↵>, um den Kontrolldialog aufzurufen.



Abbildung 131: Kontrolldialog



Verwenden des Kontrolldialogs

- Klicken Sie im Kontrolldialog auf 'Senden'. Es erscheint der Dialog 'Senden'.





Abbildung 132: Dialog 'Senden'

Verwenden des Dialogs 'Senden'



- Drücken Sie im Bereich 'Projekt' die Schaltfläche 'Senden'. Das gesamte Projekt einschließlich der POEs und der Konfigurationsdaten wird an das Zielsystem gesendet. Die SPS geht in den Status 'STOP', was Sie im Kontrolldialog sehen können.
- Drücken Sie im Kontrolldialog die Schaltfläche 'Kalt', um die Programmausführung zu starten.

Detaillierte Informationen zum Senden an das Zielsystem finden Sie in der kontextsensitiven Hilfe.



Zusätzlich enthält die Online-Hilfe detaillierte Informationen zum Senden

- eines **Bootprojekts** an das Zielsystem. Dieses Projekt wird automatisch nach dem Einschalten der SPS oder nach einem Stromausfall ausgeführt.
- der **Programmquelle** an das Zielsystem. Hier werden die gepackten Projektdateien in der SPS gespeichert. Die Programmquelle kann anschließend wieder von der SPS hochgeladen werden.
- einer **Datei jedes beliebigen Typs** (z. B. eine ASCII-Datei) an das Zielsystem. Diese Datei kann für zukünftige Anwendungen verwendet werden.

HINWEIS



Die Unterstützung dieser Funktionen ist abhängig vom Zielsystem. Einzelheiten hierzu erfahren Sie von Ihrem SPS-Hersteller.

11.7 Aufrufen von Arbeitsblättern im Online-Modus

Nach dem Kompilieren und dem Senden des Projektes an das Zielsystem, können Sie die Editoren zum Debuggen der Arbeitsblätter in den Online-Modus schalten. Dies erreichen Sie, indem Sie:

- * in der Symbolleiste auf das Symbol 'Debug ein/aus' klicken oder die Funktionstaste <F10> drücken. In beiden Fällen werden alle momentan geöffneten Arbeitsblätter automatisch in den Online-Modus geschaltet. Dies gilt auch für Arbeitsblätter, die Sie anschließend aus dem Projektbaum oder dem Instanzbaum öffnen. Das Symbol 'Debug ein/aus' erscheint bei aktiviertem Online-Modus gedrückt. Beachten Sie hierzu auch den folgenden Hinweis.
- * den Menüpunkt 'Instanz öffnen...' aus dem Kontextmenü der POE im Projektbaum oder einem geöffneten Arbeitsblatt auswählen. Beachten Sie hierzu auch den folgenden Hinweis.



HINWEIS

Wenn geöffnete Funktionsbaustein- oder Programm-Codes mehrfach instanziiert sind und Sie diese Arbeitsblätter debuggen möchten, d. h. in den Online-Modus mit dem Symbol 'Debug ein/aus' schalten möchten, erscheint eine Warnmeldung. Darin werden Sie angewiesen, diese Arbeitsblätter mit der Menüoption 'Instanz öffnen...' in den Online-Modus zu schalten. Wählen Sie in diesem Fall den Menüpunkt 'Instanz öffnen...' aus dem Kontextmenü der entsprechenden Arbeitsblätter. Anschließend erscheint der Dialog 'Instanz öffnen', in dem Sie die gewünschte Instanz auswählen müssen.

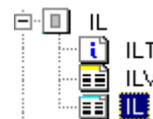
Der Dialog 'Instanz öffnen' erscheint auch, wenn der Online-Modus eingeschaltet ist und Sie Funktionsbaustein- oder Programm-Codes aus dem Projektbaum öffnen möchten, die mehrfach instanziiert sind.

In der folgenden Beschreibung nehmen wir an, dass Sie ein textuelles Arbeitsblatt im Online-Modus debuggen möchten.



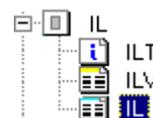
Aufrufen eines textuellen Arbeitsblattes im Online-Modus mit der Maus

- Wenn das gewünschte Arbeitsblatt noch nicht geöffnet ist, öffnen Sie den betreffenden Unterbaum mit dem Arbeitsblatt durch Doppelklicken auf den POE-Namen.
- Doppelklicken Sie auf das Arbeitsblatt-Symbol (z. B. 'IL'). Das Arbeitsblatt wird geöffnet.
- Klicken Sie in der Symbolleiste auf das Symbol 'Debug ein/aus'. Es werden alle geöffneten Arbeitsblätter in den Online-Modus geschaltet. Das Symbol 'Debug ein/aus' erscheint bei aktiviertem Online-Modus gedrückt.



Aufrufen eines textuellen Arbeitsblattes im Online-Modus mit der Tastatur

- Wenn das gewünschte Arbeitsblatt noch nicht geöffnet ist, öffnen Sie den betreffenden Unterbaum mit dem Arbeitsblatt wie folgt: Drücken Sie <↓> oder <↑>, um den POE-Namen zu markieren und drücken Sie anschließend <→>, um den Unterbaum zu öffnen.
- Drücken Sie <↓> oder <↑>, um das Arbeitsblatt-Symbol (z. B. 'IL') zu markieren und drücken Sie anschließend <↵>. Das Arbeitsblatt wird geöffnet.
- Drücken Sie <F10>. Es werden alle geöffneten Arbeitsblätter in den Online-Modus geschaltet. Das Symbol 'Debug ein/aus' erscheint bei aktiviertem Online-Modus gedrückt.



1	FALSE	LD	%IX0.2
2	FALSE	AND	%IX0.3
3	FALSE	OR	Action_INIT
4	FALSE	ST	IL_VAR
5			
6	FALSE	LD	Input_IX0_0
7		JMPC	MANUAL
8			
9			(* Timer FB TON *)
10	TRUE	LD	Timer_start
11	TRUE	ST	TON_IL.IN
12	1.500	LD	PT_TON_IL
13	1.500	ST	TON_IL.PT

Abbildung 133:AWL-Arbeitsblatt im Online-Modus

Ein textuelles Arbeitsblatt wird im Online-Modus durch eine vertikale graue Linie in zwei Teile gegliedert. Auf der linken Seite werden die Online-Werte dargestellt und die rechte Seite zeigt den Code.



HINWEIS

Sie können die Spaltenbreite ändern, indem Sie den Cursor auf die graue Trennlinie bewegen (wenn sich der Cursor auf der Linie befindet, ändert er seine Form in eine Doppellinie) und bei gedrückter Maustaste den Cursor nach links oder nach rechts bewegen.

Das folgende Beispiel zeigt ein graphisches Arbeitsblatt im Online-Modus:

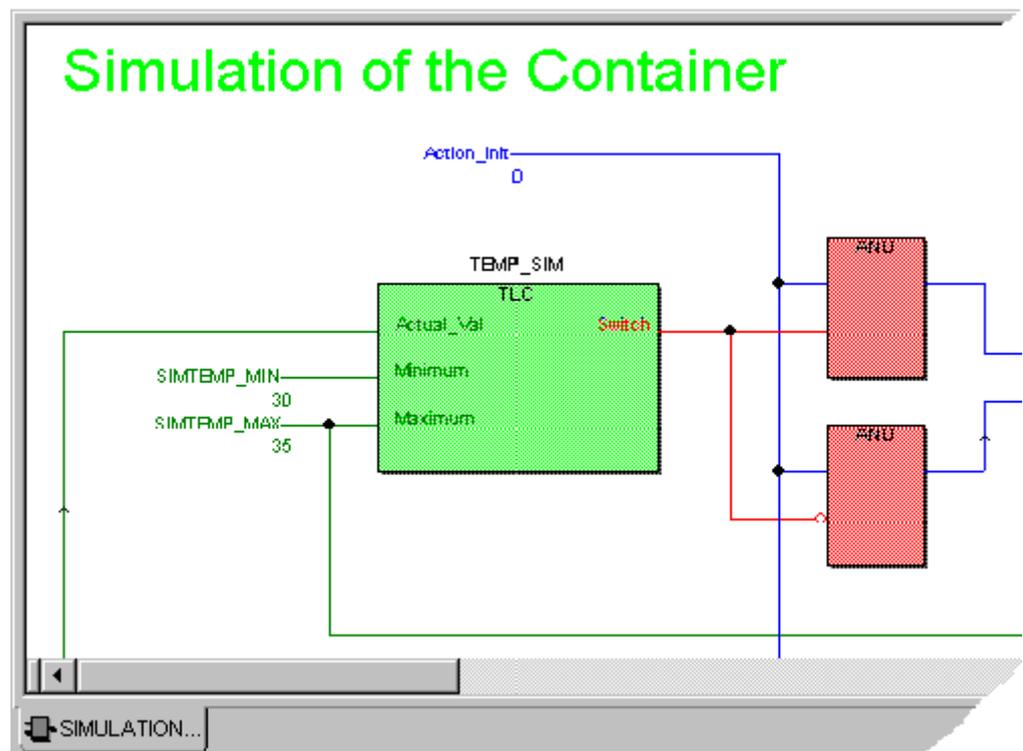


Abbildung 134: Graphisches Arbeitsblatt im Online-Modus

Die Farben, die in textuellen und in graphischen Arbeitsblättern verwendet werden, haben folgende Bedeutung:

Farbe	Bedeutung
Rot	Boolesches TRUE
Blau	Boolesches FALSE
Grün	Integer-Werte
Rot	Gesetzter Breakpoint
Orange	Erreichter Breakpoint

Abbildung 135: Bedeutung der Farben im Online-Modus



HINWEIS

Sie können die Darstellung der Online-Werte in graphischen Arbeitsblättern im Dialog 'Online-Layout' ändern. Um diesen Dialog aufzurufen, wählen Sie den Menüpunkt 'Online-Layout...' aus dem Untermenü 'Layout'.



Aufrufen eines Arbeitsblattes einer Anwenderbibliothek im Online-Modus mit der Maus

Da die Möglichkeit besteht, Code-Arbeitsblätter von eingebundenen Anwenderbibliotheken im Anzeigemodus zu öffnen (d. h. der Inhalt der Arbeitsblätter kann angezeigt aber nicht bearbeitet werden), können diese auch in den Online-Modus geschaltet werden. Das bedeutet, die Arbeitsblätter werden beim Debuggen des Projekts wie "normale" Code-Arbeitsblätter behandelt.



HINWEIS

Arbeitsblätter aus eingebundenen Firmware-Bibliotheken können weder im Anzeigemodus noch im Online-Modus geöffnet werden.



- Markieren Sie das Bibliothekarbeitsblatt, das Sie öffnen wollen. Falls der Knoten nicht bereits geöffnet ist, doppelklicken Sie auf das Knotensymbol 'Bibliotheken'.
- Suchen Sie die gewünschte Bibliothek und öffnen Sie den POE-Ordner.
- Doppelklicken Sie auf das gewünschte Symbol des Bibliothekarbeitsblattes. Das Arbeitsblatt wird geöffnet.
- Klicken Sie in der Symbolleiste auf das Symbol 'Debug ein/aus'. Es werden alle geöffneten Arbeitsblätter in den Online-Modus geschaltet. Das Symbol 'Debug ein/aus' erscheint bei aktiviertem Online-Modus gedrückt.



11.8 Einstellen des Layout und der Notation von Online-Werten

Das Programmiersystem bietet Ihnen mehrere Möglichkeiten die Darstellung von Arbeitsblättern und die Anzeige der Online-Werte Ihren Anforderungen anzupassen:

- * Es kann die Notation der Online-Werte in allen Arbeitsblättern im Online-Modus eingestellt werden. Wie Sie diese Einstellungen vornehmen wird in der folgenden Vorgehensweise beschrieben.
- * Es kann das Online-Layout für graphische Arbeitsblätter geändert werden (siehe hierzu folgenden Hinweis).
- * Es können Rückkopplungen in graphischen Arbeitsblättern hervorgehoben werden (siehe hierzu folgenden Hinweis).

- * Es kann die Abarbeitungsreihenfolge von FUs/FBs in graphischen Arbeitsblättern angezeigt werden (siehe hierzu folgenden Hinweis).



Detaillierte Informationen zum Ändern des Online-Layout, zum Hervorheben von Rückkopplungen und zum Anzeigen der Abarbeitungsreihenfolge finden Sie im jeweiligen Thema der SPS-spezifischen Online-Hilfe.



Ändern der Notation von Online-Werten

Dies erfolgt im Dialog 'Debug: *Ressource_Name*'. Beachten Sie, dass die ausgewählte Darstellungsform (Notation) für alle Arbeitsblätter gültig ist.

- Markieren Sie in einem Arbeitsblatt im Online-Modus eine beliebige Variable und öffnen Sie das Kontextmenü der Variablen.
- Wählen Sie den Menüpunkt 'Debug-Dialog'.
Es erscheint der Dialog 'Debug: *Ressource_Name*'.
- Wählen Sie in der Gruppe 'Angez. Wert' eine der Optionen 'Standard', 'Dezimal', 'Hexadezimal' oder 'Binär'.
- Bestätigen Sie den Dialog. Die Notation der angezeigten Online-Werte wird entsprechend dieser Einstellung in allen Arbeitsblättern geändert.

11.9 Umschalten zwischen Online- und Offline-Modus

Wenn Sie im Online-Modus einen Programmierfehler in einem beliebigen Arbeitsblatt gefunden haben, können Sie in den Offline-Modus schalten, um den Programmierfehler sofort zu korrigieren. Nachdem Sie den Fehler korrigiert haben, müssen Sie mit der Funktion 'Online-Änderungen' die Änderungen kompilieren und an das Zielsystem senden.



Umschalten zwischen Online- und Offline-Modus mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Debug ein/aus'.
Es werden alle geöffneten Arbeitsblätter automatisch in den Offline-Modus geschaltet. Das Symbol 'Debug ein/aus' erscheint bei aktiviertem Offline-Modus nicht gedrückt.
- Machen Sie Ihre Programmänderungen.
- Wählen Sie aus dem Untermenü 'Code' den Menüpunkt 'Online-Änderungen'. Die vorgenommenen Änderungen werden kompiliert und an das Zielsystem gesendet.
- Klicken Sie in der Symbolleiste auf das Symbol 'Debug ein/aus'.
Es werden alle Arbeitsblätter automatisch in den Online-Modus geschaltet. Das Symbol 'Debug ein/aus' erscheint bei aktiviertem Online-Modus gedrückt.





Umschalten zwischen Online- und Offline-Modus mit der Tastatur

- Drücken Sie <F10>. Es werden alle geöffneten Arbeitsblätter automatisch in den Offline-Modus geschaltet. Das Symbol 'Debug ein/aus' erscheint bei aktiviertem Offline-Modus nicht gedrückt.
- Machen Sie Ihre Programmänderungen.
- Drücken Sie <ALT> + <F9>. Die vorgenommenen Änderungen werden kompiliert und an das Zielsystem gesendet.
- Drücken Sie <F10>. Es werden alle Arbeitsblätter automatisch in den Online-Modus geschaltet. Das Symbol 'Debug ein/aus' erscheint bei aktiviertem Online-Modus gedrückt.



HINWEIS

Informationen zum Aufrufen von Funktionsbaustein- und Programm-Codes im Online-Modus, die mehrfach instanziiert sind, finden Sie im Abschnitt 'Aufrufen von Arbeitsblättern im Online-Modus'.

11.10 Umschalten in Adressstatus mit Durchlaufkontrolle

Normalerweise verwenden Sie im Online-Modus den Variablenstatus, in dem Sie das Verhalten der Variablen überprüfen können. Im Variablenstatus werden die Variablen am Ende eines Arbeitszyklus im Prozessabbildspeicher gespeichert. Sie können jedoch auch in den Adressstatus umschalten, der Ihnen die aktuellen Werte des Akkumulators entsprechend der momentanen Programmausführung liefert. Im Adressstatus wird zusätzlich die Durchlaufkontrolle (Powerflow) angezeigt, d. h. Sie können sehen, welche Programmteile derzeit ausgeführt werden und welche nicht. Dies ist beispielsweise beim Debuggen von Arbeitsblättern mit bedingten Sprüngen von Bedeutung.



HINWEIS

Durch Aktivierung des Adressstatus erhöht sich die Zykluszeit des betrachteten Programnteils. Deswegen sollte der Adressstatus nicht in den POEs der Bypass-Tasks und Event-Tasks aktiviert werden!

Adressstatus und Durchlaufkontrolle können nur zusammen verwendet werden, d. h. Sie können nicht in den Adressstatus ohne Durchlaufkontrolle schalten.

In einem geöffneten Funktions-Arbeitsblatt im Online-Modus ist der Adressstatus mit Durchlaufkontrolle immer eingeschaltet.



Umschalten in Adressstatus mit Durchlaufkontrolle mit der Maus

- Klicken Sie auf das Untermenü 'Online'. Das Untermenü wird geöffnet.
- Wählen Sie den Menüpunkt 'Adressstatus'. Das Arbeitsblatt wird in den Adressstatus mit Durchlaufkontrolle umgeschaltet.

Im Texteditor wird die Durchlaufkontrolle im Online-Modus durch drei unterschiedliche Zeichen dargestellt. Die Bedeutung dieser Zeichen wird in der folgenden Tabelle erklärt:

Zeichen der Durchlaufkontrolle	Bedeutung
Horizontale Linie	der Programmteil wird nicht durchlaufen.
Vertikale Linie	der Programmteil wird durchlaufen.
Vertikale, doppelte Linie	Programmteil wird in mehreren POEs oder in einer Schleife durchlaufen, die mehrfach abgearbeitet wird.

Abbildung 136: Zeichen der Durchlaufkontrolle in textuellen Arbeitsblättern im Online-Modus

Im graphischen Editor wird die Durchlaufkontrolle im Online-Modus mit farbigen Linien dargestellt. Die Bedeutung dieser Linien ist in der folgenden Tabelle beschrieben:

Zeichen der Durchlaufkontrolle	Bedeutung
Rote Linie	TRUE
Blaue Linie	FALSE
Kurze, horizontale Linie	der Programmteil wird durchlaufen.
Kurze, vertikale Linie	der Programmteil wird nicht durchlaufen.
Kurze, vertikale, doppelte Linie	Programmteil wird in mehreren POEs oder in einer Schleife durchlaufen, die mehrfach abgearbeitet wird.

Abbildung 137: Zeichen der Durchlaufkontrolle in graphischen Arbeitsblättern im Online-Modus

11.11 Forcen und Überschreiben von Variablen

Variablen können im Online-Modus geforct oder überschrieben werden, d. h. es kann den Variablen ein neuer Wert zugewiesen werden.

Was ist der Unterschied zwischen 'Forcen' und 'Überschreiben'?

- * Beim **Überschreiben** einer Variablen mit einem Wert wird der neue Wert nur für einen Arbeitszyklus verwendet. Nach dem Beenden dieses Zyklus wird die Variable wieder normal verarbeitet.
- * Beim **Forcen** einer Variablen wird der neue Wert solange für die Variable verwendet, bis Sie für die betreffende Variable das Forcen wieder zurücknehmen.

Die Vorgehensweise beim Forcen und Überschreiben von Variablen ist nahezu identisch.

**HINWEIS**

Seien Sie vorsichtig, wenn Sie Variablen forcen oder überschreiben, wenn Ihre SPS läuft, da das SPS-Programm mit den Werten der geforcten oder überschriebenen Variablen ausgeführt wird.

Es können nur physikalische Ein- und Ausgänge geforct werden.

Die Unterstützung dieser Funktionen ist abhängig vom Zielsystem. Einzelheiten hierzu erfahren Sie von Ihrem SPS-Hersteller.

**Forcen von Variablen mit der Maus**

- Wählen Sie den Menüpunkt 'Debug-Dialog' aus dem Kontextmenü zur Variablen in Ihrem Arbeitsblatt im Online-Modus.
Es erscheint der Dialog 'Debug: *Ressource_Name*'.

**Forcen von Variablen mit der Tastatur**

- Gehen Sie mit Hilfe der Cursortasten zu der betreffenden Variable in Ihrem Arbeitsblatt im Online-Modus.
- Drücken Sie die <UMSCHALT>-Taste und halten Sie diese gedrückt. Drücken Sie <→> oder <←>, um die Variable zu markieren.
- **Im graphischen Editor:**
Drücken Sie <↵>.
Es erscheint der Dialog 'Debug: *Ressource_Name*'.

Im Texteditor:

Wählen Sie den Menüpunkt 'Debug-Dialog' aus dem Kontextmenü zur Variablen in Ihrem Arbeitsblatt im Online-Modus.

Es erscheint der Dialog 'Debug: *Ressource_Name*'.

Oder

Doppelklicken Sie auf den betreffenden Online-Wert der Variablen im Online-Modus.

Es erscheint der Dialog 'Debug: *Ressource_Name*'.

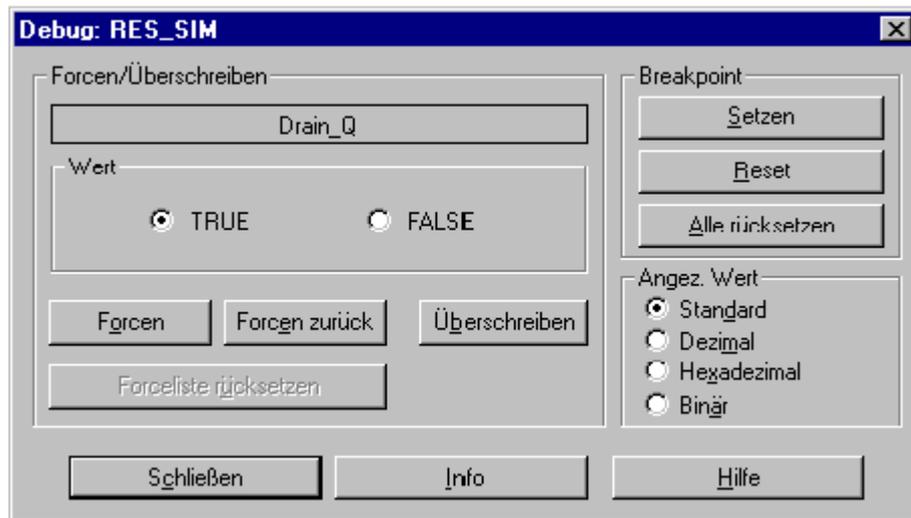


Abbildung 138: Dialog 'Debug: Ressource_Name'

Verwenden des Dialogs 'Debug: Ressource_Name'



- Aktivieren Sie die Option 'TRUE' oder 'FALSE'.
Wenn Sie eine nicht-boolesche Variable forcen oder überschreiben wollen, erscheint im Dialog ein Eingabefeld anstelle der Optionen. Geben Sie in diesem Fall den benötigten Wert ein.

HINWEIS



Der Dialog "speichert" den Wert eines Datentypes. So wird im Falle von BOOL der aktuelle Wert von der SPS gelesen und daraufhin der gegenteilige Wert vorgeschlagen. D. h. ist der aktuelle Wert "TRUE", so wird der Wert "FALSE" im Dialog markiert.



- Drücken Sie die Schaltfläche 'Forcen'.
- Bestätigen Sie den Dialog.
Die Variable wird geforct.

Wenn Sie eine oder mehrere Variablen geforct haben und Sie den Vorgang wieder rückgängig machen möchten, können Sie entweder nur einzelne geforcte Variablen oder auch alle geforcten Variablen auf den normalen Wert rücksetzen.



Rücksetzen von geforcten Variablen

- Wählen Sie den Menüpunkt 'Debug-Dialog' aus dem Kontextmenü zur geforcten Variablen in Ihrem Arbeitsblatt im Online-Modus.
- Es erscheint der Dialog 'Debug: Ressource_Name', den Sie in Abbildung 11-14 sehen können.
- Drücken Sie im Bereich 'Forcen/Überschreiben' die Schaltfläche
 - 'Forcen zurück', um die markierte Variable auf den normalen Wert rückzusetzen.
 - 'Forceliste rücksetzen', um alle geforcten Variablen gleichzeitig rückzusetzen.

11.12 Setzen und Rücksetzen von Breakpoints

Im Online-Modus können Sie in Code-Arbeitsblättern Breakpoints setzen oder rücksetzen. Ein Breakpoint hat zur Folge, dass das Programm gestoppt wird, wenn es auf den Breakpoint läuft.

Wenn Breakpoints gesetzt wurden, bietet der Kontrolldialog mehrere Schaltflächen, um die SPS im Debug-Modus zu steuern: Mit der Schaltfläche 'Weiter' können Sie die Programmausführung am Breakpoint fortsetzen. Wenn Sie das Programm von vorne starten möchten, drücken Sie einfach die Schaltfläche 'Wiederanl.' im Kontrolldialog.

HINWEIS



Seien Sie vorsichtig, wenn Sie Breakpoints verwenden, während Ihre SPS läuft. Das Programm wird gestoppt, wenn es auf den Breakpoint aufläuft.

Das Verhalten der I/O's beim Erreichen eines Breakpoints ist von der angeschlossenen SPS abhängig.

Sie können Breakpoints nicht in Funktionsbaustein-Instanzen setzen. Das Setzen eines Breakpoints in einer Instanz bedeutet automatisch, dass die Programmausführung jedes Mal angehalten wird, wenn der Funktionsbaustein aufgerufen wird.



Setzen eines Breakpoints mit der Maus

- Doppelklicken Sie auf die gewünschte Zeile oder das Objekt in Ihrem Arbeitsblatt im Online-Modus.
Es erscheint der Dialog 'Debug: Ressource_Name'.



Setzen eines Breakpoints mit der Tastatur

- Gehen Sie mit Hilfe der Cursortasten zu der gewünschten Zeile oder dem Objekt in Ihrem Arbeitsblatt im Online-Modus.

- **Im graphischen Editor:**

Drücken Sie <↵>.

Es erscheint der Dialog 'Debug: Ressource_Name'.

- **Im Texteditor:**

Wählen Sie den Menüpunkt 'Debug-Dialog' aus dem Kontextmenü zur Variablen in Ihrem Arbeitsblatt im Online-Modus.

Es erscheint der Dialog 'Debug: Ressource_Name'.

Oder

Doppelklicken Sie auf den betreffenden Online-Wert der Variablen im Online-Modus. Es erscheint der Dialog 'Debug: Ressource_Name'.

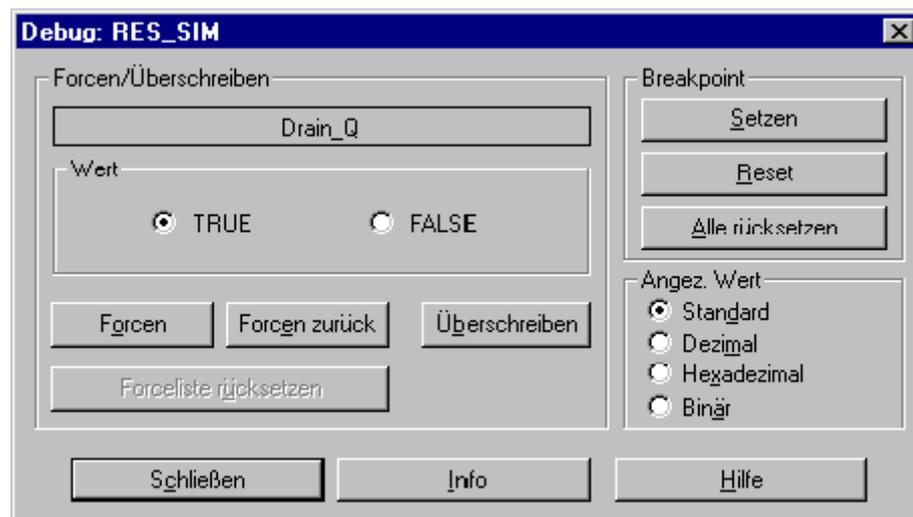


Abbildung 139: Dialog 'Debug: Ressource_Name'



Verwenden des Dialogs 'Debug: Ressource_Name'



- Drücken Sie die Schaltfläche 'Setzen'.
Der Breakpoint wird gesetzt.



HINWEIS

In den Arbeitsblättern wird ein gesetzter Breakpoint rot und ein erreichter Breakpoint orange dargestellt.



Wenn Sie Breakpoints gesetzt haben, bietet Ihnen das System mehrere Möglichkeiten, das SPS-Programm zu debuggen. Diese Funktionen sind im folgenden Abschnitt 'Debuggen mit gesetzten Breakpoints' beschrieben.

Wenn Sie Breakpoints gesetzt haben und diesen Vorgang wieder rückgängig machen möchten, können Sie entweder einzelne Breakpoints oder alle Breakpoints rücksetzen.



Rücksetzen von Breakpoints

- **Im graphischen Editor:**

Drücken Sie <↵>.

Es erscheint der Dialog 'Debug: *Ressource_Name*'.

- **Im Texteditor:**

Wählen Sie den Menüpunkt 'Debug-Dialog' aus dem Kontextmenü zur Variablen in Ihrem Arbeitsblatt im Online-Modus.

Es erscheint der Dialog 'Debug: *Ressource_Name*'.

Oder

Doppelklicken Sie auf den betreffenden Online-Wert der Variablen im Online-Modus. Es erscheint der Dialog 'Debug: *Ressource_Name*'.

- Drücken Sie im Bereich 'Breakpoint' die Schaltfläche
 - 'Reset', um den Breakpoint für das markierte Objekt rückzusetzen.
 - 'Alle rücksetzen', um alle gesetzten Breakpoints rückzusetzen.

11.13 Debuggen mit gesetzten Breakpoints

Wenn Sie einen oder mehrere Breakpoints gesetzt haben, können Sie mit Hilfe der Step- und Trace-Funktion Ihr SPS-Programm debuggen. Dies ermöglicht Ihnen die Programmausführung Zeile für Zeile fortzusetzen, nachdem das Programm auf einen Breakpoint gelaufen ist.

Was ist der Unterschied zwischen der Step- und Trace-Funktion?

- * Wenn Sie mit der Funktion **Step** arbeiten und eine Funktion oder einen Funktionsbaustein erreichen, werden diese Elemente übersprungen und es wird der nächste Befehl

bzw. das nächste grafische Element ausgeführt. Dies bedeutet, dass Sie mit 'Step' Funktionen oder Funktionsbausteine nicht debuggen können.

- * Wenn Sie mit der Funktion **Trace** arbeiten, wird beim Erreichen einer Funktion oder eines Funktionsbausteins in den Code der Funktion oder des Funktionsbausteins hineingesprungen, d. h. das entsprechende Code-Arbeitsblatt wird zum Debuggen geöffnet.

Beide Funktionen werden im Kontrolldialog ausgeführt.



HINWEIS

Sie können die Funktionen 'Step' und 'Trace' nur dann verwenden, wenn ein Breakpoint gesetzt wurde.

In der folgenden Beschreibung nehmen wir an, dass Sie in einem Code-Arbeitsblatt in ST im Online-Modus einen Breakpoint gesetzt haben.



Stepping und Tracing nach einem Breakpoint mit der Maus

- Wenn der Kontrolldialog nicht geöffnet ist, klicken Sie in der Symbolleiste auf das Symbol 'Kontrolldialog anzeigen'. Es erscheint der Kontrolldialog.



Stepping und Tracing nach einem Breakpoint mit der Tastatur

- Wenn der Kontrolldialog nicht geöffnet ist, drücken Sie <STRG> + <F10>. Es erscheint der Kontrolldialog.



Abbildung 140: Kontrolldialog bei gesetztem Breakpoint



Verwenden des Kontrolldialogs



- Drücken Sie die Schaltfläche 'Step', um zum nächsten Befehl oder Objekt zu springen und die Programmausführung an diesem Punkt zu stoppen. Wenn eine Funktion oder ein Funktionsbaustein erreicht wird, wird dieser übersprungen.
- Drücken Sie die Schaltfläche 'Trace', um zum nächsten Befehl oder Objekt zu springen und die Programmausführung an diesem Punkt zu stoppen. Wenn eine Funktion oder ein Funktionsbaustein erreicht wird, wird das Code-Arbeitsblatt der Funktion oder des Funktionsbausteins zum Debuggen geöffnet.

HINWEIS



Im Arbeitsblatt wird die Stelle, an der die Programmausführung angehalten wurde, orange markiert. In textuellen Arbeitblättern markiert zusätzlich ein gelber Pfeil die betreffende Zeile, wie Sie in der folgenden Abbildung sehen können.

Das folgende Beispiel zeigt ein ST-Arbeitsblatt im Online-Modus, in dem zwei Breakpoints gesetzt wurden (in Zeile 4 und Zeile 15). Die Programmausführung wurde in Zeile 15 angehalten.

```

1      2  CASE MODUS OF
2      -8000 1: ROBOT_X:= ROBOT_X + 200;
3      -4000 ROBOT_Z := ROBOT_Z + ADD_ARM ;
4      2     MODUS:=1;
5      -8000 IF ROBOT_X >= ROBOT_RANGE_POS_1 THEN
6      2     MODUS:=2;
7      END_IF;
8      -8000 2: ROBOT_X:= ROBOT_X - 200;
9      -4000 ROBOT_Z := ROBOT_Z - ADD_ARM ;
10     2     MODUS:=2;
11     -8000 IF ROBOT X <= ROBOT RANGE NEG 1 THEN
12     2     MODUS:=1;
13     END_IF;
14     END_CASE;
15     ROBOT Y:= ROBOT X;
16     325  COUNTER_1 := COUNTER_1+1;
17     325  IF COUNTER_1 >1000 THEN
18     325  COUNTER_1 :=-0;
19     END_IF;
    
```

Abbildung 141: ST-Arbeitsblatt im Online-Modus mit einem gesetztem Breakpoint

11.14 Verwenden des Watch-Fensters

Sie können im Watch-Fenster Variablen aus verschiedenen Arbeitsblättern im Online-Modus sammeln, um einen Überblick darüber zu bekommen, wie diese Variablen zusammenarbeiten. Nachdem Sie eine Variable einmal in das Watch-Fenster eingefügt haben, müssen Sie die Arbeitsblätter, in der diese Variable verwendet wird, nicht mehr öffnen, um den aktuellen Wert sehen zu können. Zusätzlich können Sie mit dem Watch-Fenster Elemente aus anwenderdefinierten Datentypen debuggen, wie z. B. Felder und Strukturen.



HINWEIS

Das Debuggen von anwenderdefinierten Datentypen mit Hilfe des Watch-Fensters wird im folgenden Abschnitt 'Debuggen von anwenderdefinierten Datentypen mit dem Watch-Fenster' beschrieben.

Sie können im Watch-Fenster beliebig viele Variablen einfügen und auch wieder löschen. Das Forcen und Überschreiben von Variablen ist hier ebenfalls möglich. Dies erreichen Sie, indem Sie aus dem Kontextmenü einer Variablen im Watch-Fenster den Menüpunkt 'Debug-Dialog' auswählen und im Online-Dialog die entsprechenden Einstellungen vornehmen.



HINWEIS

Wenn Sie den Debug-Dialog aus dem Watch-Fenster heraus öffnen, können Breakpoints für die markierte Variable nicht gesetzt bzw. rückgesetzt werden. Für diesen Fall müssen Sie den Debug-Dialog über das Kontextmenü der betreffenden Variable öffnen.

In der folgenden Beschreibung nehmen wir an, dass Sie eine Variable in das Watch-Fenster einfügen möchten. Um dies durchführen zu können, muss zuerst das Watch-Fenster geöffnet werden.



Aufrufen des Watch-Fensters mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Watch-Fenster'.
Abhängig von seinem vorherigen Status wird das Meldungsfenster jetzt ein- oder ausgeblendet.



Aufrufen des Watch-Fensters mit der Tastatur

- Drücken Sie <ALT> + <F10>.
Abhängig von seinem vorherigen Status wird das Meldungsfenster jetzt ein- oder ausgeblendet.

Die folgende Abbildung zeigt das Watch-Fenster.

Variable	Wert	Typ	Instanz
Level	750	INT	C_IPC.R_IPC.T_100ms.PRG_FBD.Level
Temp	34	INT	C_IPC.R_IPC.T_100ms.PRG_FBD.Temp
Intake_Q	FALSE	BOOL	C_IPC.R_IPC.T_100ms.PRG_FBD.Intake_Q
Heater_Q	FALSE	BOOL	C_IPC.R_IPC.T_100ms.PRG_FBD.Heater_Q
Drain_Q	TRUE	BOOL	C_IPC.R_IPC.T_100ms.PRG_FBD.Drain_Q
Container States		Container_Struct_Array	C_IPC.R_IPC.T_100ms.PRG_FBD.Container_States

Watch 1 | Watch 2 | Watch 3 | Watch 4

Abbildung 142: Watch-Fenster mit eingefügten Variablen

Das Watch-Fenster enthält die folgenden Informationen:

- * 'Variable': Zeigt den Variablennamen. Anwenderdefinierte Datentypen wie z. B. Felder und Strukturen erkennen Sie an einem '+'-Zeichen vor dem Variablennamen (z. B. 'Container States' im oben abgebildeten Watch-Fenster). Die Elemente dieser Datentypen zeigen Sie an, indem Sie auf das '+'-Zeichen klicken. Normale Variablen werden wie z. B. Variable 'Level' im oben abgebildeten Watch-Fenster angezeigt.
- * 'Wert': Zeigt den aktuellen Wert der Variablen.
- * 'Typ': Zeigt den Datentyp.
- * 'Instanz': Zeigt den Pfad der Instanz, in der die Variable verwendet wird. Der Pfad besteht immer aus der Konfiguration, der Ressource, der Task, dem zugewiesenen Programmnamen und dem Variablennamen.

Sie können im Watch-Fenster mehrere Seiten unabhängig voneinander verwalten. Die einzelnen Seiten zeigen Sie an, indem Sie auf die Reiter des Watch-Fensters klicken.



Einfügen von Variablen in das Watch-Fenster mit der Maus

- Markieren Sie die Variable im Arbeitsblatt im Online-Modus.
 - Klicken Sie mit der rechten Maustaste, um das Kontextmenü zu öffnen.
 - Wählen Sie den Menüpunkt 'In Watch-Fenster einfügen'.
- Die Variable wird in das Watch-Fenster eingefügt.

Um Variablen im Watch-Fenster zu löschen, gehen Sie entsprechend der folgenden Beschreibung vor.



Löschen von Variablen im Watch-Fenster mit der Maus

Sie können entweder einzelne Variablen oder alle Variablen im Watch-Fenster löschen.

- * Um **eine einzelne Variable in der Liste zu löschen**, markieren Sie die gewünschte Variable und drücken die Taste <ENTF>. Sie können die Variable auch löschen, indem Sie das Kontextmenü der Variablen öffnen (durch Klicken mit der rechten Maustaste) und anschließend den Menüpunkt 'Löschen' auswählen. Dies ist auch für ein einzelnes Element innerhalb einer anwenderdefinierten Struktur oder eines Feldes möglich.
- * Um **alle Variablen** aus dem Watch-Fenster zu löschen, öffnen Sie im Watch-Fenster das Kontextmenü einer beliebigen Variablen und wählen den Menüpunkt 'Alles löschen'.

11.15 Debuggen von anwenderdefinierten Datentypen mit dem Watch-Fenster

Sie können nur im Watch-Fenster anwenderdefinierte Datentypen wie z. B. Felder und Strukturen debuggen. Um dies zu ermöglichen müssen Sie zuerst das Watch-Fenster aufrufen und anschließend die Variable einfügen, die als Feld oder Struktur definiert ist. Anschließend können Sie die Elemente des Felds oder der Struktur im Watch-Fenster debuggen.

In den folgenden Schritten nehmen wir an, dass Sie ein Feld einer Struktur debuggen möchten. Dazu müssen Sie zuerst das Feld der Struktur in das Watch-Fenster entsprechend der Beschreibung im vorhergehenden Abschnitt einfügen. Abbildung 11-18 zeigt, wie ein Feld einer Struktur (im Beispiel 'Container States') im Watch-Fenster dargestellt wird. Nachdem Sie diesen Schritt durchgeführt haben, müssen die Elemente des Felds der Struktur im Watch-Fenster angezeigt werden.



Debuggen eines Felds einer Struktur mit der Maus

- Klicken Sie im Watch-Fenster auf das '+'-Zeichen, das vor dem Namen des Felds der Struktur erscheint.

+	Container States
---	------------------

Es werden die Elemente des Felds der Struktur angezeigt.

- Klicken Sie auf das '+'-Zeichen, das vor dem Namen der Struktur (z. B. Struktur [0]) erscheint.

+	Container States
+	[0]
+	[1]

Sie können jetzt die Elemente der Struktur sehen (siehe hierzu die folgende Abbildung).

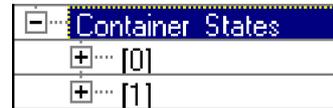


Debuggen eines Felds einer Struktur mit der Tastatur

- Drücken Sie im Watch-Fenster <↑> oder <↓>, um in die Zeile mit dem Feld der Struktur zu gelangen.



Drücken Sie <→>, um die Elemente des Felds der Struktur anzuzeigen.



- Drücken Sie <↓>, um eine Struktur zu markieren (z. B. Struktur [0]) und drücken Sie anschließend <→>, um die Elemente der Struktur anzuzeigen.

Sie können jetzt die Elemente der Struktur sehen (siehe hierzu die folgende Abbildung).

Variable	Wert	Typ	Instanz
Container States		Container_Struct_Array	C_IPC.R_IPC...
[0]		Container_Struct	C_IPC.R_IPC...
Drain	FALSE	BOOL	C_IPC.R_IPC...
Heater	TRUE	BOOL	C_IPC.R_IPC...
Intake	FALSE	BOOL	C_IPC.R_IPC...
Level	650	INT	C_IPC.R_IPC...
Temp	30	INT	C_IPC.R_IPC...
State of container	Intake: CLOSE Heater: ON Drain: CLOSE	STRING	C_IPC.R_IPC...
Values of container	Level: 650 Liter Temperature: 30 Grad	STRING	C_IPC.R_IPC...
[1]		Container_Struct	C_IPC.R_IPC...

Abbildung 143: Watch-Fenster mit Elementen des Felds einer Struktur

HINWEIS



Sie können die Elemente von anwenderdefinierten Datentypen im Watch-Fenster forcen und überschreiben sowie für die Variablen Breakpoints setzen und rücksetzen. Dies erreichen Sie, indem Sie aus dem Kontextmenü einer Variablen im Watch-Fenster den Menüpunkt 'Debug-Dialog...' auswählen und im Debug-Dialog die entsprechenden Einstellungen vornehmen.



Das Forcen und Überschreiben von Variablen und das Setzen und Rücksetzen von Breakpoints ist in den Abschnitten 'Forcen und Überschreiben von Variablen' und 'Setzen und Rücksetzen von Breakpoints' beschrieben.

11.16 Arbeiten mit der Logikanalyse

Die Logikanalyse ist ein leistungsfähiges Werkzeug zum Aufzeichnen von Variablenwerten innerhalb eines bestimmten Zeitintervalls, das vom Benutzer durch die Eingabe von Abtastzyklen festgelegt wird.

Mit der Logikanalyse können Sie das Verhalten Ihres SPS-Programms prüfen, indem Sie Variablen aufzeichnen (während die SPS läuft) und diese grafisch im Fenster der Logikanalyse anzeigen. Sie haben die Möglichkeit mehrere Kurven von Variablenwerten in Abhängigkeit der eingestellten Triggerbedingungen und weiteren Einstellungen anzuzeigen.

Alle aufgezeichneten Werte und Einstellungen für die Logikanalyse werden automatisch gemeinsam mit dem Projekt gespeichert.

Das System unterstützt die Logikanalyse in Verbindung mit einer oder mehreren Ressourcen. Dabei wird jeder Ressource ein getrenntes Logikanalyse-Arbeitsblatt zugewiesen. Eine neue Ressource kann nach dem Kompilieren des Projekts in das Fenster der Logikanalyse eingefügt bzw. im Fenster gelöscht werden.

HINWEIS



Das Aufzeichnen von Werten ist nur im Online-Modus möglich.



Aufrufen des Logikanalyse-Fensters mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Logikanalyse'.
Abhängig von seinem vorherigen Status wird das Logikanalyse-Fenster jetzt ein- oder ausgeblendet.



Aufrufen des Logikanalyse-Fensters mit der Tastatur

- Drücken Sie <ALT> + <F11>.
Abhängig von seinem vorherigen Status wird das Logikanalyse-Fenster jetzt ein- oder ausgeblendet.

Nach **dem ersten Öffnen** des Fensters enthält das Fenster noch keine Werte, da noch keine Variablen eingefügt wurden.

Die folgende Abbildung zeigt das Logikanalyse-Fenster mit den erfassten Kurven für die Variablen `Level` und `Temp`.

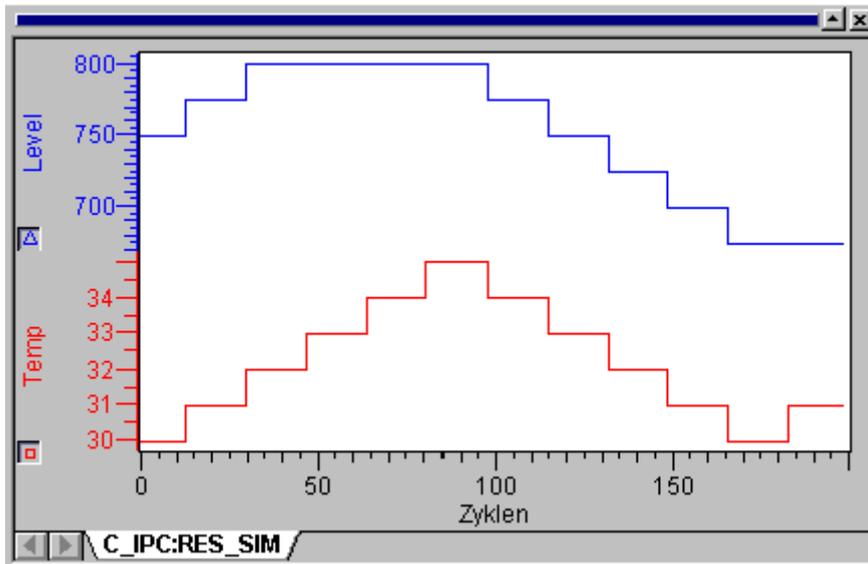
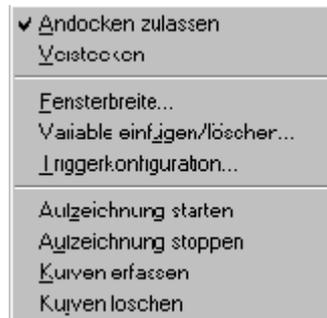


Abbildung 144: Logikanalyse mit zwei erfassten Kurven

Sie können im Logikanalyse-Fenster das Kontextmenü öffnen, indem Sie mit der rechten Maustaste auf den unteren Rand des Fensters klicken. Das Kontextmenü enthält alle verfügbaren Befehle und Dialoge für die Logikanalyse. Diese sind auch als Symbole in der Symbolleiste aufgeführt. In den folgenden Beschreibungen werden die Symbole in der Symbolleiste verwendet.



Die folgende Vorgehensweise beschreibt das Arbeiten mit der Logikanalyse, d. h. wie Variablen eingefügt werden, die Triggerkonfiguration eingestellt wird, die Aufzeichnung gestartet wird, usw.



Weitere detaillierte Informationen hierzu finden Sie in der allgemeinen SPS-Hilfe.



HINWEIS

Voraussetzungen für das Arbeiten mit der Logikanalyse:

Stellen Sie vor dem Aufrufen der Logikanalyse sicher, dass

- das kompilierte Projekt an die SPS gesendet wurde.
- die SPS oder Simulation in Betrieb ist (läuft).
- das System im Online-Modus ist.

Nachdem Sie das Logikanalyse-Fenster geöffnet haben, müssen Sie zuerst Variablen, für die Werte aufgezeichnet werden sollen, in die Logikanalyse einfügen. Nach dem ersten Starten der Logikanalyse enthält diese keine Variablen. Um eine Variable in die Logikanalyse einzufügen, gehen Sie wie folgt vor:



Einfügen von Variablen in die Logikanalyse

- Stellen Sie sicher, dass das System im Online-Modus ist.
- Wählen Sie in einem beliebigen textuellen oder grafischen Code-Arbeitsblatt bzw. in einem Variablen-Arbeitsblatt eine Variable aus, indem Sie diese mit der rechten Maustaste anklicken. Sie öffnen damit das Kontextmenü.
- Wählen Sie den Menüpunkt 'In Logikanalyse einfügen'.
- Die markierte Variable wird in das Fenster der Logikanalyse eingefügt.
- Wiederholen Sie diese Schritte für alle Variablen, die Sie in die Logikanalyse einfügen möchten.



Löschen von Variablen in der Logikanalyse

Es kann vorkommen, dass eine oder mehrere Variablen in der Logikanalyse gelöscht werden müssen, da diese nicht mehr länger benötigt werden. Für diesen Fall muss der Dialog 'Variablen' der Logikanalyse aufgerufen werden.

- Klicken Sie in der Symbolleiste auf das Symbol 'Einfügen/Löschen von Variablen'.
Es erscheint der Dialog 'Variablen'. Dieser enthält eine Liste aller Variablen, die mit der Logikanalyse verbunden sind.

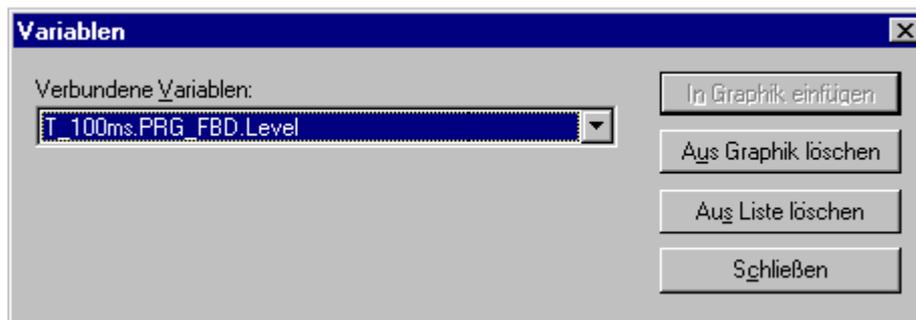


Abbildung 145: Dialog 'Variablen' der Logikanalyse

Im Dialog 'Variablen' kann eine Variable auf zwei verschiedene Arten in der Logikanalyse gelöscht werden:

* **Löschen einer Variablen im Logikanalyse-Fenster:**

In diesem Fall erscheint die Variable nicht mehr im Logikanalyse-Fenster, ist aber immer noch in der Variablenliste vorhanden. Sie können diese gelöschte Variable erneut in die Logikanalyse einfügen, indem Sie aus dem Listenfeld des Dialogs 'Variablen' die Variable auswählen und die Schaltfläche 'In Graphik einfügen' drücken.

* **Löschen einer Variablen aus der Variablenliste der Logikanalyse:**

Hier wird die Variable sowohl in der Grafik als auch in der Variablenliste der Logikanalyse gelöscht. Wenn Sie anschließend die Variable erneut in der Logikanalyse verwenden möchten, müssen Sie diese über das Kontextmenü der Variablen wieder einfügen.



Einstellen der Breite des Datenfensters für die Logikanalyse

Sie haben jetzt die gewünschten Variablen in die Logikanalyse eingefügt. Im nächsten Schritt müssen Sie die Breite des Datenfensters für die Logikanalyse einstellen, um die x-Achse zu definieren.

- Öffnen Sie im Logikanalyse-Fenster das Kontextmenü und wählen den Menüpunkt 'Fensterbreite...'.
Es erscheint der Dialog 'Datenfenster'.



Abbildung 146: Dialog 'Datenfenster' der Logikanalyse

- Geben Sie in das Textfeld die Anfangsbreite des Fensters ein. Der Standardwert ist 200 Abtastzyklen.



Einstellen der Triggerkonfiguration

Ein wichtiger Schritt vor dem Aufzeichnen von Werten mit der Logikanalyse ist das Einstellen der Triggerkonfiguration.

- Klicken Sie in der Symbolleiste auf das Symbol 'Triggerkonfiguration'.
Es erscheint der Dialog 'Triggerkonfiguration'.



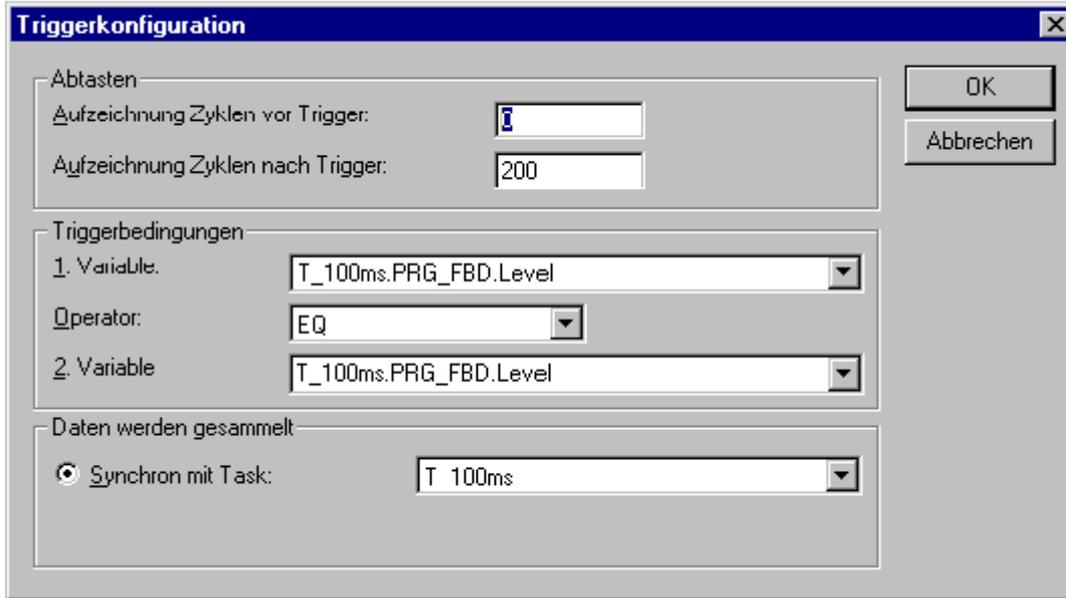


Abbildung 147: Dialog 'Triggerkonfiguration' der Logikanalyse



Detaillierte Informationen zur Bedeutung der einzelnen Dialogfelder und Schaltflächen finden Sie in den entsprechenden Themen der allgemeinen SPS-Hilfe.

- Im Bereich **Abtasten** legen Sie den Zeitraum fest, in dem die Logikanalyse Werte aufzeichnet. Dieses Intervall wird mit Hilfe der Anzahl von Abtastzyklen definiert. Geben Sie die Anzahl der Abtastzyklen vor und nach dem Triggerereignis an, d. h. in welchem Zeitraum die Logikanalyse Werte aufzeichnen soll.
- Wählen Sie im Bereich **Triggerbedingungen** die '1. Variable' und '2. Variable' sowie den 'Operator', der die Triggerbedingung in Abhängigkeit von der ersten und zweiten Variable festlegt.

HINWEIS



Achten Sie beim Einstellen der Triggerkonfiguration darauf, dass die Variablen und der Operator so ausgewählt werden, dass die Triggerbedingung erfüllt werden kann. Die Logikanalyse kann nur dann Werte aufzeichnen, wenn die Triggerbedingung erfüllt wird. Ignorieren Sie in diesem Fall die Meldung 'Aufzeichnen von Werten...'.
Beispiel: Mit der Einstellung '1.Variable' = '2. Variable' und 'Operator' = NE wird die Triggerbedingung **niemals** erfüllt.

- Wählen Sie im Bereich **Daten werden gesammelt** eine Task aus. Die Logikanalyse zeichnet Werte synchron mit der in diesem Listenfeld ausgewählten Task auf. Deshalb ist es wichtig, dass die Triggerbedingungen und die ausgewählte Task sich aufeinander beziehen.



Starten der Aufzeichnung von Werten

Nachdem die Triggerbedingungen eingestellt wurden, können Sie Werte für die Variablen aufzeichnen, die mit der Logikanalyse verbunden sind.

- Klicken Sie in der Symbolleiste auf das Symbol 'Aufzeichnung starten'. Die Logikanalyse startet mit der Aufzeichnung der Werte und führt diese solange fort, bis das Zeitintervall, das im Dialog 'Triggerkonfiguration' eingestellt ist, abgelaufen ist. Bei einer fehlerfreien Aufzeichnung wird der Prozess automatisch nach der angegebenen Anzahl von Abtastzyklen gestoppt.



Abbrechen der Aufzeichnung von Werten

Angenommen Sie haben einen falschen Wert für die Abtastzyklen eingegeben oder einen anderen Parameter nicht korrekt eingestellt, können Sie die Aufzeichnung stoppen, bevor das definierte Zeitintervall abgelaufen ist.

- Um das Aufzeichnen zu stoppen, klicken Sie in der Symbolleiste auf das Symbol 'Aufzeichnung stoppen'.



Einstellen der Anzeige im Logikanalyse-Fenster

Nach dem Aufzeichnen einer oder mehrerer Variablen werden die Kurven automatisch im Fenster der Logikanalyse angezeigt. Sie haben hier mehrere Möglichkeiten die grafische Anzeige der Kurven zu verändern.

Dehnen oder komprimieren der vertikalen bzw. horizontalen Achse

Sie können die horizontale und vertikale Achse für jede Variable dehnen und komprimieren, d. h. auseinanderziehen und zusammenschieben, um die Anzeige zu vergrößern bzw. zu verkleinern. Dazu

- klicken Sie mit der linken oder rechten Maustaste auf die gewünschte Achse, halten die Maustaste gedrückt und bewegen die Maus in die gewünschte Richtung.

Verschieben des sichtbaren Bereichs der vertikalen oder horizontalen Achse

Sie können zusätzlich den sichtbaren Bereich im Fenster verschieben. Dies ist dann nützlich, wenn Sie die Achse gedehnt, d. h. die Anzeige vergrößert haben und dadurch die benötigten Werte nicht mehr im sichtbaren Bereich liegen. Dazu

- klicken Sie gleichzeitig mit der linken und rechten Maustaste auf die gewünschte Achse, halten die Maustasten gedrückt und bewegen die Maus, um den sichtbaren Bereich zu verschieben.

HINWEIS



Ein Verschieben des sichtbaren Bereichs ist nur dann möglich, wenn der Bereich größer als der angezeigte Bereich ist.



Auslesen von bestimmten Werten in der grafischen Darstellung der Logikanalyse

Mit Hilfe einer horizontalen oder vertikalen Cursor-Linie können Sie den genauen Wert einer Variablen auslesen.

Dazu

- bewegen Sie den Mauszeiger an den oberen/unteren Rand (um eine horizontale Linie zu erhalten) oder an den linken/rechten Rand (um eine vertikale Linie zu erhalten). Unter dem Mauszeiger erscheint eine symbolische Doppellinie.
- Klicken Sie mit der linken Maustaste, halten diese gedrückt und verschieben die Cursor-Linie zur Signalkurve.

Nachdem Sie die Cursor-Linie an den gewünschten Punkt im Fenster verschoben haben, erscheint ein kleines Informationsfenster, das die genauen Werte für diesen Punkt in der Kurve zeigt.

Wenn Sie eine **vertikale Cursor-Linie** verwenden, zeigt das Informationsfenster die Werte aller im Logikanalyse-Fenster eingefügten Variablen in Bezug zur Zyklanzahl. Das hier gezeigte Beispiel zeigt die Logikanalyse mit den zwei eingefügten Variablen Temp und Level.

[Zyklen]	: 118.3
△ Temp	: 4.00
□ Level	: 100.0

Wenn Sie eine **horizontale Linie** verwenden, zeigt das Informationsfenster die Werte aller im Logikanalyse-Fenster eingefügten Variablen ohne deren Bezug zur x-Achse, wie hier zu sehen ist.

[Temp]	: 29.897
[Level]	: 570.1



Löschen der Kurven im Logikanalyse-Fenster

Um in der Logikanalyse alle Kurven und aufgeführten Variablen in einem Schritt zu löschen, gehen Sie wie folgt vor:

- Öffnen Sie das Kontextmenü der Logikanalyse.
- Wählen Sie den Menüpunkt 'Kurven löschen'.

Damit Sie anschließend wieder eine Aufzeichnung durchführen können, müssen Sie erneut Variablen in die Logikanalyse einfügen.

DRUCKEN DER PROJEKTE MIT EIGENEM SEITENLAYOUT

12.1 Drucken des Projekts

Das Untermenü 'Datei' enthält alle Befehle zum Definieren der Druckereinstellungen, zum Aufrufen der Seitenansicht für die aktuelle Seite und zum Drucken des gesamten Projekts oder von Teilen daraus.

Sie haben mehrere Möglichkeiten Ihre Projektdokumentation zu drucken. Zum Einstellen der Druckoptionen gibt es zwei Dialoge:

- * Im Dialog 'Projekt drucken' wählen Sie aus, welche Teile des Projekts gedruckt werden sollen und stellen den Druckmodus ein.
- * Auf der Seite 'Standardseitenlayouts' des Dialogs 'Optionen' können Sie das Seitenlayout auswählen, mit dem die Daten gedruckt werden.

Diese beiden Dialoge und die verschiedenen Druckmodi sind in den folgenden Abschnitten beschrieben.

12.1.1 Steuern des Druckprozesses mit dem Dialog 'Projekt drucken'

Falls Sie nur einzelne Teile Ihres Projektes drucken wollen, müssen Sie vor dem Aufruf des Dialogs 'Projekt drucken' das gewünschte Ordnersymbol (z. B. ein POE-Symbol) im Projektbaum markieren.



HINWEIS

Bibliotheken und Arbeitsblätter im Online-Modus können nicht gedruckt werden.



Auswählen des gewünschten Ordnersymbols (Knoten) im Projektbaum

- Markieren Sie im Projektbaum die gewünschten Ordner-Symbole, deren Inhalte Sie drucken wollen. Sie können hier ein oder mehrere Arbeitsblätter (siehe folgender Hinweis), eine einzelne POE oder den gesamten Unterbaum 'Logische POEs' markieren. Wenn Sie das gesamte Projekt drucken wollen, überspringen Sie diesen Schritt und fahren mit dem nächsten Schritt fort.



HINWEIS

So markieren Sie mehrere benachbarte Objekte im Projektbaum:

Markieren Sie das erste gewünschte Objekt im Projektbaum, indem Sie es mit der linken Maustaste anklicken.

Drücken Sie anschließend die <UMSCHALT>-Taste und halten Sie diese gedrückt. Klicken Sie auf das letzte gewünschte Objekt und lassen Sie dann die <UMSCHALT>-Taste los. Alle Objekte zwischen dem zuerst und dem zuletzt angeklickten Symbol sind nun markiert.

So markieren Sie mehrere nicht benachbarte Objekte:

Markieren Sie das erste gewünschte Objekt im Projektbaum, indem Sie es mit der linken Maustaste anklicken. Drücken Sie die <STRG>-Taste und halten Sie diese gedrückt. Klicken Sie nacheinander auf jedes gewünschte Objekt.



Anzeigen der Seitenränder

Vor dem Drucken empfiehlt es sich die Seitenränder der zu druckenden Arbeitsblätter einzublenden. Die Seitenränder werden als blaue Linien dargestellt.

Speziell bei graphischen Arbeitsblättern können Sie durch Anzeigen der Ränder prüfen, ob der gesamte Inhalt des Arbeitsblattes auf die Seite passt. Unter Umständen müssen Sie dazu die Ansicht des Arbeitsblattes verkleinern, so dass das gesamte Arbeitsblatt und die Seitenränder sichtbar werden.

- Wählen Sie den Menüpunkt 'Ränder' aus dem Untermenü 'Layout'.
Beachten Sie, dass die Seitenränder für jedes Arbeitsblatt einzeln angezeigt werden müssen.



Aufrufen des Dialogs 'Projekt drucken'

- Wählen Sie aus dem Untermenü 'Datei' den Menüpunkt 'Projekt drucken...'. Es erscheint der Dialog 'Projekt drucken', den Sie in der folgenden Abbildung sehen können.

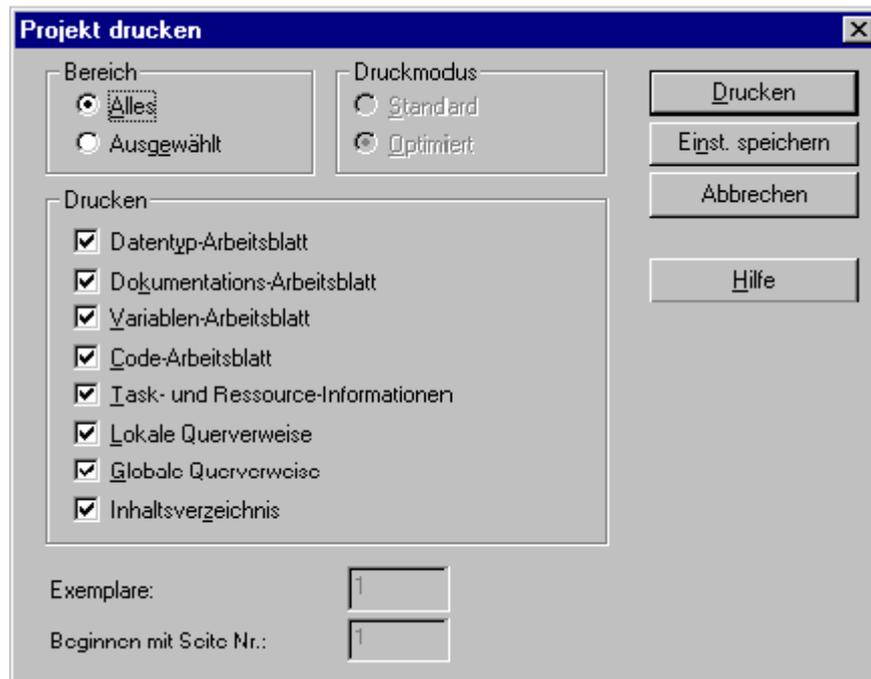


Abbildung 148: Dialog 'Projekt drucken'



Arbeiten mit dem Dialog 'Projekt drucken'

- Wählen Sie den zu druckenden Bereich aus.
 - 'Alles' bedeutet, dass das gesamte Projekt, einschließlich aller Unterbäume gedruckt wird.
 - 'Ausgewählt' bedeutet, dass nur diejenigen Symbole gedruckt werden, die Sie vor dem Aufruf des Dialogs 'Projekt drucken' im Projektbaum markiert haben.
- Wählen Sie die zu druckenden Punkte aus, indem Sie das zugehörige Kontrollkästchen im Bereich 'Drucken' aktivieren/deaktivieren. Beachten Sie dabei, dass sich jedes Kontrollkästchen nur auf den eingestellten Bereich (d. h. 'Alle' oder 'Ausgewählt') bezieht. Beispiel: Wenn Sie die Kontrollkästchen 'Dokumentations-Arbeitsblatt', 'Variablen-Arbeitsblatt' und 'Code-Arbeitsblatt' markiert haben, werden nur die Arbeitsblätter gedruckt, die innerhalb des gewählten Bereichs liegen.
- Stellen Sie den gewünschten **Druckmodus** ein.
 - '**Standard**' bedeutet, dass jeder ausgewählte Punkt mit dem zugeordneten Standardseitenlayout gedruckt wird.
Um das Standardseitenlayout für den Standarddruckmodus einzustellen, öffnen Sie den Dialog 'Optionen', blättern auf die Seite 'Standardseitenlayouts' und editieren dort die Felder 'Text' und 'Graphik'
 - '**Optimiert**' bedeutet, dass die ausgewählten Punkte mit so wenig Papier wie möglich ausgedruckt werden. Dabei werden die einzelnen Arbeitsblätter direkt hintereinander gedruckt. Es wird nur dann eine neue Seite verwendet, wenn eine neue POE oder ein neuer AS-Schritt beginnt. Beim optimierten Drucken können ein Inhaltsverzeichnis und Querverweise gedruckt werden.

Das Standardseitenlayout für optimiertes Drucken stellen Sie im Dialog 'Optionen' auf der Seite 'Standardseitenlayouts' ein (Listenfeld 'Optimiertes Drucken').



Die notwendigen Schritte, um ein bestimmtes Seitenlayout als Standardseitenlayout für die verschiedenen Druckmodi zu definieren, sind im Abschnitt 'Definieren eines Seitenlayouts als Standardseitenlayout' in diesem Kapitel beschrieben.



Die Vorgehensweise zum Erstellen und Ändern eines Seitenlayouts sind im Abschnitt 'Editieren eines Seitenlayouts' in diesem Kapitel beschrieben.



- Falls gewünscht, ändern Sie die Anzahl der zu druckenden Exemplare.
- Geben Sie bei Bedarf eine andere Seitennummer für die erste zu druckende Seite ein.



HINWEIS

Die Anzahl der zu druckenden Exemplare und die erste Seitennummer können in der aktuellen Systemversion nicht verändert werden.



- Bestätigen Sie den Dialog.
Die ausgewählten Daten werden auf dem Standarddrucker gedruckt.

12.1.2 Definieren eines Seitenlayouts als Standardseitenlayout

Sie können ein Seitenlayout (z. B. ein mit dem Seitenlayout-Editor selbst erstelltes Layout) als Standardseitenlayout definieren. Dieses Standardseitenlayout wird immer verwendet, wenn Sie ein Projekt (oder Teile davon) ausdrucken. Das Standardseitenlayout wird im Dialog 'Optionen' eingestellt.



Aufrufen der Seite 'Seitenlayouts' im Dialog 'Optionen'

- Wählen Sie aus dem Untermenü 'Extras' den Menüpunkt 'Optionen'.
Es erscheint der Dialog 'Optionen'.
- Klicken Sie auf den Reiter 'Seitenlayouts'.

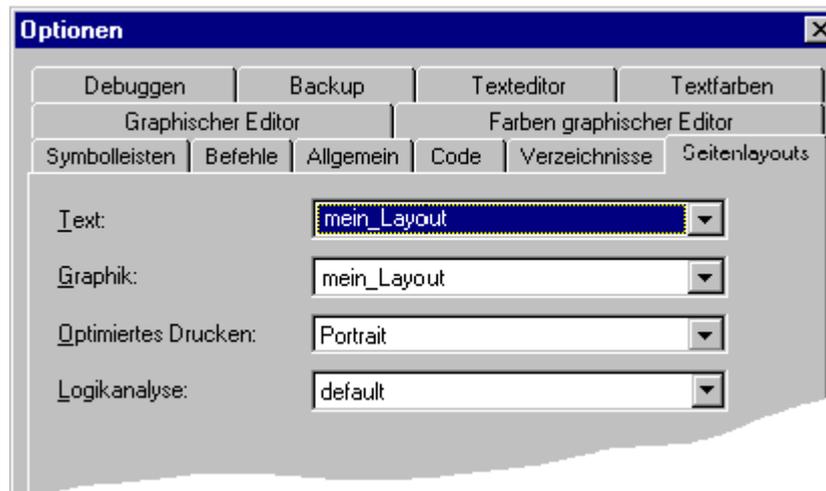


Abbildung 149: Seite 'Seitenlayouts' im Dialog 'Optionen'



Verwenden der Seite 'Seitenlayouts'

Auf dieser Seite befinden sich vier Listenfelder, in denen Sie verschiedene Standardseitenlayouts definieren können. In jedem Listenfeld werden alle Seitenlayouts zur Auswahl angeboten, die im Verzeichnis für Seitenlayouts gespeichert sind. Dabei kann es sich entweder um vordefinierte Layouts handeln (die bereits mit dem Programm geliefert wurden) oder um selbst erstellte Layouts, die Sie mit Hilfe des Seitenlayout-Editors erstellt haben.

In [Abbildung 149](#) auf Seite 219 wurde das selbst erstellte Seitenlayout 'Mein_Layout' als Standardseitenlayout für Text und Graphik im Druckmodus 'Standard' definiert. Für den Druckmodus 'Optimiert' wurde dagegen das vordefinierte Layout 'Portrait' als Standardlayout gewählt. Für die Logikanalyse ist das Standardlayout eingestellt.

- Öffnen Sie jedes Listenfeld und wählen Sie das gewünschte Seitenlayout aus.
 - **'Text'**: Hier definieren Sie das Layout für Textarbeitsblätter, wie z. B. Dokumentations-Arbeitsblätter, Variablen-Arbeitsblätter, textuelle Code-Arbeitsblätter, Querverweise, etc. Das eingestellte Layout im Feld 'Text' wird nur im Druckmodus 'Standard' verwendet.
 - **'Graphik'**: Hier definieren Sie das Seitenlayout für graphische Arbeitsblätter, wie z. B. Programmcodes in den Sprachen FBS, KOP oder AS. Das im Feld 'Graphik' eingestellte Seitenlayout wird nur verwendet, wenn Sie im Modus 'Standard' drucken.
 - **'Optimiertes Drucken'**: In diesem Feld definieren Sie das Seitenlayout für Text und Graphik im Druckmodus 'Optimiert'.
 - **'Logikanalyse'**: Hier definieren Sie das Seitenlayout für die zu druckenden Informationen der Logikanalyse.
- Bestätigen Sie den Dialog.

12.2 Arbeiten mit dem Seitenlayout-Editor

Der Seitenlayout-Editor ist ein Programm, mit dem Sie neue Seitenlayouts erstellen oder bestehende Layouts verändern können.

Nachdem Sie ein neues Seitenlayout gespeichert haben, befindet sich dieses in einem eigens dafür vorgesehenen Seitenlayout-Verzeichnis und kann im Dialog 'Optionen' als Standardseitenlayout definiert werden.

Bevor Sie Ihr eigenes Seitenlayout editieren können, müssen Sie den Seitenlayout-Editor aufrufen.



Aufrufen des Seitenlayout-Editors

- Wählen Sie aus dem Untermenü 'Extras' den Menüpunkt 'Seitenlayout-Editor'.
Der Seitenlayout-Editor wird geöffnet.



HINWEIS

Der Seitenlayout-Editor wird normalerweise mit dem Standardseitenlayout geöffnet.

12.2.1 Erstellen eines neuen Seitenlayouts

Erstellen Sie ein neues Seitenlayout mit folgenden Schritten:



Erstellen eines neuen Seitenlayouts mit der Maus

- Wählen Sie aus dem Untermenü 'Datei' den Menüpunkt 'Neu'.
Der Seitenlayout-Editor wird mit dem Seitenlayout 'untitled' (Unbenannt) geöffnet.



Erstellen eines neuen Seitenlayouts mit der Tastatur

- Drücken Sie <ALT> + <D>.
Das Untermenü 'Datei' wird geöffnet.
- Drücken Sie <N>.
Der Seitenlayout-Editor wird mit dem Seitenlayout 'untitled' (Unbenannt) geöffnet.

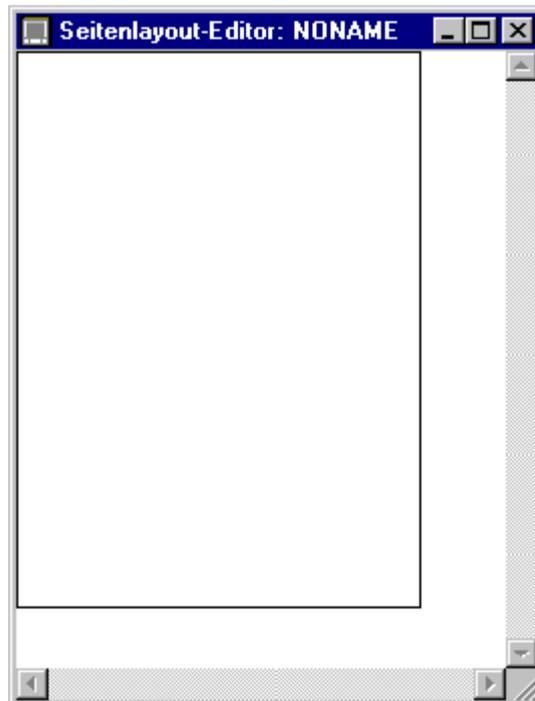


Abbildung 150: Seitenlayout-Editor mit einem neuen Seitenlayout

12.2.2 Definieren des Codebereichs

Nachdem Sie ein neues Seitenlayout erstellt haben, müssen Sie zunächst den Bereich für den Inhalt der Arbeitsblätter festlegen. Dieser Bereich heißt 'Codebereich' und wird durch ein rotes Rechteck dargestellt.



Definieren des Codebereichs mit der Maus

- Klicken Sie in der Symbolleiste mit der linken Maustaste auf das Symbol 'Codebereich'.
- Unter dem Mauszeiger erscheint ein symbolisches Rechteck.
- Setzen Sie den Mauszeiger an die Stelle, an der Sie eine Ecke des Codebereiches platzieren möchten.
- Drücken Sie die linke Maustaste und halten Sie diese gedrückt.
- Ziehen Sie mit der Maus ein Rechteck auf.
- Lassen Sie die Maustaste los.
Der Codebereich ist definiert.



Definieren des Codebereichs mit der Tastatur

- Drücken Sie <S>.
Unter dem Mauszeiger erscheint ein symbolisches Rechteck.



- Gehen Sie durch Drücken der Cursor-Tasten an die Stelle, an der Sie eine Ecke des Codebereiches platzieren möchten.
- Drücken Sie die <LEERTASTE> und halten sie diese gedrückt.
- Gehen Sie mit den Cursortasten an die gewünschte Position für die diagonal gegenüberliegende Ecke des Rechtecks.
- Lassen Sie die <LEERTASTE> los.
Der Codebereich ist definiert.

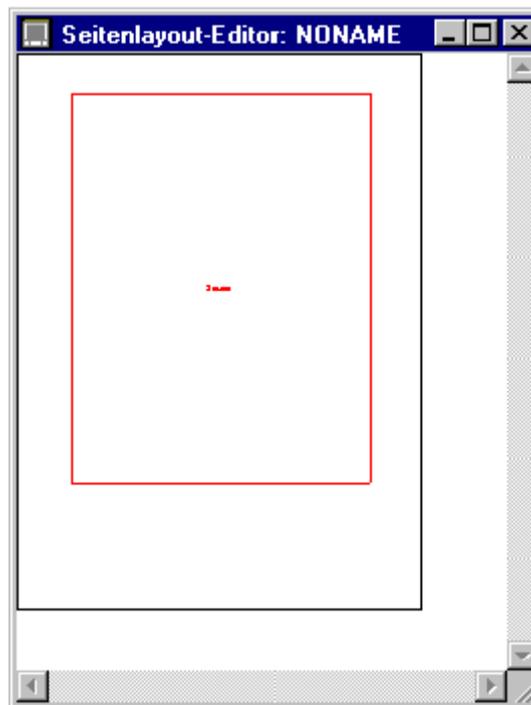


Abbildung 151: Seitenlayout mit dem neuen Codebereich



HINWEIS

Sie können den Codebereich auf der Seite verschieben, indem Sie ihn mit der linken Maustaste markieren und die Maustaste gedrückt halten. Bewegen Sie das Rechteck durch Ziehen der Maus an die gewünschte Position.

Um die Größe eines bereits definierten Codebereiches zu ändern, klicken Sie den Codebereich mit der rechten Maustaste an. Wählen Sie im Kontextmenü den Menüpunkt 'Objekteigenschaften...'. Es erscheint der Dialog 'Codebereich einrichten'. In diesem Dialog können Sie die Position, die Größe und andere Eigenschaften des Codebereichs einstellen.

12.2.3 Einfügen von Elementen in ein Seitenlayout

Sie können Elemente, wie z. B. Rechtecke, Linien oder Texte, in Ihr Seitenlayout einfügen. In der folgenden Beschreibung nehmen wir an, dass Sie eine horizontale Linie auf der Seite einfügen wollen.



Einfügen einer Linie mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Linie'.
Unter dem Mauszeiger erscheint eine symbolische Linie.
- Drücken Sie die linke Maustaste und halten Sie diese gedrückt.
- Ziehen Sie mit der Maus eine Linie.
- Lassen Sie die Maustaste los.
Die Linie ist gezogen.



Einfügen einer Linie mit der Tastatur

- Drücken Sie <I>.
Unter dem Mauszeiger erscheint eine symbolische Linie.
- Drücken Sie die <LEERTASTE> und halten sie diese gedrückt.
- Gehen Sie mit den Cursorstasten an die gewünschte Position für das Ende der Linie.
- Lassen Sie die <LEERTASTE> los.
Die Linie ist gezogen.



HINWEIS

Eine Beschreibung, wie Sie Rechtecke, Texte und Bitmaps in das Seitenlayout einfügen können, finden Sie in der kontextsensitiven Hilfe.

Die Bitmaps müssen als 16-Farben-Bitmaps vorliegen!

12.2.4 Editieren von Umgebungselementen

Weitere Objekte, die in ein Seitenlayout eingefügt werden können, sind Umgebungselemente. Umgebungselemente sind Platzhalter, beispielsweise für Seitenzahlen, den Firmennamen oder andere Texte. Es gibt mehrere Standard-Umgebungselemente, die 'System Items' genannt werden.



Editieren von Umgebungselementen mit der Maus

- Klicken Sie in der Symbolleiste auf das Symbol 'Umgebungselemente'.
Unter dem Mauszeiger erscheint ein 'T'.
- Klicken Sie mit der linken Maustaste auf die Stelle im Layout, an der Sie das Umgebungselement einfügen wollen.
Es erscheint der Dialog 'Umgebungstext einrichten'.





Editieren von Umgebungselementen mit der Tastatur

- Drücken Sie <V>. Unter dem Mauszeiger erscheint ein 'T'.
- Gehen Sie mit den Cursortasten an die Stelle im Layout, an der Sie das Umgebungselement einfügen wollen.
- Drücken Sie die <LEERTASTE>. Es erscheint der Dialog 'Umgebungstext einrichten'.



Einfügen eines Umgebungselements mit dem Dialog 'Umgebungstext einrichten'



- Wählen Sie das gewünschte Element aus.
- Klicken Sie auf die Schaltfläche 'Schrift...', um bei Bedarf eine neue Schriftart oder einen neuen Schriftgrad auszuwählen.
- Bestätigen Sie den Dialog. Der Platzhalter mit dem Namen des Elements wird in das Seitenlayout eingefügt.



HINWEIS

Die System-Umgebungselemente (System Items) und weitere Möglichkeiten zum Verändern oder Löschen vorhandener Elemente sind in der kontextsensitiven Hilfe beschrieben.

Sie können den Inhalt des Platzhalters über die Seitenansicht anschauen.

12.3 Arbeiten mit der Seitenansicht

Mit der Seitenansicht können Sie sich einen Eindruck vom Erscheinungsbild Ihres Arbeitsblatts nach dem Drucken verschaffen. Sie hilft Ihnen, die Elemente auf Ihrer Seite übersichtlich und strukturiert anzuordnen.

Mit der Seitenansicht können Sie den Inhalt des aktiven Fensters mit dem zugewiesenen Seitenlayout darstellen lassen. Wenn Sie kein Seitenlayout zugewiesen haben, wird der Inhalt mit dem Standardseitenlayout dargestellt.



HINWEIS

Querverweise werden in der Seitenansicht nicht angezeigt.

Die Seitenansicht kann nicht für Arbeitsblätter im Online-Modus verwendet werden.



Aufrufen der Seitenansicht mit der Maus

- Vergewissern Sie sich, dass das gewünschte Arbeitsblatt das aktive Fenster ist.
- Wählen Sie aus dem Untermenü 'Datei' den Menüpunkt 'Seitenansicht'.
Es erscheint die Seitenansicht mit dem aktuellen Arbeitsblatt.



Aufrufen der Seitenansicht mit der Tastatur

- Vergewissern Sie sich, dass das gewünschte Arbeitsblatt das aktive Fenster ist.
- Drücken Sie <ALT> + <D>.
Das Untermenü 'Datei' wird geöffnet.
- Drücken Sie <a>.
Es erscheint die Seitenansicht mit dem aktuellen Arbeitsblatt.

13

SYSTEMGRENZEN FÜR PROPROG WT II

13.0.1 Grenzen für Projekte

Knoten im Projektbaum	8000
Konfigurationen im Projektbaum	100
Ressourcen im Projektbaum	100
Programminstanzen pro Ressource	1000
Tasks pro Ressource	15
Programminstanzen pro Task	500
Globale Variablen	15000
Lokale Variablen pro POE	15000
Summe der Ein- und Ausgangsparameter von Funktionen und Funktionsbausteinen	128
Eingebundene Bibliotheken	60
POEs in einer Bibliothek	1000
POEs in einem Projekt	2000
Programmcode Arbeitsblätter pro POE	64
Variablen Arbeitsblätter pro POE	20
Globale Variablen Arbeitsblätter pro Ressource	20
Globale Programminstanz-Variablen Arbeitsblätter pro Ressource	20
Anzahl von Variablen pro Arbeitsblatt	2000 Zeilen und 64000 Zeichen, falls die Anzeige im Dialog erfolgen soll.

Summe Funktionen und Funktionsbausteine mit unterschiedlichem Typ je POE	620
Funktionen und Funktionsbausteine mit gleichem Typ je POE	1024
Sprünge und Marken pro POE	750
Sprünge und Returns pro Netzwerk	20
Schritte / Transitionen pro POE	110
Aktionen pro POE	165
Kontakte / Spulen pro POE	3000
Verschachtelung von Funktionen / Funktionsbausteinen	127
Verschachtelungstiefe für anwenderdefinierte Datentypen	8
Elemente in einer Struktur	200
Elemente in einem Feld	32766
Verschachtelungstiefe FOR / CASE / IF	8
I/O-Gruppen	200



HINWEIS

Zur besseren Strukturierung des Projekts sollten umfangreiche Variablendeklarationen nicht auf einem einzigen Arbeitsblatt stattfinden, sondern es sollte eine Verteilung auf mehrere Arbeitsblätter angestrebt werden. Prinzipiell ist die angegebene Anzahl von Variablen in einem Arbeitsblatt definierbar. Um jedoch bei der Deklaration von neuen Variablen die maximale Unterstützung des Variablen-Deklarationsdialogs zu erhalten, sollten pro Variablen-Arbeitsblatt nicht mehr als 5000 Zeilen und nicht mehr als 300000 Bytes verwendet werden.

Außerdem sollte je Zeile nur eine Variable deklariert werden, weil sonst keine eindeutige Anzeige im Online-Betrieb möglich ist.

13.0.2 Unterstützte Programmiersprachen

Textuelle Programmiersprachen	<ul style="list-style-type: none"> · Anweisungsliste (AWL) · Strukturierter Text (ST)
Grafische Programmiersprachen	<ul style="list-style-type: none"> · Funktionsbausteinsprache (FBS) · Kontaktplan (KOP) · Ablaufsprache (AS)

13.0.3 Datentypen

ANY_BIT	BOOL(1/8), BYTE(8), WORD(16), DWORD(32)
ANY_NUM	SINT(8), INT(16), DINT(32), USINT(8), UINT(16), UDINT(32), REAL(32) Die Grenzen der REAL-Größen liegen bei 3,4 E38 und -3,4 E38.
Zeit (TIME), Felder (ARRAY), Strukturen (STRUCT) und Zeichenfolgen (STRING)	

13.0.4 Implementationsbesonderheit anwenderdefinierter Datentypen

- Anwenderdefinierte Datentypen können innerhalb von Anwender-Funktionsbausteinen und -Programmen verwendet werden. Es ist nicht möglich, diese Datentypen in Anwender-Funktionen zu verwenden.
- Beim Debuggen von vollständigen Feldern und Strukturen müssen die Variablen in die Sammelliste geschrieben werden. Die Variablen können nur aus dem Variablen-Arbeitsblatt und nicht aus dem Programmcode-Arbeitsblatt in die Sammelliste geschrieben werden. Die Komponenten der Datentypen müssen in der Sammelliste geöffnet werden, wie in der kontextsensitiven Hilfe beschrieben.
- Multidimensionale Felder müssen deklariert werden. Es ist jedoch möglich, Felder von Feldern zu deklarieren, wie im folgenden Beispiel dargestellt:

```

Datentyp-Deklaration:

TYPE
    graph      :      ARRAY[1..10] OF INT;
    mein_feld  :      ARRAY[1..3]  OF graph;
END_TYPE

Variablen-Deklaration:

VAR
    var1      :      mein_feld;
    var2      :      INT;
END_VAR

Programmcode-Deklaration in ST:

var2 := var1[1][3];

```

Abbildung 152: Zugreifen auf Elemente eines Felds eines Felds

- Als Feldindex kann nur der Datentyp 'INT' verwendet werden. Ausdrücke wie 'i + 1' sind nicht möglich.
- In Feldern und Strukturen entspricht der Datentyp BOOL einem Byte.
- Während der Verwendung von lokalen Variablen für Felder und Strukturen muss beachtet werden, dass die entsprechenden Adressblöcke der I/O-Konfiguration in einer einzigen Definition stehen müssen.

- Um ein Feldelement innerhalb einer Struktur zu deklarieren, muss das entsprechende Feld vor der Deklaration der Struktur deklariert werden. Dies wird im folgenden Beispiel gezeigt.

```
TYPE
    Feld_Verwendet_In_Maschine: ARRAY[1..23] OF INT;
    maschine      :
    STRUCT
        x_pos      : INT;
        leistung   : Feld_Verwendet_In_Maschine;
    END_STRUCT;
END_TYPE
```

Abbildung 153: Deklarieren eines Feldelements innerhalb einer Struktur



HINWEIS

Alias-Datentypen und Datentypen für Unterbereiche sowie für Aufzählungen sind noch nicht verfügbar.

13.0.5 Implementationsbesonderheit des Datentyps STRING

- Wenn eine Variable vom Datentyp STRING in PROPROG wt II verwendet wird, wird sie immer mit einer Größe von 85 Zeichen gespeichert. 80 Byte entsprechen der maximalen Länge der Zeichenfolge und 5 Byte werden für internen Gebrauch reserviert.
- Wenn eine Konstante für Zeichenfolgen in einem Programmcode-Arbeitsblatt verwendet wird, benötigt sie ein Byte pro Zeichen plus 5 Bytes im OmegaOS Speicher.
- Funktionen und Funktionsbausteine für Zeichenfolgen können nicht in Netzwerken verwendet werden. Es können nur einzelne Funktionen und Funktionsbausteine verwendet werden, die das Ergebnis in Variablen speichern.
- Funktionen für Zeichenfolgen können beim Editieren in ST nicht verschachtelt werden.
- Zeichenfolgen können in Funktions-POEs nicht verwendet werden.
- Beim Überschreiben von Zeichenfolgen im Dialog 'Online-Debug' können nur 80 Zeichen verwendet werden.
- Bei Konstanten für Zeichenfolgen kann 'Arbeitsblatt überschreiben' nicht verwendet werden.



Bitte schlagen Sie in der OmegaOS-Dokumentation nach, um detaillierte Informationen über STRING-Datentypen und STRING-Funktionen zu erhalten.

13.0.6 Implementationsbesonderheit des ST

- In einer FOR-Anweisung kann nur aufwärts gezählt werden. Daher muss der Inkrementwert hinter dem Schlüsselwort BY immer positiv sein.
- In einer FOR-Anweisung können die Variable 'i' und der Inkrementwert innerhalb des Codes der Anweisung geändert werden. Es wird keine Fehlermeldung oder Warnung angezeigt, wenn Sie das folgende Beispiel eingeben:

Beispiel: FOR i=0 TO 10 BY5
 DO i:=10;

Verwenden Sie derartige Anweisungsstrukturen äußerst vorsichtig.

- In einem Funktionsaufruf können keine Zuweisungen verwendet werden. Das folgende Beispiel ist nicht möglich: a:= SIN(b:=3);
- In einem Ausdruck werden alle Operationen verarbeitet. Teile eines Ausdrucks, die keine Auswirkung auf das Ergebnis haben, werden ebenfalls verarbeitet. Im folgenden Beispiel wird c>d auch verarbeitet, wenn a<b FALSE ist.

Beispiel: a:=a<b AND c>d;

13.0.7 Implementationsbesonderheit des VAR_IN_OUT

- Ein VAR_IN_OUT-Ausgang kann nicht mit einem VAR_IN_OUT-Eingang eines anderen Funktionsblocks verbunden werden.

Bitte beachten Sie die folgenden Aspekte, die von der IEC 61131-3 empfohlen werden:

- VAR_IN_OUT kann nur für Funktionsblöcke verwendet werden
- Ein VAR_IN_OUT-Parameter wird einmal im Funktionsblock deklariert. Er ist in der FBS als Eingangs- und Ausgangsparameter der Funktionsblöcke sichtbar.
- Der Eingang und der Ausgang eines VAR_IN_OUT-Parameters muss mit derselben Variable verbunden werden.

13.0.8 Folgende Besonderheiten sind in PROPROG wt II zu beachten

- Nach der Übernahme eines Projekts aus einer älteren PROPROG wt Version muss das Projekt mit "Projekt neu erzeugen" übersetzt werden.
- Im Watch-Fenster können keine Variablen der Gruppen VAR_GLOBAL_PG und VAR_GLOBAL_FB angezeigt werden.
- Aus dem Seitenlayout-Editor heraus können keine Seitenlayouts gelöscht werden.
- Leerzeichen im Projektpfad und/oder -namen sind nicht erlaubt. Der Projektname ist auf 24 Zeichen begrenzt.
- Wenn im Projektbaum ein Knoten gelöscht wird, wird der Inhalt der Zwischenablage des Projektbaums ebenfalls gelöscht.

- 
-
- Es ist nicht möglich, Variablen mit mehr als 30 Zeichen aus dem Variablendialog auszuwählen.
 - Die Variablen "Varglobal_PG" und "Varglobal_FB" können nicht in die Logik-Analyse eingefügt werden.
 - Arrays und Strukturen können nicht aus der Bibliothek in das Watch-Fenster kopiert werden.
 - Auf String-Variablen ist ein Setzen von Breakpoints, Step und Trace nicht möglich.
 - Bei der Arbeit mit der Logik-Analyse darf die Konfiguration nicht gelöscht werden.
 - Funktionsbaustein-Parameter können nicht in die Watchliste eingetragen werden (z. B. TON1.IN).



Abbildungsverzeichnis

Beispiel für Konfigurationselemente.....	13
Darstellung der Default-Task mit zwei Programmen.....	15
Unterbaum 'Logische POEs'	15
Instanzbäum mit den Instanzen der Ressource 'R_IPC'.....	16
Arbeitsblätter eines Funktionsbausteins in FBS.....	16
Symbole einer AS-POE.....	17
Unterbaum 'Datentypen'.....	18
Das Projekt mit seinen Unterbäumen	19
Unterbaum 'Bibliotheken'	19
Benutzeroberfläche mit geöffnetem Beispielprojekt 'Example'	24
Beispiel für einen Tooltip (Symbol 'Speichern') und der zugehörigen Beschreibung in der Statuszeile 28	
Beispiel für ein Symboleistenfenster	29
Der Shortcut-Manager zum Hinzufügen/Ändern von Shortcuts	30
Dialog 'Shortcut zuweisen' bevor Sie die neuen Shortcut-Tasten gedrückt haben.....	31
Dialog 'Shortcut zuweisen' nachdem Sie die neuen Shortcut-Tasten gedrückt haben	32
Standard-Shortcuts in alphabetischer Reihenfolge.....	35
Die Reiter der Arbeitsblätter im Arbeitsbereich	36
Das Meldungsfenster mit seinen verschiedenen Reitern	37
Querverweifenster	39
Symbole im Querverweifenster	40
Dialog 'Querverweis-Filter' zum Filtern der in der Querverweisliste angezeigten Einträge.....	40
Arbeiten mit Wildcards im Dialog 'Querverweis-Filter'	41
Die Reiter des Projektbaums	45
Der Editor-Assistent für die verschiedenen Sprachen	48
Der Editor-Assistent für Variablen- und Datentyp-Arbeitsblätter.....	49
Navigieren im graphischen Arbeitsblatt mit Hilfe des Übersichtsfensters	52
Der Dialog 'PROPROG wt II'	53
Dialog 'Neues Projekt' mit den verfügbaren Projekt-Vorlagen und dem Projekt-Assistenten	56
Projekt 'Untitled' mit Programm 'Untitled' und den entsprechenden Arbeitsblättern	57
Dialog 'Als Vorlage speichern'.....	58
Dialog 'Eigenschaften' zum Ändern der Eigenschaften vorhandener POEs.....	60
Dialog 'Einfügen' zum Einfügen einer neuen POE.....	62
Projekt mit eingebundener Bibliothek 'Contain'.....	66
Dialog 'Import / Export' (nach Auswählen des Menüpunkts 'Export...')	74
Dialog 'Export Projektübersetzung'	74
Dialog 'Import / Export' (nach Auswählen des Menüpunkts 'Import...')	75
Dialog 'Import Projektübersetzung'	76
Dialog 'Projekteigenschaften'	77
Dialog 'Konvertierung des Quell-Codes'	78
Numerische Literale	81
Zeichenfolge- (String-) Literale.....	82
Zeitdauer-Literale	82
Elementare Datentypen	83
Generische Datentypen	84
Deklaration eines Datentyps für Felder.....	85
Programmierbeispiel für Felder.....	86
Deklaration eines Felds eines Felds	87
Zugreifen auf Elemente eines Felds eines Felds	87
Initialisieren von Feldelementen.....	88
Deklaration einer Datenstruktur	88
Deklaration eines Felds einer Struktur	89
Deklaration einer Struktur eines Felds	90



Zuweisen von Werten zu Komponenten einer Struktur	90
Deklaration eines Datentyps für Zeichenfolgen	91
Editor-Assistent in einem Datentyp-Arbeitsblatt	92
Datentyp-Arbeitsblatt 'Type' mit vorbereiteten Schlüsselwörtern, eingefügt mit dem Editor-Assistenten	92
Deklaration einer symbolischen, einer adressierten und einer direkt dargestellten Variablen ...	94
Beispiel für die Deklaration einer remanenten Variablen.....	95
Deklaration und Initialisierung einer symbolischen, einer adressierten und einer direkt dargestellten Variablen.....	96
Schlüsselwörter für Variablen-Deklarationen.....	98
Editor-Assistent in einem Variablen-Arbeitsblatt.....	99
Variablen-Arbeitsblatt mit vorbereitetem Schlüsselwort 'VAR', eingefügt mit dem Editor-Assistenten	99
Instanziierung eines Funktionsbausteins.....	100
Beispiel für einen Instanzbaum.....	101
Anweisung vom Typ 'Zuweisung' in ST	105
Operatoren in ST	105
Anweisungen in ST.....	107
Vorbereitete CASE-Anweisung, mit dem Editor-Assistenten eingefügt.....	108
Dialog 'Variable'	110
Dialog 'Automatische Variablen-Deklaration'	111
Funktionsaufruf in ST.....	112
Aufruf eines Funktionsbausteins in ST	112
Vorbereitete Funktion MAX und Funktionsbaustein CTU, eingefügt mit dem Editor-Assistenten	114
Beispiel einer Anweisungsliste	116
Operatoren, Modifier und Operanden in AWL	117
Modifier in AWL und ihre Bedeutung	117
Operator 'ADD' und Funktionsbaustein 'RS', beide eingefügt mit dem Editor-Assistenten	118
Dialog 'Variable' für eine zuvor markierte Variable	120
Dialog 'Automatische Variablen-Deklaration'	121
Dialog 'Variable'.....	122
Beispiel für einen Sprung.....	123
Beispiel für das Aufrufen einer Funktion.....	123
Beispiel für den Aufruf eines Funktionsbausteins.....	124
Vorbereitete Funktion MAX und Funktionsbaustein CTU, eingefügt mit dem Editor-Assistenten	126
Dialog 'Funktion/Funktionsbaustein'	130
Dialog 'Variable'	132
Dialog 'Automatische Variablen-Deklaration'	133
Funktion und Funktionsbaustein vor dem Verbinden	136
Die Verbindung ist hergestellt.....	136
Die Funktion wurde an eine freie Stelle verschoben.	
Die Verbindungslinie wird automatisch geführt.....	136
Der Ausgang 'CV' des Funktionsbausteins 'CTU' wird als gewünschter Verbindungspunkt markiert, bevor die Funktion 'ADD' eingefügt wird.....	137
Die Verbindung wird automatisch beim Einfügen der Funktion 'ADD' erstellt	137
Die Funktion wurde an eine freie Stelle verschoben.	
Die Verbindungslinie wird automatisch geführt.....	138
Kontakte und Spulen in KOP	142
Beispiel eines KOP-Netzwerks	143
KOP-Anfangsnetzwerk	144
KOP-Anfangsnetzwerk mit eingefügtem Kontakt 'C002'.....	145
KOP-Netzwerk mit einem parallelen Zweig	146



KOP-Netzwerk mit einem parallelen Zweig, eingefügt im Editiermodus für KOP-Zweige	147
Dialog 'Kontakt/Spule'	148
Dialog 'Automatische Variablen-Deklaration'	149
KOP-Netzwerk mit geänderten Eigenschaften.....	150
KOP-Netzwerk mit Funktionsbaustein 'CTU'.....	152
AS-Netzwerk mit einem Schritt und einer Transition.....	157
AS-Netzwerk mit vier Schritten	159
Dialog 'Schritt'	160
Dialog 'Verzweigung'	161
AS-Netzwerk mit zwei alternativen Zweigen	162
Dialog 'Verzweigung'	163
Teil eines AS-Netzwerks mit einem alternativen und einem simultanen Zweig	164
Dialog 'Aktion'.....	166
Dialog 'Automatische Variablen-Deklaration'	167
Aktionsblock mit Variablenname	168
Dialog 'Transition'.....	169
Transition als direkte Verbindung spezifiziert (mit blauem Verbindungspunkt).....	169
Dialog 'Variable'	170
Dialog 'Automatische Variablen-Deklaration'	171
Variable verbunden mit einer Transition	172
Dialog 'Einfügen'	173
Dialog 'Einfügen'	176
Dialog 'Ressource-Einstellungen für ...'	177
Einfügen einer DEFAULT-Task.....	178
Task-Einstellungen für DEFAULT-Task	179
Einfügen einer CYCLIC-Task.....	179
Task-Einstellungen für eine CYCLIC-Task	180
Task-Einstellungen für System-Task	181
Dialog 'Einfügen'	182
Fehlermeldung nach dem Kompilieren eines Projektes.....	184
Fehlermeldungen, wie sie im Meldungsfenster ausgegeben werden	185
Fälle, in denen der Menüpunkt 'Online-Änderungen' verwendet werden kann	186
Kontrolldialog	188
Dialog 'Senden'	189
AWL-Arbeitsblatt im Online-Modus	191
Graphisches Arbeitsblatt im Online-Modus	192
Bedeutung der Farben im Online-Modus	192
Zeichen der Durchlaufkontrolle in textuellen Arbeitsblättern im Online-Modus.....	196
Zeichen der Durchlaufkontrolle in graphischen Arbeitsblättern im Online-Modus	196
Dialog 'Debug: Ressource_Name'	198
Dialog 'Debug: Ressource_Name'	200
Kontrolldialog bei gesetztem Breakpoint.....	202
ST-Arbeitsblatt im Online-Modus mit einem gesetztem Breakpoint.....	203
Watch-Fenster mit eingefügten Variablen.....	205
Watch-Fenster mit Elementen des Felds einer Struktur	207
Logikanalyse mit zwei erfassten Kurven	209
Dialog 'Variablen' der Logikanalyse	210
Dialog 'Datenfenster' der Logikanalyse	211
Dialog 'Triggerkonfiguration' der Logikanalyse	212
Dialog 'Projekt drucken'.....	217
Seite 'Seitenlayouts' im Dialog 'Optionen'	219
Seitenlayout-Editor mit einem neuen Seitenlayout	221
Seitenlayout mit dem neuen Codebereich	222
Zugreifen auf Elemente eines Felds eines Felds	229



Deklarieren eines Feldelements innerhalb einer Struktur 230



Stichwortverzeichnis

A	
Abgeleitete Datentypen	17, 85
Ablaufsprache	
siehe AS	
Achse im Logikanalyse-Fenster	213
ADD	116
Addition	105, 116
Adressierte Variable	93
Adresstatus	195
Online-Modus	195
Aktion	156, 165
Lupe	173
Qualifier	156, 165
Aktionsblock	156, 157, 158
Als Vorlage speichern...	58
Alternativer Zweig in AS	156, 161, 164
AND	116
Anfangsschritt	156, 158
Umwandeln	160
Anfangswert	96
Anweisung	104
Auswahanweisungen	106
AWL	116
Einfügen in AWL mit dem Editor-Assistenten	117
Einfügen in ST mit dem Editor-Assistenten	107
Rücksprunganweisungen	106
Schlüsselwörter	106
Wiederholungsanweisungen	106
Zuweisung	105
Anweisungsliste	
siehe AWL	
Anwenderbibliothek	19, 65
Anwenderdefinierte Datentypen	17, 85
Zeichenfolge	91
ANY_BIT	84
ANY_INT	84
ANY_NUM	84
Arbeitsbereich	35
Arbeitsblatt	
Dokumentation	58
Drucken im Online-Modus	215
einfügen	63
Löschen	67
Online-Modus	190
schließen	52
Seitenansicht	224
Speichern	68
Variable	58
Archiv	70
Arithmetische Funktion	14
AS	156
Aktion	156
Aktionsblock	156
alternativer Zweig	161
Editiermodus für Zweige	164
Editor aufrufen	155
Kommentar	156
Lupe	156, 173
Mischen mit KOP	172
Netzwerk	156
Netzwerk ändern	160
Netzwerk einfügen	157, 158
Qualifier	156
Qualifier für Aktionen	165
Schritt	156
simultaner Zweig	162
Transition	156
Variable einfügen	165, 168
AS-Arbeitsblätter	46
Assistent	
Projekt-Assistent	55
Aufruf	
Funktionen und Funktionsbausteine in	
AWL	123
Funktionsbaustein in AWL	117
Aufrufen von Funktionen und Funktionsbausteinen	151
Ausdruck	104, 105
Auswahanweisung	106
Auswahlfunktion	14
Automatisches Backup	72
AWL	116
Anweisung	116
Funktionen aufrufen	123
Funktionsbausteine aufrufen	124
Kommentar	116
Marke	123
Modifier	117
Operand	116
Operator	116
Sprung	123
Variable einfügen	119
AWL-Arbeitsblätter	47
AWL-Editor aufrufen	115
B	
Beenden von PROPROG wt II	53
Benutzeroberfläche	
allgemeine Beschreibung	24
anpassen	24
Arbeitsbereich	35
Editor-Assistent	48
Hauptfenster	35
Meldungsfenster	37
Menüleiste	25



Stichwortverzeichnis

Querverweis-Fenster	38	Automatische FB-Deklaration	114, 125, 152
Statuszeile	43	Automatische Variablen-Deklaration	111, 121, 133, 149, 150, 167, 171
Symbolleisten	27	Codebereich einrichten	222
Übersichtsfenster	51	Debug 'Ressource_Name'	198, 200
Untermenüs	25	Eigenschaften	60
Besonderheiten in PROPROG wt II	231	Einfügen	62, 174, 176, 182
Bibliothek	19	Export Projektübersetzung	74
Anzeigen von Bibliothekenarbeitsblättern	65	FB-Instanzen	125, 129, 152
Drucken	215	Funktion/Funktionsbaustein	131
Einbinden	20, 65	Import / Export	74, 75
Bistabile Funktionsbausteine	14	Import Projektübersetzung	76
Bitfolge-Funktion	14	Kontakt/Spule	148
Bitmap im Seitenlayout	223	Kontrolldialog	188, 203
BOOL	83	Konvertierung des Quell-Codes	78
Boolesches Exklusiv-ODER	105, 116	Neues Projekt	56, 57
Boolesches ODER	105, 116	Optionen	219
Boolesches UND	105, 116	Projekt drucken	217
Breakpoint	199	Projekt speichern / packen unter...	70
Breakpoints		Projekteigenschaften	77
Debuggen mit gesetzten Breakpoints	201	Querverweis-Filter	40
BY	106	Ressource-Einstellungen für ...	177
BYTE	83	Schritt	160
C		Seitenlayout	219
CAL	117, 124	Senden	189
CASE	106	Shortcut zuweisen	31
Codebereich	221	Shortcut-Tasten	30
Code-Teil	16	Transition	169
csv (Dateierweiterung)	73, 75	Triggerkonfiguration	211
D		Umgebungstext einrichten	224
Datentyp		Variable	110, 119, 120, 122, 133, 170
abgeleitet	85	Variablen (der Logikanalyse)	211
Anfangswert	96	Verzweigung	161, 163
anwenderdefiniert	85	DINT	83
elementar	83	direkt dargestellte Variable	94
Feld	85	Direkte Verbindung	156
generisch	84	DIV	117
Struktur	88	Division	105, 117
Zeichenfolge	91	DO	106
Datentyp-Arbeitsblatt	47, 91	Dokumentation	
Datentyp-Deklaration		Handbuch	8
Editieren mit dem Editor-Assistenten	92	kontextsensitive Hilfe	8, 43
Datentypen	17, 229	Symbole	8
Implementationsbesonderheit anwender-		Dokumentations-Arbeitsblatt	73
definierter Datentypen	229	Drag & Drop in graphischen Arbeitsblättern	134
Datentypen für Zeichenfolgen	91	Drucken	215
Datentypen-Deklaration	18	Anzeigen der Seitenränder vor dem Drucken	216
Debuggen	194, 206	Bereich auswählen	217
Deklarationsteil	16	Druckmodus 'Optimiert'	217
Dialog		Druckmodus 'Standard'	217
Aktion	166	Elemente auswählen	217
Änderungen speichern?	53		



Duplizieren von Funktionseingängen	139	Kommentar	128
Durchlaufkontrolle	195	Mischen mit AS	172
DWORD	83	Negieren von Eingängen und Ausgängen	138
E		Netzwerk	128, 134
Edit		Objekte verbinden	134
AS-Arbeitsblätter editieren	155	Variable einfügen	132
Editiermodus für KOP-Zweige	146	FBS-Arbeitsblätter	46
Editor		Fehler	
Arbeitsblätter im Online-Modus	190	Anzeige im Meldungsfenster	37
AWL-Arbeitsblätter editieren	115	Programmierfehler korrigieren	194
FBS-Arbeitsblätter editieren	127	Fehler beim Kompilieren	184
KOP-Arbeitsblätter editieren	141	Fehlerliste (Meldungsfenster)	185
ST-Arbeitsblätter editieren	103	Felder (Datentyp)	85
Editor-Assistent	119, 128	Debuggen	206
allgemeine Beschreibung	48	Felder von Strukturen	89
Anweisungen einfügen in ST	107	mehrdimensionales Feld	87
Aufrufen von Funktionen/Funktionsbausteinen in AWL	123	Strukturen mit Feldern	90
Aufrufen von Funktionen/Funktionsbausteinen in ST	112	Filtern der Querverweisliste	40
Auswahlbereich	49	Firmware-Bibliothek	19, 65
AWL-Anweisungen einfügen	117	Flankenerkennung, Funktionsbausteine für	14
Editieren von Datentyp-Deklarationen	92	FOR	106
Einfügen von Schlüsselwörtern für Variablen-Deklarationen	98	Forcen von Variablen	196
Funktionen/Funktionsbausteine ersetzen	131	Formalparameter	132, 138
Funktionen/Funktionsbausteine in KOP einfügen	151	Funktion	43
in ST	104	allgemeine Beschreibung	14
Listenfeld 'Gruppe'	49	Aufruf in AWL	123
Eigenschaften	59	Aufruf in KOP	151
Ein- und Ausgänge deklarieren	95	Aufruf in ST	112
Eingangsparameter	123	Duplizieren von Eingängen	139
elementare Datentypen	17, 83	Eigenschaften ändern	130
ELSE	106	Ersetzen mit dem Editor-Assistenten	131
ELSIF	106	Firmware	65
END_CASE	106	Negieren von Eingängen und Ausgängen	138
END_FOR	106	Typangabe	14
END_IF	106	zur Typumwandlung	14
END_REPEAT	107	Funktionsbaustein	43
END_VAR	98	allgemeine Beschreibung	14
END_WHILE	106	Aufruf in AWL	117, 124
EQ	117	Aufruf in KOP	151
Ereignis-Task	12	Aufruf in ST	112
Erstellen eines Projektes	55	Eigenschaften ändern	130
EXIT	107	Ersetzen mit dem Editor-Assistenten	131
F		Firmware	65
Farben der Statuszeile	43	Instanz	15
Farben im Online-Modus	192	Instanzen deklarieren	96, 100
FBS	128	Negieren von Eingängen und Ausgängen	138
Editor aufrufen	127	Typen	14
		Funktionsbaustein-Sprache	
		siehe FBS	



Stichwortverzeichnis

G

GE	117
Generische Datentypen	17, 84
Gleichheit	105
globale Variable	95
Graphischer Editor	
allgemeine Beschreibung	46
Editieren in AS	155
Editieren in FBS	127
Grenzen für Projekte	227
GT	117
Gültigkeitsbereich	
Variable	94

H

Hardwarevoraussetzungen	21
Hauptfenster	35
Hilfe zu FU/FB	43

I

Icons im Querverweisfenster	39
IEC 61131	11
Konfiguration	12
POE	13
Programm	14
Ressource	12
Task	12
IEC 61131-3	
Datentypen	17
Funktion	14
Funktionsbaustein	14
Instanziierung	15
Konfigurationselemente	12
Programmiersprache	20
Variablen	17
IEC-Code	182
IF	106
Initialisieren von Feldern	88
Initialisieren von Variablen	95
Installation	22
Instanz	15, 100
Instanz öffnen	190
Instanzbaum	16, 45, 101
Instanziierung	15, 100
Funktionsbausteine	125, 129, 152
Instanzname	16, 100, 182
INT	83
Internen Speicher	100
interner Speicher	14, 15
Interrupt-Task	12

J

JMP	117, 123
-----	----------

K

Klammerung	105
Kommentar	73
AS	156
AWL	116
FBS	128
KOP	143
ST	104
Kompilieren	182
Fehler	184
Make	183
Online-Änderungen von POEs	185, 194
Komplement	105
Konfiguration	12, 13
Einfügen	175
Konfigurationselemente	12, 13
Kontakt	142
Eigenschaften ändern	147
Einfügen	143
Kontaktplan	
siehe KOP	
kontextsensitive Hilfe	43
Konvertierung des Quell-Codes	78
KOP	142
Editor aufrufen	141
Eigenschaften von Kontakten / Spulen ändern	147
Kommentar	143
Kontakt	142
Kontakt einfügen	143
Mischen mit AS	172
Netzwerk	142
paralleler Zweig	142, 145, 146
Spule	142
Spule einfügen	143
Stromschiene	142
Variable	143
Variable einfügen	150
Verbindungsline	142
verdrahtetes ODER	142, 145
KOP-Arbeitsblätter	46
Korrigieren von Programmierfehlern	194

L

Layout	
Einstellung	193
LD (Operator)	116
LE	117
Linie	
Seitenlayout	223
Literal	
allgemeine Beschreibung	81
numerisch	81
Zeichenfolge	82



Zeitdauer	82	Farben	192
Logikanalyse		Felder und Strukturen debuggen	206
allgemeine Beschreibung	208	Funktions-Arbeitsblatt	195
Dialog 'Triggerkonfiguration'	211	Layout	192, 193
Dialog 'Variablen'	211	Variablenstatus	195
Einstellen der Anzeige im Logikanalyse-Fenster	213	Watch-Fenster	204
Untermenü im Untermenü 'Online'	26	Operand	
Voraussetzungen für die Verwendung	210	AWL	116
lokale Variable	94	ST	104, 105
LT	117	Operator	
Lupe	156, 173	AWL	116
		ST	104, 105
M		Optimiertes Drucken	217, 219
Make	183, 184	OR	116
Marke	123	P	
Maus, Verwendung	23	Packen eines Projekts	70
Meldungsfenster	37	Paralleler Zweig in KOP	142
Fehler beim Kompilieren	185	paralleler Zweig in KOP	145, 146
Warnungen beim Kompilieren	185	POE	
Menüleiste	25	Eigenschaften ändern	59
Mischen		Einfügen	61
FBS und KOP	151	Löschen	67
Modifizier		Typen	13
AWL	116, 117	Potenzierung	105
Modulo	105	Präfix für Speichergröße	94
MUL	117	Programm	
Multiplikation	105, 117	allgemeine Beschreibung	14
		Instanz	16, 101, 182
N		interner Speicher	14
NE	117	Task zuweisen	181
Negation	105	Programmierfehler	194
Negative Spule	142	Programmiersprache	
Negieren von Eingängen und Ausgängen	138	FBS	128
Netzwerk		Graphik	20
AS	156	KOP	142
AS-Netzwerk einfügen	157	ST	104
FBS	128, 134	Text	20
KOP	142	Programmiersprachen	228
Numerische Funktion	14	Programm-Module	13
numerische Literale	81	Programm-Organisationseinheit	
		siehe POE	
O		Programmstart	22
Öffner	142	Projekt	
Online-Änderungen	183, 185, 194	'Untitled'	57
Online-Layout		automatisches Packen	72
Einstellung	193	erstellen	55
Online-Modus	190	Erzeugen mit Projekt-Assistent	55
Durchlaufkontrolle im Online-Modus	196	Exportieren einer Projektübersetzungsdatei	73
Durchlaufkontrolle in graphischen Arbeitsblättern	196	Importieren einer Projektübersetzungsdatei	75
Ein-/Ausschalten	194	Kompilieren	182
		Packen in einer Archivdatei	70



Stichwortverzeichnis

Senden	187	Schließen von Arbeitsblättern	52
Speichern als Vorlage	58	Schließler	142, 143
Speichern unter ...	69	Schlüsselwörter in ST	104
Übersetzen der Projektsprache	73	Schritt	156
Umschalten der Projektsprache	77	Anfangsschritt	156
Vorlage	57	Eigenschaften ändern	160
Projekt neu erzeugen	183	Einfügen	157, 158
Projekt-Assistent	55	Seitenansicht	224
Projektbaum	18	Seitenlayout	
Arbeitsblatt einfügen	63	Definieren als Standardseitenlayout	218
Konfiguration einfügen	175	Erstellen eines neuen Layouts	220
Konfigurationselemente	13	Seitenansicht	224
POE einfügen	61	Seitenlayout-Editor	
Ressource einfügen	175	allgemeine Beschreibung	47
Task einfügen	175	Aufrufen	220
Unterbaum 'Bibliotheken'	20, 45	Definieren eines Codebereichs	221
Unterbaum 'Datentypen'	18, 45	Grafik (Bitmap) einfügen	223
Unterbaum 'Hardwarestruktur'	45	Linie einfügen	223
Unterbaum 'Logische POEs'	15, 45	Rechteck einfügen	223
Projektbaum-Editor	45	System Item einfügen	223
Projektübersetzungsdatei		Text einfügen	223
exportieren	73	Umgebungselement einfügen	223
Projektvorlage	57	Seitenränder	216
PROPROG wt II		Senden	187
Beenden	53	Setup	22
Benutzeroberfläche	24	SETZE-Spule	142
Installation	22	Shortcut-Manager	30
Programmstart	22	Shortcuts	29
Q		Hinzufügen/Ändern	30
Qualifier in AS	156, 165	löschen	31
Querverweifenster	38	Standard-Shortcuts	32
Symbole in der Spalte 'Variable'	39	Zuweisen eines neuen Shortcut	31
R		Simultaner Zweig in AS	156, 162, 164
R	116	SINT	83
Ränder	216	Softwarevoraussetzungen	22
REAL	83	Sortieren der Querverweisliste	42
Rechteck im Seitenlayout	223	Speichern	68
remanente Variable	95	Speichern unter	69
Anfangswert	96	Speicherort	94
REPEAT	107	Präfix	94
Ressource	12, 13	Speicherplatz auf der Festplatte, Anzeige in der Statuszeile	43
Einfügen	175	Sprache	
Einstellungen	177	Exportieren einer Projektübersetzungsdatei	73
Kontrolldialog	188, 203	Importieren einer Projektübersetzungsdatei	75
RET	117	Übersetzen der Projektsprache	73
RETAIN	95	Umschalten der Projektsprache	77
RETURN	107	Sprachelement	13
RÜCKSETZE-Spule	142	Sprung	117, 123
Rücksprung	117	SPS-Code	182
S		Spule	142
S	116	Eigenschaften ändern	147



Einfügen	143	Einfügen/Löschen von Variablen	210
ST	116	FB hinzufügen	61
allgemeine Beschreibung	104	FBS-Arbeitsblatt	127
Anweisung	104	FP duplizieren	139
Anweisungen editieren	106	FP-Negation ein/aus	138
Aufrufen des Editors	103	Funktion hinzufügen	61
Ausdruck	104, 105	Kontakt rechts einfügen	144
CASE-Anweisung	106	Kontakt/Spule über Markierung einfügen	145
EXIT-Anweisung	107	Kontakt/Spule unterhalb Markierung einfügen	145
FOR-Anweisung	106	Kontaktnetzwerk	144
Funktion aufrufen	112	Kontrolldialog anzeigen	188, 202
Funktionsbaustein aufrufen	112	KOP-Arbeitsblatt	141
IF-Anweisung	106	KOP-Zweig einfügen	146
Implementationsbesonderheit	231	Linie	223
Kommentar	104	Logikanalyse	208
Operand	104, 105	Make	184
Operator	104, 105	Objekt hinzufügen	63, 64, 66
REPEAT-Anweisung	107	Objekte verbinden	135
Rücksprung-Anweisung	107	Öffner	147
Schlüsselwörter der Anweisungen	106	POE-Name	67
Variable einfügen	109	Programm hinzufügen	61
WHILE-Anweisung	106	Programmname	59
Zuweisung	112	Ressource-Name	175
Zuweisung editieren	105	Simultan-/Alternativverzweigung einfügen	161, 162
Standard-Anfangswert	96	Speichern	68
Standarddruckmodus	217	ST-Arbeitsblatt	103
Standard-Funktionsbaustein	14	Task-Name	181
Standardseitenlayout	219	Transition-Schritt-Kette einfügen	157, 158
ST-Arbeitsblätter	47	Triggerkonfiguration	211
Statuszeile	43	Umgebungselemente	223
Step (Debuggen)	203	Variable	109, 120, 132, 170
STRING	84	Variablen-Arbeitsblatt	64
Implementationsbesonderheit	230	Watch-Fenster	204
Stromschiene	142	Symbole	
Struktur	88	Beschreibung in der Statusleiste	27
Debuggen	206	im Querverweisfenster	39
Initialisieren	90	Tooltips aktivieren	27
Strukturen mit Feldern	90	symbolische Variable	93
SUB	116	Symbolleisten	27
Subtraktion	105, 116	System Item	223
Symbol		Systemtask	12
AS-Arbeitsblatt	155	T	
AS-Zweig einfügen	165	Task	12, 13
Aufzeichnung starten	213	Einfügen	175
Aufzeichnung stoppen	213	Programm zuweisen	181
AWL-Arbeitsblatt	115	Tastatur, Verwendung	23
Bibliotheken (Ordner)	66	Tastatur-Shortcuts	29
Code-Arbeitsblatt	63	Texteditor	
Codebereich	221	allgemeine Beschreibung	47
Datentyp-Arbeitsblatt	91		
Datentypen (Ordner)	64		
Debug ein/aus	187, 190, 194		
Editor-Assistent	108, 113, 118, 124, 129, 151		



Stichwortverzeichnis

Deklarieren von Datentypen	91	Einfügen in AS	165, 168
Editieren in AWL	115	Einfügen in AWL	119
Editieren in ST	103	Einfügen in das Watch-Fenster	204
TIME	83, 84	Einfügen in die Logikanalyse	210
TO	106	Einfügen in FBS	132
Tooltips	27	Einfügen in KOP	150
Trace	203	Einfügen in ST	109
Transition	156, 158	Forcen	196
direkte Verbindung	156	globale	95
Eigenschaften ändern	168	Gültigkeitsbereich	94
Einfügen	157, 158	Initialisieren	95
Lupe	173	lokale	94
Variable	168	Löschen in Logikanalyse	211
Triggerkonfiguration (Logikanalyse)	211	remanente	95
		Status im Online-Modus	195
U		symbolische	93
Überschreiben von Variablen	196	Überschreiben	196
Übersetzen der Projektsprache	73	Watch-Fenster	204
Übersetzungsdatei		Variablen-Arbeitsblätter	47
exportieren	73	Variablen-Deklaration	16, 17
importieren	75	Einfügen von Schlüsselwörtern mit dem	
Umschalten der Projektsprache	77	Editor-Assistenten	98
Übersichtsfenster	51	Instanzieren eines Funktionsbausteins	
UDINT	83	96,	100
UINT	83	Präfix für Speichergröße	94
Umgebungselement	223	Präfix für Speicherort	94
Ungleichheit	105	Schlüsselwörter	96
Unterbaum 'Bibliotheken'	45	Speicherort	94
Unterbaum 'Hardwarestruktur'	13, 45	Typen	96
Unterbaum 'Bibliotheken'	19	Verbinden von graphischen Objekten beim	
Unterbaum 'Datentypen'	18, 45	Einfügen	137
Unterbaum 'Instanzen'	16	Verbinden von Objekten in FBS	134
Unterbaum 'Logische POEs'	15, 45	Verbindungslineie	142
Untermenüs		verdrahtetes ODER	142, 145
allgemeine Beschreibung	25	Vergleich	105, 117
UNTIL	107	Vergleichsfunktion	14
USINT	83	Vorlage	
		Speichern eines Projekts als Projektvorlage	58
V			
VAR	96	W	
VAR_EXTERNAL	95, 97	Warnungen	
VAR_EXTERNAL_FB	97	Anzeige im Meldungsfenster	37
VAR_EXTERNAL_PG	97	Watch-Fenster	204
VAR_GLOBAL	95, 97	Felder und Strukturen debuggen	206
VAR_GLOBAL_FB	98	Variable einfügen	204
VAR_GLOBAL_PG	97	WHILE	106
VAR_IN_OUT	97	Wildcards	41
Implementationsbesonderheit	231	Filtern von Querverweisen	41
VAR_INPUT	94, 96	WL	
VAR_OUTPUT	94, 97	Modifier	116
Variable	17	WORD	83
adressierte	93		
Deklarieren	98		
direkt dargestellte	94		

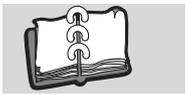


X

XOR 116

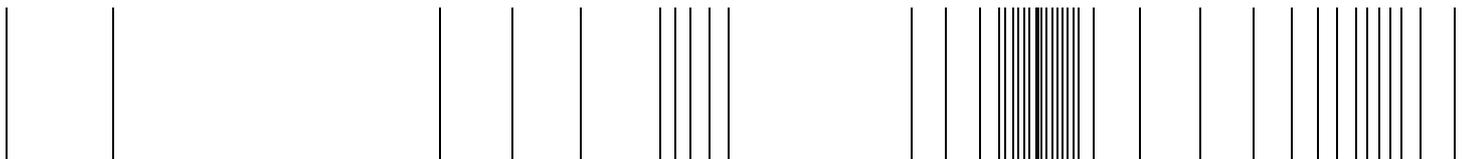
Z

Zähler, Funktionsbausteine für 14
Zeichenfolge-Funktion 14
Zeichenfolge-Literal 82
Zeitdauer-Literal 82
Zeitgeber, Funktionsbausteine für 14
Zeitverhalten 12
Zuweisen von Programmen zu Tasks 181
Zuweisung 105
Zweig
 alternativer Zweig in AS 156
 KOP 142
 simultaner Zweig in AS 156
zwt-Dateien 70
Zyklische Task 12



Stichwortverzeichnis

be in motion



Baumüller Nürnberg GmbH Ostendstraße 80-90 90482 Nürnberg T: +49(0)911-5432-0 F: +49(0)911-5432-130 www.baumueller.de

Alle Angaben in dieser Betriebsanleitung sind unverbindliche Kundeninformationen, unterliegen einer ständigen Weiterentwicklung und werden fortlaufend durch unseren permanenten Änderungsdienst aktualisiert. Bitte beachten Sie, dass Angaben/Zahlen/Informationen aktuelle Werte zum Druckdatum sind.
Zur Ausmessung, Berechnung und Kalkulationen sind diese Angaben nicht rechtlich verbindlich. Bevor Sie in dieser Betriebsanleitung aufgeführte Informationen zur Grundlage eigener Berechnungen und/oder Verwendungen machen, informieren Sie sich bitte, ob Sie den aktuellsten Stand der Informationen besitzen.
Eine Haftung für die Richtigkeit der Informationen wird daher nicht übernommen.