

Panasonic®

PROGRAMMIER-SOFTWARE

Control FPWIN Pro V6.0 Referenzhandbuch





Panasonic Electric Works Europe AG, im Folgenden kurz PEW genannt, weist darauf hin, dass Informationen und Hinweise in diesem Handbuch technischen Änderungen unterliegen können, da die Produkte von PEW ständig weiterentwickelt werden. Dieses Handbuch ist keine Zusicherung von PEW im Hinblick auf die dort beschriebenen technischen Vorgänge oder bestimmte dort wiedergegebene Produkteigenschaften. PEW übernimmt keine Haftung für die in diesem Handbuch enthaltenen Druckfehler oder sonstige Ungenauigkeiten, es sei denn, dass PEW die Fehler oder Ungenauigkeiten nachweislich bekannt sind oder diese PEW aufgrund grober Fahrlässigkeit unbekannt sind und PEW von einer Behebung der Fehler oder Ungenauigkeiten aus diesen Gründen abgesehen hat. PEW weist den Anwender ausdrücklich darauf hin, dass dieses Handbuch nur eine allgemeine Beschreibung technischer Vorgänge und Hinweise enthält, deren Umsetzung nicht in jedem Einzelfall in der vorliegenden Form sinnvoll sein kann. In Zweifelsfällen ist daher unbedingt mit PEW Rücksprache zu nehmen.

Dieses Handbuch ist urheberrechtlich geschützt. PEW behält sich alle Rechte vor. Ohne die vorherige schriftliche Zustimmung von PEW ist die Anfertigung von Kopien oder Teilkopien sowie die Übersetzung dieses Handbuchs in eine andere Sprache nicht zulässig.

Verbesserungsvorschläge zu diesem Handbuch werden gerne entgegengenommen unter: tech-doc@eu.pewg.panasonic.com

- © Nachdruck und Vervielfältigung, auch auszugsweise, nur mit der ausdrücklichen Genehmigung von:

Panasonic Electric Works Europe AG
Rudolf-Diesel-Ring 2
D-83607 Holzkirchen

- © MS-DOS und Windows sind eingetragene Warenzeichen der Microsoft Corporation.
- © IBM Personal Computer AT ist eingetragenes Warenzeichen der International Business Machines Corporation.

Wichtige Symbole

Die folgenden Symbole werden in diesem Handbuch verwendet:



Warnung!

Unter dem Warndreieck werden im Handbuch besondere Sicherheitshinweise und Warnungen gegeben. Bei deren Nichteinhaltung können je nach speziellem Fall Personenschäden und/oder erhebliche Sachschäden auftreten.



⬥ VORSICHT

Verfahren Sie mit Vorsicht! Bei Nichtbeachtung besteht Verletzungsgefahr oder die Gefahr von Geräteschäden bzw. Datenverlust.



⬥ Hinweis

Enthält wichtige zusätzliche Informationen.



⬥ BEISPIEL

Enthält ein Beispiel zur Veranschaulichung des vorhergehenden Textabschnitts.



⬥ Vorgehensweise

Kennzeichnet eine Schritt-für-Schrittanleitung.



⬥ REFERENZ

Weist auf eine zusätzliche Informationsquelle hin.



◆ KERNAUSSAGEN

Zusammenfassung von Leitgedanken.



◆ TASTATURBEFEHLE

Hinweise auf nützliche Tastaturbefehle (Shortcuts).



◆ ERLÄUTERUNGEN

Gibt eine kurze Erläuterung zu einer Funktion, z.B. warum und wann deren Einsatz sinnvoll ist.

 nächste Seite

Zeigt an, dass der Text auf der nächsten Seite fortgesetzt wird.

Außerdem werden in diesem Handbuch folgende Schreibweisen verwendet, um Elemente der Software-Oberfläche und der Tastatur zu kennzeichnen.

"Textfeld"	Eingabefelder oder Optionsnamen werden in Anführungszeichen angegeben.
[Schaltfläche]	Schaltflächen werden in eckigen Klammern angegeben, z.B. [Abbrechen].
<Taste>	Tastatureingaben werden mit spitzen Klammern gekennzeichnet.

Inhaltsverzeichnis

1. Installation und erste Schritte	1
1.1 Programm installieren und starten	2
1.2 Erste Schritte	3
2. Benutzeroberfläche	5
2.1 Start-Dialog	6
2.2 Hauptfenster	7
2.3 Projektnavigator	8
2.4 Programmierfenster	10
2.5 Statusleiste	12
3. Projekte in Control FPWIN Pro	15
3.1 Woraus besteht ein Projekt in Control FPWIN Pro?	16
3.1.1 SPS	17
3.1.1.1 Systemregister einstellen	17
3.1.1.2 E/A Adressen	18
3.1.1.3 MEWNET-F-Adressliste	20
3.1.1.4 Programm-Code	22
3.1.2 Bibliotheken	22
3.1.3 Tasks	22
3.1.4 Strukturierter Datentyp (SDT)	22
3.1.5 Globale Variablen	22
3.1.6 POEs	23

3.2	Projekte erstellen.....	24
3.2.1	Neues Projekt erstellen	24
3.2.2	Projekt von der SPS öffnen	24
3.2.3	Gesichertes Projekt wiederherstellen.....	25
3.2.4	Ein bestehendes Projekt öffnen	25
3.3	Projekte bearbeiten	26
3.3.1	Datensicherung (Backup) erstellen	26
3.3.2	Projekt wiederherstellen (Restore)	26
3.4	Objekte bearbeiten	27
3.5	Passwörter und Sicherheitsstufen	28
3.5.1	Zugriff auf geschützte Objekte	29
3.6	Aufrufbaum	30
3.7	Benutzt-von-Liste.....	31
3.8	Kontrolle	32
4.	Programm-Organisationseinheiten	33
4.1	Programm-Organisationseinheiten.....	34
4.1.1	Programme.....	35
4.1.2	Funktion (FUN)	36
4.1.3	Funktionsbaustein (FB)	37
4.2	POE neu anlegen	39
5.	Variablen	41
5.1	Variablen	42
5.1.1	Globale Variablen	42
5.1.1.1	Adressen	43
5.1.1.2	Globale Variablen deklarieren	43
5.1.2	Lokale Variablen.....	44
5.1.2.1	Globale Variablen in den POE-Kopf übernehmen	45

5.1.2.2	Überflüssige globale Variablen im POE-Kopf löschen.....	46
5.1.2.3	Lokale Variablen deklarieren.....	47
5.1.2.4	Variablen im Programm zuweisen	50
5.1.3	Systemvariablen.....	52
5.2	Variablen ändern.....	53
5.3	Variablen exportieren.....	54
5.4	Variablen importieren.....	56
5.5	Array und strukturierter Datentyp.....	58
5.6	Querverweislisten	59
5.7	Rezeptureditor	60
6.	Tasks	61
6.1	Tasks	62
6.2	Programm einer Task nachträglich zuweisen	63
6.3	Zweite Task für FP10SH/120K und FP2SH 120k	66
7.	Netzwerke	69
7.1	Netzwerke Einführung.....	70
7.2	Netzwerkliste.....	71
7.3	Marken und Titel definieren	72
7.4	Netzwerk deaktivieren/aktivieren	73
8.	Programmiereditoren	75
8.1	KOP- und FBS-Editor.....	76
8.1.1	Objekte verbinden	77
8.1.1.1	Nützliche Tipps.....	78

8.1.1.2	Schaltflächen in der Symbolleiste	80
8.1.1.3	Kontextmenü	80
8.1.1.4	Programmierbeispiel	81
8.1.1.5	Enable Input und Enable Output	81
8.1.1.6	Programmiersymbole bearbeiten	82
8.1.1.7	KOP oder FBS-Programm überprüfen	83
8.2	Structured Text Editor (ST).....	84
8.2.1	Ausdrücke.....	84
8.2.2	Operanden.....	85
8.2.3	Operatoren	85
8.2.4	Anweisungen	86
8.2.5	Kommentare	89
8.2.6	Eingabehilfen.....	90
8.2.7	Eingabeschablonen für ST-Editor	90
8.2.7.1	OP/FUN/FB-Auswahl	92
8.2.7.2	Operanden eingeben.....	93
8.2.8	Besonderheiten des ST-Editors	93
8.2.9	Programmierbeispiel.....	94
8.3	AWL-Editor	95
8.3.1	Eingabehilfen.....	96
8.4	Ablaufsprache (AS-Editor).....	97
9.	Projekt in die SPS übertragen.....	103
9.1	Projekt zur SPS übertragen: Einführung	104
9.2	SPS-Typ	105
9.3	Adressbereiche Adressbereiche.....	106
9.4	Programm kompilieren	109
9.4.1	Inkrementales Kompilieren.....	109
9.5	Speicherbelegung prüfen	111
9.6	Schnittstellen-Parameter	113

9.6.1	Netzwerk-Parameter.....	113
9.7	Online-Modus.....	114
9.8	Sicherheitseinstellungen	115
9.9	Projekt auf die SPS übertragen	117
9.9.1	Änderungen im RUN-Modus übertragen.....	118
9.9.2	Programm im RUN-Modus online ändern	119
9.9.3	Programm-Code und SPS-Konfiguration rückübertragen.....	120
9.9.4	Programm löschen und Systemregister zurücksetzen.....	121
9.9.5	Programm-Code und Systemregister vergleichen	122
9.9.6	Menüs / Online / SPS-Konfiguration.....	122
9.9.7	SPS-Modus ändern	122
9.10	Arbeitsweise des Compilers.....	123
9.10.1	Reihenfolge beim Abarbeiten von Netzwerken in einer POE	123
9.10.2	Abarbeitungsreihenfolge in KOP- und FBS-Netzwerken	123
9.10.3	Aufteilen von Netzwerken.....	124
9.10.4	Behandlung von Variablen im Compiler	125
9.10.5	Behandlung von Sprüngen im Compiler	126
9.10.6	Behandlung von FUN/FB im Compiler	126
9.10.7	Festimplementierte FUN/FB.....	126
9.10.8	Benutzerdefinierte Funktionen	127
9.10.9	Benutzerdefinierte Funktionsbausteine	128
9.10.9.1	Funktionsbausteininstanzen im Selbsthaltebereich	129
9.10.10	Interrupt-Programme im Compiler	131
9.10.10.1	Sichern der Indexregister in Interrupt-Programmen.....	131
10.	Fehlersuche	133
10.1	Fehlersuche Einführung.....	134
10.2	Einzelschrittmodus.....	135
10.3	Haltepunktmodus	136
10.4	Haltepunkte.....	138

10.4.1 Haltepunkte im AWL/ST-Editor	138
10.4.2 Haltepunkte im KOP-Editor	139
10.4.3 Haltepunkte im FBS-Editor	139
10.4.4 Haltepunkte im AS-Editor	140
 11. Überwachung	141
11.1 Monitorbetrieb Einführung	142
 12. Zusatzspeicher	143
12.1 IC-Karte, EEPROM, EPROM	144
 13. Export/Import.....	145
13.1 Export/Import Einführung.....	146
13.2 Einführung in Reusability Level ST.....	147
13.3 Projekte exportieren/importieren	148
13.4 Objekte exportieren/importieren	150
13.5 Programm-Code exportieren/importieren	152
 14. Tastaturbelegung	153
14.1 Tastaturbefehle für allgemeine Funktionen	154
 15. Glossar	157
 Index	163

Kapitel 1

Installation und erste Schritte

1.1 Programm installieren und starten

Bevor Sie Control FPWIN Pro installieren, prüfen Sie, ob Ihr Rechner folgenden Anforderungen entspricht:

- Pentium-Prozessor oder kompatibler
- 128 MB RAM oder mehr
- CD-ROM Laufwerk
- Festplatte mit mindestens 100 MB freiem Speicherplatz
- VGA-Monitor oder kompatibler
- Maus
- COM-Schnittstelle (seriell) für den Anschluss an Ihre SPS

Bevor Sie Control FPWIN Pro installieren, muss Windows 95, 98, NT V4.0, 2000, ME, XP oder Vista bereits auf Ihrem PC installiert sein.

Bitte lesen Sie die Informationen zum Upgrade genau durch, um Konflikte mit älteren Versionen zu vermeiden.

Upgrade-Informationen:



◆ Hinweis

- **Soll eine Installation unter Windows 2000, XP oder Vista durchgeführt werden, benötigen Sie die Administratorrechte auf Ihrem Computer.**
- **Eine gewisse Erfahrung im Umgang mit Microsoft Windows wird vorausgesetzt.**



◆ Vorgehensweise

1. **Windows starten**
2. **CD-ROM von Control FPWIN Pro in CD-ROM-Laufwerk einlegen**
Es wird automatisch ein Browser gestartet, der Ihnen eine Übersicht über den Inhalt der CD-ROM bietet.
3. **Sprache auswählen**
4. **"Software installieren" wählen**
Befolgen Sie die Anweisungen am Bildschirm, die Sie durch die Installation führen.
Zum Starten des Programms:
5. **Programme → Panasonic MEW Control → FPWIN Pro 6 → FPWIN Pro 6**

1.2 Erste Schritte

Auf der Produkt-CD-ROM befindet sich das Lernprogramm "Erste Schritte", mit dem Sie eine schnelle und einfache Einführung in das Programmieren mit Control FPWIN Pro erhalten. Das Lernprogramm wird zusammen mit Control FPWIN Pro auf Ihrem Rechner installiert. Starten Sie das Programm über das Startmenü mit **Programme → Panasonic MEW Control → FPWIN Pro 6 → FPWIN Pro 6**.

Die folgenden Dateien sind ebenfalls auf der beiliegenden CD enthalten und unter "Helpful Files" zu finden:

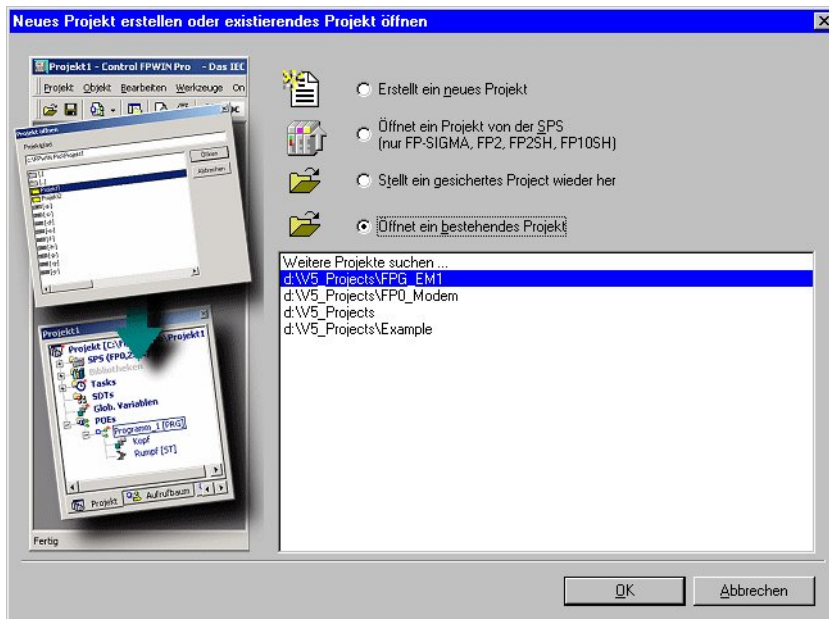
- IEC61131_3_basics.pdf für eine Einführung in den IEC61131–3 Standard.
- Falls Sie sich das Lernprogramm ausdrucken möchten, verwenden Sie die PDF-Datei 1stSteps_German.pdf, die sich auf der CD-ROM im Verzeichnis \1st Steps Demo befindet. Aktivieren Sie im Acrobat Reader unter "Erweiterte Optionen" die Option "Als Bild drucken", bevor Sie die Datei ausdrucken.
- Ein Link auf unsere Webseite zum Herunterladen der Handbücher.

Kapitel 2

Benutzeroberfläche

2.1 Start-Dialog

Wenn Sie FPWIN Pro starten, erscheint ein Assistent zum Öffnen oder Erstellen eines Projekts:



Im Start-Dialog können Sie folgende Aktionen ausführen:

Der Assistent erstellt ein neues Projekt, in das Sie Programme, Funktionen oder Funktionsbausteine einfügen können. Sie können auch ein leeres Projekt erstellen.

FPWIN Pro öffnet ein Projekt, das im Kommentarspeicher der SPS gespeichert wurde.

Ein Projekt wird geöffnet, das zuvor mit **Extras** → **Projekt sichern (Backup)** gesichert wurde.

Mit dieser Option können Sie ein Projekt auf Ihrem Computer oder auf dem Netzlaufwerk öffnen. Zum Öffnen eines der zuletzt verwendeten Projekte doppelklicken Sie in der Liste der Projekte auf den gewünschten Namen.

Zum Öffnen eines hier nicht aufgeführten Projekts doppelklicken Sie auf **Weitere Projekte suchen**.

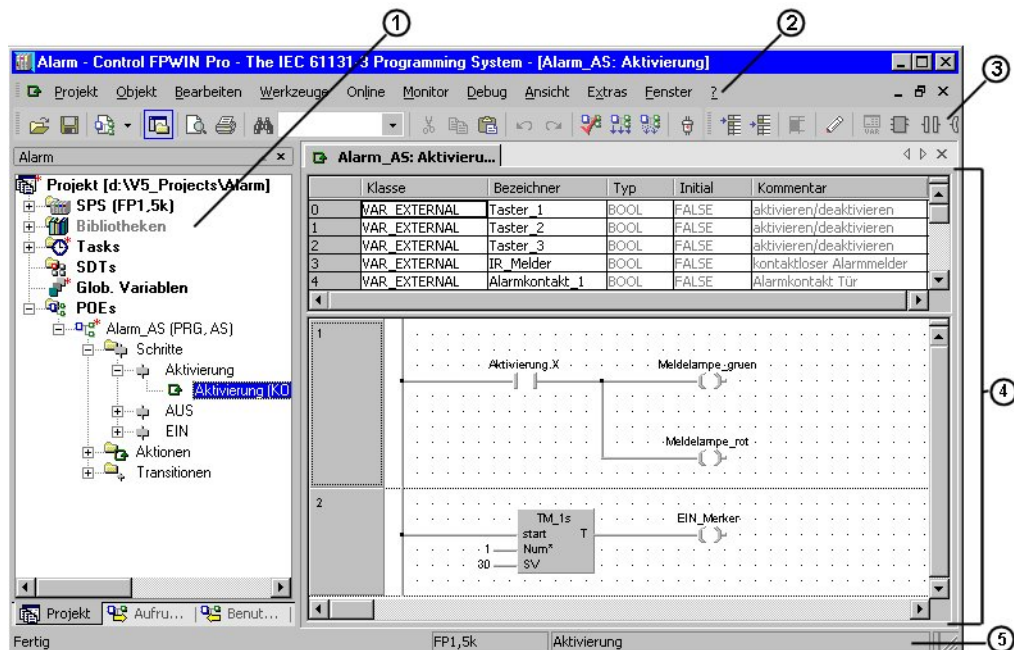


◆ Hinweis

Die Anzahl der aufgelisteten Projekte können Sie unter **Extras** → **Optionen** → **Programm-Optionen** → **Allgemein** verändern.

2.2 Hauptfenster

Die graphische Benutzeroberfläche von Control FPGWIN Pro besteht aus folgenden Elementen:



Elemente der Benutzeroberfläche

①	Projektnavigator
②	Menüleiste
③	Werkzeugleiste
④	Programmierfenster, bestehend aus Kopf und Rumpf einer POE
⑤	Statusleiste

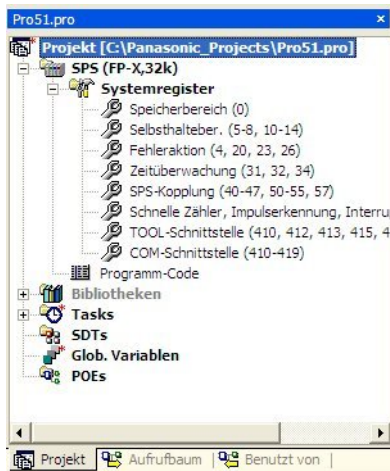
Die einzelnen Elemente der Benutzeroberfläche können beliebig auf dem Bildschirm positioniert werden. Ziehen Sie hierzu die Titelleiste des Elements mit der Maus an die gewünschte Position.

Viele Befehle der Symbolleiste und der Menüs können auch über ein Kontextmenü aufgerufen werden. Klicken Sie mit der rechten Maustaste, um ein Kontextmenü zu öffnen. Der Projektnavigator, die Programmier-Editoren (außer AS), das Fenster "Abtasten im Trace" und der Rezeptureditor verfügen über Kontextmenüs.

2.3 Projektnavigator

Durch die übersichtliche graphische Darstellung der Projekthierarchie im Navigator behalten Sie auch bei komplexen Projekten den Überblick. Im Navigator können Sie leicht auf alle Objekte des Projekts (Seite) mit einem Doppelklick zugreifen. Mit **Ansicht** → **Navigator**

aus-/einblenden oder  blenden Sie den Projektnavigator ein bzw. aus.



Projektnavigator

Der Projektnavigator besitzt drei verschiedene Ansichten:

- Projektansicht
- Aufrufbaum (siehe Seite 30)
- Benutzt von (siehe Seite 31)

Mit dem Eintrag **Zeige** des Kontextmenüs (rechte Maustaste innerhalb des Projektnavigators) können Sie die verschiedenen Ansichten des Projektnavigators konfigurieren. Dieselben Konfigurationsmöglichkeiten haben Sie mit **Extras** → **Optionen** → **Programm-Optionen** → **Navigator**.

Anzeigen der Deklaration eines Symbols oder der Referenz eines Objekts

Über das Kontext-Menü des Navigators oder des Programmier-Editors können Sie sich die Deklaration einer Variablen oder die Referenz einer POE anzeigen lassen. FPWIN Pro öffnet die entsprechende Deklaration bzw. die Referenz und markiert das Symbol.

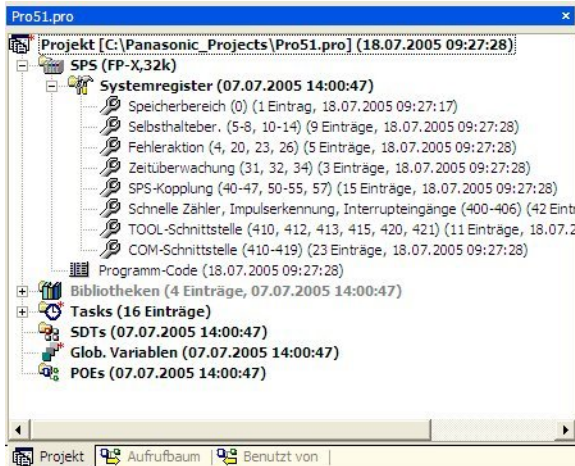


◆ Vorgehensweise

1. Variable/POE im Navigator oder im Programmier-Editor auswählen und rechts klicken
2. Im Kontextmenü Gehe zur Deklaration von **oder** Gehe zur Referenz von **wählen**
3. Mit <F3> gelangen Sie zum nächsten Eintrag

Weitere Informationen zu diesen Optionen finden Sie in der Hilfe unter dem Stichwort "Navigator".

Wählen Sie **Sortieren**, um die Reihenfolge festzulegen, in der die Objekte im Projektnavigator dargestellt werden sollen. Das Sortieren beginnt von oben nach unten.



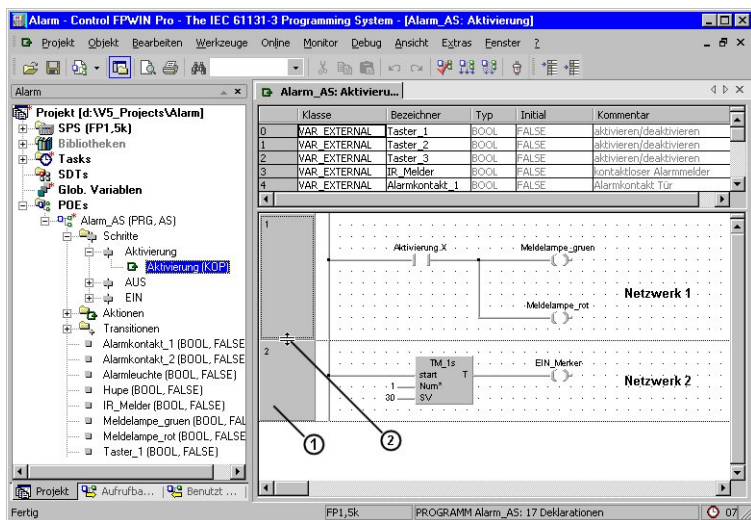
Projektnavigator mit Zusatzinformation

2.4 Programmierfenster

Im Programmierfenster geben Sie das Programm in Ihrem bevorzugten Programmier-Editor ein. Das Programmierfenster ist in einzelne Netzwerke aufgeteilt. In jedem Netzwerk steht ein Programmschritt.

Das Programmierfenster erscheint, sobald Sie im POE-Pool eine POE mit einem Doppelklick öffnen.

Links vom Programmierfenster befindet sich neben jedem Netzwerk ein Netzwerk-Infofenster.



Programmierfenster

①	Netzwerk-Infofenster
②	Netzwerkhöhe einstellen

Im Netzwerk-Infofenster können Sie die Netzwerkhöhe einstellen:



◆ Vorgehensweise

1. Cursor auf die waagerechte Linie setzen

Es erscheint ein Doppelpfeil.

2. Maustaste gedrückt halten und Cursor an die gewünschte Stelle ziehen

3. Maustaste loslassen

An der Stelle, an der Sie die Maustaste loslassen, befindet sich die neue Netzwerkhöhe.

Optimieren der Netzwerkhöhe:

1. In Netzwerk-Infofenster klicken, um das Netzwerk zu markieren

Benutzen Sie die Tasten <Umschalt> und <Strg> für die Mehrfachauswahl.

2. Werkzeuge → Netzwerkhöhe optimieren

Die ausgewählten Netzwerke werden auf Mindesthöhe verkleinert.

Ein neues Netzwerk können Sie mit folgenden Symbolen vor bzw. hinter dem bestehenden Netzwerk einfügen:



Netzwerk vorher



Netzwerk nachher

Oder verwenden Sie den Befehl **Bearbeiten → Neues Netzwerk → Anfang/Vorher/Nachher/Ende**, um ein neues Netzwerk am Anfang bzw. am Ende des Programms oder vor bzw. hinter dem bestehenden Netzwerk einzufügen

2.5 Statusleiste

Der untere Bildschirmbereich schließt mit der Statusleiste ab. In der Statusleiste finden Sie Informationen über Ihren SPS-Typ, die Uhrzeit, die Schnittstellen-Parameter und weitere Meldungen.



Statusleiste

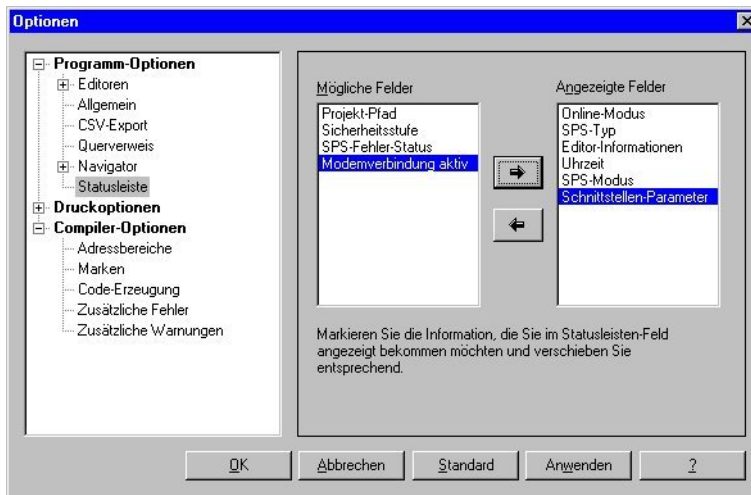
①	SPS-Typ
②	Editor-Informationen
③	Uhrzeit
④	Online-Modus
⑤	Schnittstellen-Parameter

Sie können die Statusleiste im Dialogfeld "Optionen" anpassen.



♦ Vorgehensweise

1. Auf Statusleiste doppelklicken



2. Unter "Mögliche Felder" Element auswählen, das Sie hinzufügen möchten



3. wählen

4. Unter "Angezeigte Felder" Element auswählen, das Sie entfernen möchten



5. wählen

6. [OK]

Beachten Sie hierzu die Quick-Info, die angezeigt wird, wenn Sie mit der Maus für etwa zwei Sekunden auf ein Feld der Statusleiste zeigen.



◆ Hinweis

Mit [Standard] können Sie die Statusleiste auf die Standardeinstellungen zurücksetzen.

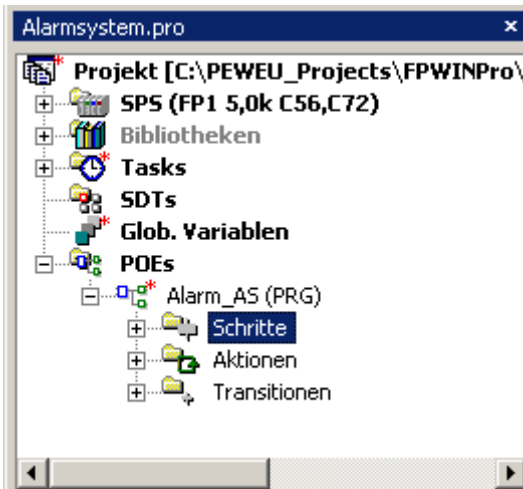
Kapitel 3

Projekte in Control FPWIN Pro


3.1 Woraus besteht ein Projekt in Control FPWIN Pro?

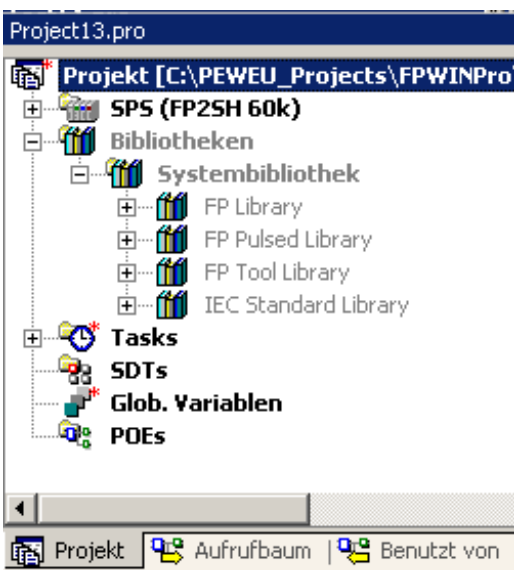
Eine Steuerungsaufgabe wird unter FPWIN Pro als Projekt bezeichnet. Ein Projekt besteht aus all den Objekten, die im Projektnavigator aufgelistet werden.

Wenn Sie ein Projekt anlegen, erscheinen zunächst folgende Objekte:




Projekt in FPWIN Pro

Objekte mit Unterpunkten oder Pools sind mit einem Pluszeichen () markiert. Mit einem Doppelklick auf die Zeile öffnen Sie einen Pool, z.B. einen Bibliotheken-Pool:



Bibliotheken-Pool öffnen

Ein Minuszeichen () bedeutet, dass alle vorhandenen Unterpunkte bereits angezeigt werden.

3.1.1 SPS

Unter **SPS** können Sie Systemregister einstellen sowie die E/A-Adressen für die Module auf den Baugruppenträgern und die dezentralen E/A-Adressen für die Slaves festlegen.

3.1.1.1 Systemregister einstellen

Systemregister sind Speicherbereiche, die z. B. für die Einstellung von Selbsthaltebereichen für Zeitgeber, Zähler, Merker und Datenregister reserviert sind.

Außerdem können Sie hier die Parameter für die SPS-Schnittstellen definieren und einstellen, wie die SPS im Fehlerfall reagieren soll.



◆ Hinweis

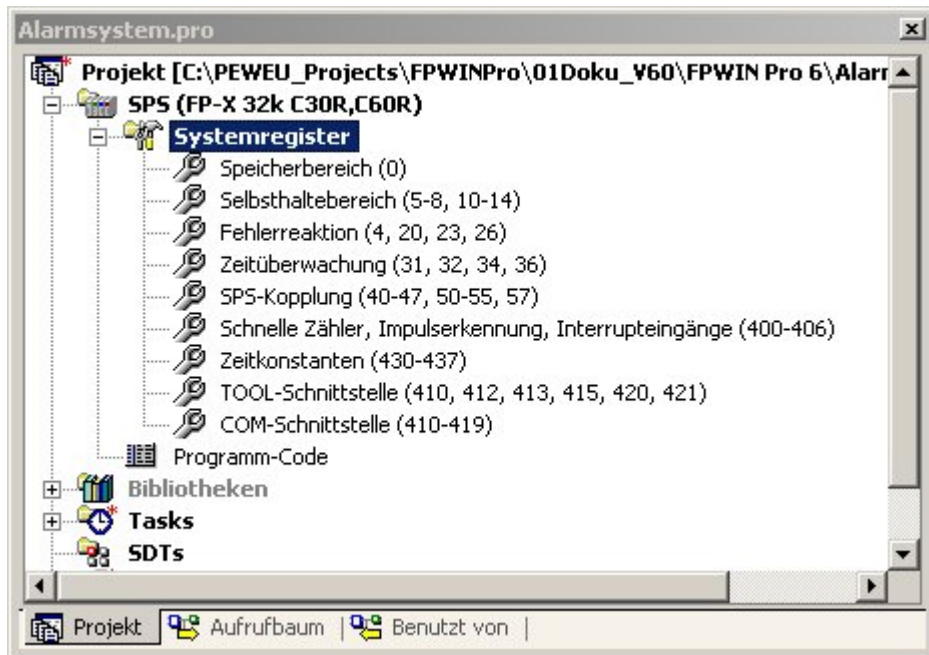
- Die Größe des Speicherbereichs hängt vom SPS-Typ ab. Die Summe aller Speicherbereiche für Systemregister, Anwenderprogramm und Maschinenprogramm darf die Gesamtgröße des jeweiligen SPS-Speichers nicht überschreiten.
- Die beiden höchsten Datenregister (vier Datenregister bei Steuerungen mit zweiter Task (siehe Seite 66)) stehen dem Benutzer nicht zur Verfügung, da sie immer im Selbsthaltebereich liegen und vom Compiler verwendet werden.



◆ Vorgehensweise

1. Im Projektnavigator auf "SPS" doppelklicken
2. Auf "Systemregister" doppelklicken

Sie erhalten eine Liste mit allen Systemregistern. Die in Klammern angegebene Nummer ist identisch mit der Systemregisternummer. Unter "Speicherbereich (0-3)" definieren Sie z.B. die Speichergröße für Maschinenprogramme. Eine Liste mit allen Systemregistern und Angaben zur Speicherbereichsgröße Ihrer SPS finden Sie in Ihrer Hardware-Beschreibung.



3. Auf beliebigen Eintrag doppelklicken
4. Einstellungen eingeben

3.1.1.2 E/A Adressen

Bei modularen Steuerungen muss jedes Modul, das auf dem Baugruppenträger angebracht wird, konfiguriert, das heißt, in die Adressliste eingetragen werden.

Sie haben zwei Möglichkeiten:

- Konfiguration im Online-Modus aus der SPS laden
- E/A-Adressen für jeden Steckplatz manuell eintragen

Konfiguration im Online-Modus aus der SPS laden



◆ Vorgehensweise

1. Online → Online Modus oder



Für diese Vorgehensweise muss die SPS installiert sein.

2. Online → SPS-Konfiguration

3. Konfiguration auswählen, die Sie rückübertragen wollen

4. [Rückübertragen]

Die in der SPS gespeicherte Konfiguration wird in FPGWIN Pro übertragen. Zwischenzeitlich neu gesteckte Module werden nicht automatisch erkannt.

oder:

[E/A Adressen registrieren]

Wenn sich die SPS im RUN Modus befindet erscheint eine Abfrage ob die SPS in den PROG Modus geschaltet werden soll. Die neu erstellte Konfiguration kann anschließend in das FPGWIN Pro Projekt übertragen werden.

E/A-Adressen für jeden Steckplatz manuell eintragen



♦ Vorgehensweise

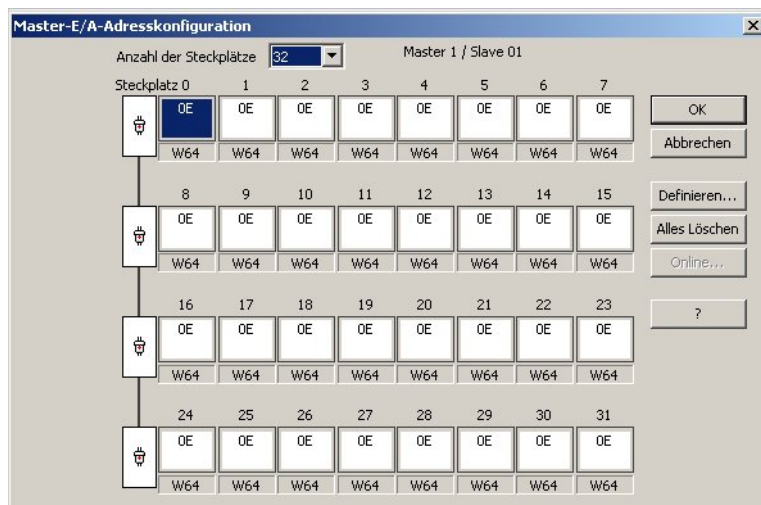
1. Im Projektnavigator auf "SPS" doppelklicken

2. Auf "MEWNET-F E/A-Adressen" doppelklicken

Es erscheint ein Dialog, in dem schematisch die Grundplatten (Backplane) mit den Steckplätzen dargestellt sind, auf die Module gesteckt werden können.

3. Unter "Anzahl der Steckplätze" die Gesamtzahl der Steckplätze einstellen

Nicht belegte Steckplätze werden mit 3 Sternchen ("***") gekennzeichnet und grau dargestellt.



4. Gewünschten Steckplatz auswählen

5. [Definieren]



6. Gewünschten Modultyp wählen

7. [OK]

8. Alle weiteren Module eintragen

9. Im Dialogfeld "Master E/A-Adresskonfiguration" [OK] wählen

Sie können die E/A-Konfiguration zusammen mit dem Projekt speichern, kompilieren und in die SPS übertragen oder, wenn Sie sich im Online-Modus befinden, die E/A-Konfiguration direkt in die SPS übertragen (**Online** → **SPS-Konfiguration** (siehe Seite 122)).

3.1.1.3 MEWNET-F-Adressliste

Jeder Slave muss konfiguriert, das heißt, in die Adressliste eingetragen werden. Bei allen anderen Steuerungen haben Sie zwei Möglichkeiten:

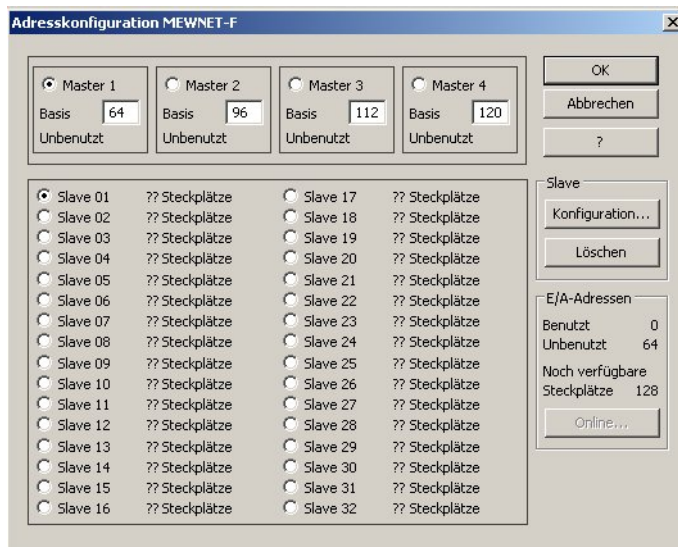
- Konfiguration im Online-Modus aus der SPS laden (siehe Seite 18)
- E/A Adressen in die Master E/A Adressen Konfigurationsliste eintragen:



♦ Vorgehensweise

1. Im Projektnavigator auf "SPS" doppelklicken

2. Auf "MEWNET-F-Adressliste" doppelklicken



3. Master (Master 1, Master 2 ...) anklicken

Für jeden Slave wird die Anzahl Steckplätze angezeigt. Wenn ein Slave vorhanden ist, steht unter "Basis" der Text "Benutzt".

4. "Basis" in Anzahl Worten eingeben (0 bis 127)

5. Gewünschten Slave anklicken

6. [Konfiguration]

Es erscheint ein Dialog, in dem schematisch die Grundplatten (Backplane) mit den Steckplätzen dargestellt sind, auf die Module gesteckt werden können.

7. Unter "Anzahl der Steckplätze" die Gesamtzahl der Steckplätze einstellen

Nicht belegte Steckplätze werden mit 3 Sternchen (***) gekennzeichnet und grau dargestellt.

8. Gewünschten Steckplatz auswählen

9. [Definieren]

10. Gewünschten Modultyp wählen

11. [OK]

12. Schritte 8 bis 11 wiederholen, bis alle Module eingetragen sind

13. Im Fenster "Slave E/A Adressen Konfiguration" OK anklicken

14. [OK]

Sie können die E/A-Konfiguration zusammen mit dem Projekt speichern, kompilieren und in die SPS übertragen oder, wenn Sie sich im Online-Modus befinden, die E/A-Konfiguration direkt in die SPS übertragen (**Online** → **SPS-Konfiguration** (siehe Seite 122)).

3.1.1.4 Programm-Code

Durch einen Doppelklick auf **Programm-Code** im Projektnavigator erscheint der Basiscode, der in die Steuerung geladen und von ihr ausgeführt wird. Der Code wird eingefügt, sobald Sie Ihr Programm auf die SPS geladen oder das Programm von der SPS heruntergeladen (siehe Seite 120) haben.

3.1.2 Bibliotheken

Mit Doppelklick auf **Bibliotheken** öffnen Sie die Liste aller verfügbaren Bibliotheken. In den Bibliotheken finden Sie Funktionen und Funktionsbausteine, die Ihnen viel Programmierarbeit abnehmen.

Wie Sie mit Bibliotheken umgehen und sich eine eigene Bibliothek erzeugen, wird in der Online-Hilfe unter "Bibliotheken" detailliert beschrieben.

3.1.3 Tasks

Tasks (siehe Seite 62) stehen an der Spitze der Projekthierarchie. In einer Task legen Sie fest, wie das Programm ausgeführt wird, z.B. zyklisch, ereignisgesteuert oder zeitgesteuert.

Sie steuern alle POEs vom Typ PRG (Programm). Die Tasks befinden sich unter **Tasks** im (Task-Pool) des Projektnavigators.

3.1.4 Strukturierter Datentyp (SDT)

Mit einem strukturierten Datentyp (SDT) können zusammengesetzte Datentypen definiert werden. Ein SDT wird zunächst unter **SDTs** definiert und anschließend in der Liste der globalen Variablen oder im POE-Kopf wie die Standarddatentypen (BOOL, INT usw.) verwendet.



◆ Hinweis

Ein SDT kann nicht als Variable eines anderen SDT verwendet werden; es gibt keine ARRAYS von SDTs.

3.1.5 Globale Variablen

Die globalen Variablen sind symbolische Namen, die den Ein-/Ausgängen zugeordnet werden. Sie sind global in dem Sinne, dass Sie in allen POEs vom Typ PRG oder FB verwendbar sind.

Weitere Informationen zum Thema Globale Variablen (siehe Seite 42) finden Sie im Kapitel über Variablen.

3.1.6 POEs

Das Objekt **POEs** (POE-Pool) ist eine Liste, die alle von Ihnen programmierten Programm-Organisationseinheiten enthält.



◆ Hinweis

Objekte, die noch nicht kompiliert oder seit dem letzten Kompilieren geändert wurden, werden im Projektnavigator mit einem Stern (*) gekennzeichnet.



Geöffneter POE-Pool

①	Objekt noch nicht kompiliert
---	------------------------------

3.2 Projekte erstellen

3.2.1 Neues Projekt erstellen

Mit Hilfe des Assistenten können Sie ein neues Projekt in Control FPGWIN Pro erstellen. Sie können einen Pfadnamen festlegen und einen beliebigen Projektnamen eingeben. Siehe auch "Erste Schritte".

Der Assistent erscheint unmittelbar nach dem Start von FPGWIN Pro. Er wird aber auch mit **Project** → **New ..** gestartet.

Weitere Informationen zum Assistenten finden Sie in der Hilfe unter dem Stichwort "Assistent".

3.2.2 Projekt von der SPS öffnen

Sie können ein Projekt, das z.B. mit **Online** → **Projekt auf der SPS speichern** in der SPS gespeichert wurde, in FPGWIN Pro übertragen und dort öffnen. Wenn Sie den Menüeintrag **Projekt** → **Öffnen von SPS** auswählen oder diese Option aus dem Projekt-Wizard (siehe Seite 6) wählen, wird ein leeres Projekt in dem ausgewählten Pfad erstellt. Anschließend wird versucht eine Verbindung zur SPS herzustellen. War dies erfolgreich, wird der Programmcode und die Projektinformationen, falls vorhanden, rückübertragen. Zu den Projektinformationen gehören: Inhalte aller Editoren, die SPS-Konfiguration, die Compiler-Optionen und die Inhalte der Benutzerbibliotheken. Sind in den Projektinformationen Benutzerbibliotheken vorhanden, werden diese im Projektverzeichnis erstellt.



◆ Hinweis

Sie können nur dann ein Projekt von der SPS öffnen, wenn Ihre SPS über einen Konfigurationsspeicher verfügt; (siehe Hardware-Beschreibung). Folgende Steuerungen besitzen diese Funktionalität:

SPS	Zusatzkarte erforderlich?	Wenn ja, welche?
FP-X	nein	
FP-Sigma	nein	
FP2	ja	Expansion Memory Unit: FP2-EM1, FP2-EM2 oder FP2-EM3
FP2SH	nein	
FP10SH	ja	ROM Operation Board: AFP6208

Wenn Sie eine andere Steuerung verwenden, wird anstelle der Projektinformationen der Programm-Code von der SPS in Ihren PC übertragen. Der Programm-Code wird anschließend in ein KOP-Programm konvertiert (vergleiche mit Import von FPWIN-GR-Dateien).



◆ Vorgehensweise

1. Projekt → Projekt von der SPS öffnen

oder diese Option im Projekt-Assistent wählen.

Im Eröffnungsfenster "Öffnet ein Projekt von der SPS" auswählen und [OK] wählen.

2. Pfadnamen eingeben oder auswählen

3. [Übernehmen]

Falls Sie die Meldung erhalten, dass Programm-Code und Code auf der SPS unterschiedlich sind, bestätigen Sie die Meldung mit [OK]. Diese Fehlermeldung erscheint, wenn das Projekt auf der Steuerung mit **Projekt → Kompiliere inkremental** (siehe Seite 109) kompiliert wurde.

3.2.3 Gesichertes Projekt wiederherstellen

Mit dieser Option des Assistenten können Sie ein Projekt öffnen, das zuvor mit **Extras → Projekt sichern (Backup)** (siehe Seite 26) gesichert und komprimiert wurde. Wählen Sie das Datenbankformat aus, das für die Sicherung verwendet wurde, suchen Sie den Pfad und wählen Sie den Dateinamen aus. Wählen Sie [Öffnen], um das gewählte Projekt wiederherzustellen.

3.2.4 Ein bestehendes Projekt öffnen

Sie können ein bestehendes Projekt mit **Projekt → Öffnen öffnen** oder ein Projekt aus der Liste der zuletzt geöffneten Projekte auswählen.

Die Liste der zuletzt geöffneten Projekte wird beim Start von Control FPWIN Pro im Eröffnungsfenster angezeigt, aber auch am Ende des Menüs "Projekt".

Wenn Sie ein Projekt öffnen, das mit einer früheren Version erstellt wurde, erhalten Sie einen Hinweis zum Konvertieren des Projekts (siehe Online-Hilfe, Stichwort "Projekte aus älteren Versionen verwenden").



◆ Hinweis

Sie können die Anzahl der kürzlich geöffneten Projekte unter **Extras → Optionen → Programm-Optionen → Allgemein, "Anzahl zuletzt geöffneter Projekte"**.

3.3 Projekte bearbeiten

Wenn Sie ein Projekt angelegt haben, können Sie dieses Projekt bearbeiten. Das heißt, Sie können folgende Bearbeitungsfunktionen ausführen:

- Öffnen
- Schließen
- Speichern bzw. Speichern unter
- Datensicherung (Backup) erstellen
- Gesicherte Daten wiederherstellen (Restore)

Die Verwendung der Befehle Öffnen, Schließen, Speichern oder Speichern unter entspricht den unter Windows üblichen Standard-Befehlen.

3.3.1 Datensicherung (Backup) erstellen

Mit **Extras** → **Projekt sichern (Backup)** können Sie Projekte in zwei verschiedenen komprimierten Formaten speichern. Im PCD-Format (packed data base format) wird die komplette Projektdatenbank gesichert. Im PCE-Format (packed export format) werden nur die Projektdaten mit Bibliotheken gesichert.



◆ Hinweis

- Beim Erstellen der Datensicherung werden die Projektdaten komprimiert.
- Datensicherungen, die Sie mit **Extras** → **Projekt sichern (Backup)** erstellt haben, können Sie nur mit **Extras** → **Projekt wiederherstellen (Restore)** (siehe Seite 26) wieder öffnen.

3.3.2 Projekt wiederherstellen (Restore)

Mit **Extras** → **Projekt wiederherstellen (Restore)** oder im Assistenten mit der Option "Stellt ein gesichertes Projekt wieder her" können Sie ein Projekt laden und automatisch dekomprimieren lassen, das Sie zuvor mit **Extras** → **Projekt sichern** im komprimierten Datenbankformat (*.PCD) oder im komprimierten Exportformat (*.PCE) gespeichert haben.

3.4 Objekte bearbeiten

Alle Objekte, die im Projektnavigator stehen, können Sie mit den Untermenüs von "Objekt" und "Bearbeiten" editieren.

Mit dem Menü "Objekt" nehmen Sie Einfluß auf äußere Merkmale eines Objekts, das heißt, Sie können es z.B.:

- Öffnen
- Umbenennen
- Mit einem Kommentar versehen
- Nach Syntaxfehlern durchsuchen lassen
- Ausdrucken und
- Im- bzw. exportieren

Außerdem können Sie sich:

- Objekteigenschaften
- Aufrufbaum und Liste anzeigen lassen.

Mit dem Menü "Bearbeiten" bearbeiten Sie den Inhalt eines Objekts und können neue Objekte erstellen.

3.5 Passwörter und Sicherheitsstufen

Um den Zugriff auf die einzelnen Objekte Ihres Projekts zu steuern, können Sie jedem Objekt eine Sicherheitsstufe zuweisen. Zu jeder Sicherheitsstufe definieren Sie vorher ein Paßwort. Nur Personen, die das Paßwort der jeweiligen Sicherheitsstufe kennen, können z.B. das entsprechende Objekt ändern.



♦ Vorgehensweise

1. Projekt → Passwörter ändern

2. Setzen Sie für jede Sicherheitsstufe ein Paßwort

Wenn Sie zum ersten Mal ein Passwort setzen, geben Sie nur unter "Neues Passwort" und "Bestätigung" Ihre Passwörter nacheinander für jede Sicherheitsstufe ein und wählen jedes Mal [Ändern]. Sie erhalten vom System eine Bestätigung, wenn das Passwort geändert worden ist.

3. Im Projektnavigator das Objekt auswählen, dessen Zugriff Sie schützen wollen

4. Objekt → Eigenschaften

Im Dialogfeld "Objekt-Eigenschaften" können nur Sicherheitsstufen eingegeben werden, die bereits definiert worden sind.

5. Auf die gewünschte Sicherheitsstufe und ggf. auf "Lesen erlaubt für niedrigere Stufen" klicken

Nach einem Neustart ist der Zugriffsschutz aktiviert.

3.5.1 Zugriff auf geschützte Objekte

Mit **Projekt → Sicherheitsstufe ändern** erhalten Sie Zugriff auf ein durch Passwort geschütztes Objekt.



◆ Hinweis

- Für Objekte, deren Sicherheitsstufe höher ist als die des Projekts, können Sie die Sicherheitsstufe nicht ändern.
- Wenn Sie nur auf der niedrigeren Sicherheitsstufe des Projekts zugreifen dürfen, können Sie Objekte mit höheren Sicherheitsstufen lesen, wenn "Lesen erlaubt für niedrigere Stufen" unter Objekt → Eigenschaften des jeweiligen Objekts definiert wurde.



◆ Vorgehensweise

1. Objekt im Projektnavigator anklicken
2. Projekt → Sicherheitsstufe ändern

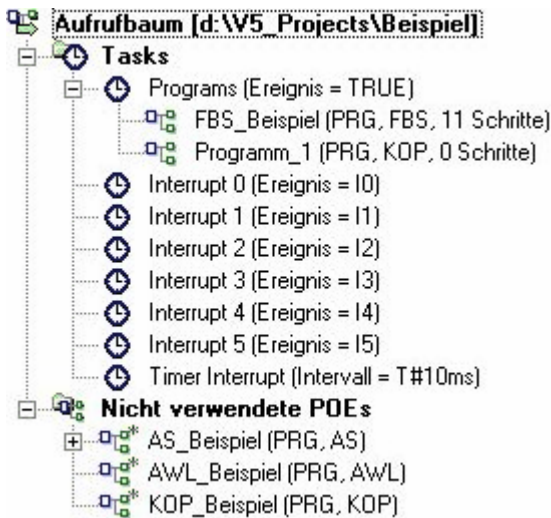


3. Gewünschte Sicherheitsstufe auswählen und Passwort eingeben
4. [OK]

Sie haben jetzt wieder Zugriff auf das Objekt und können es bei Bedarf ändern. Der Zugriffsschutz bleibt auch nach der Änderung erhalten. Sie brauchen ihn nicht neu zu definieren.

3.6 Aufrufbaum

Die Ansicht "Aufrufbaum" enthält die Einträge "Tasks" und "Nicht verwendete POEs". Unter "Tasks" wird dargestellt, welche POEs und globalen Variablen den jeweiligen Tasks zugeordnet sind. Die Objekthierarchie zeigt Ihnen z.B., ob Funktionen oder Funktionsbausteine in einem Programm verwendet wurden und in welcher Abhängigkeit sie zueinander stehen.



Aufrufbaum



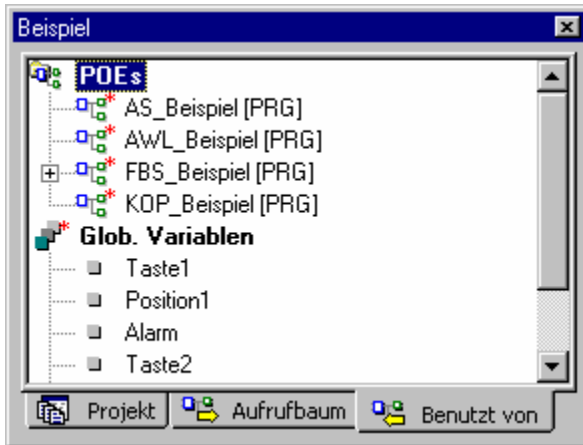
◆ Hinweis

- Die benutzten globalen Variablen werden nur angezeigt, wenn Sie im Kontextmenü des Navigators Zeige → Globale Variablen gewählt haben.
- Weitere Informationen zu den Einstellungen des Aufrufbaums, z.B. zur Anzeige der globalen Variablen, siehe "Objekte im Aufrufbaum anzeigen"

3.7 Benutzt-von-Liste

Die Ansicht "Benutzt von" enthält die Einträge "Globale Variablen" und "POEs".

- POE-Pool mit seinen POEs als Knoten
- die globale Variablenliste mit ihren globalen Variablen als Knoten.



"Benutzt-von"-Liste

Die Darstellung ist hierarchisch, wobei die Verzweigung auf der untersten Hierarchie-Ebene beginnt und auf der obersten Hierarchie-Ebene endet.


3.8 Kontrolle

Programme und Variablenlisten sollten Sie von FPDWIN Pro überprüfen lassen, bevor Sie sie kompilieren. Wir empfehlen, die Kontrolle bei jeder einzelnen Komponente vorzunehmen, nachdem Sie sie fertiggestellt haben.



♦ Vorgehensweise

Kontrolle

1. Im Projektnavigator das Objekt auswählen, das Sie kontrollieren wollen
2. Objekt → Kontrolle oder  oder Kontextmenü im Navigator → Kontrolle
Das Dialogfeld "Compiler-Meldungen" wird angezeigt.

Warnung suchen

1. Im Dialogfeld "Compiler-Meldungen" gewünschte Fehlermeldung/Warnung markieren



2. [Anzeigen] oder

Doppelklick auf die Fehlermeldung.

FPDWIN Pro öffnet automatisch den Programmteil, in dem der Fehler gefunden wurde und markiert diesen.



♦ Hinweis

Wenn mehrere Fehler auftreten beheben Sie immer den Fehler als erstes, der zuerst in den Compiler-Meldungen genannt wird und wiederholen Sie Objekt → Kontrolle. Alle anderen Fehler sind unter Umständen Folgefehler. Klicken Sie auf [-> Fehler], um zum nächsten Fehler zu gelangen.

Kapitel 4

Programm-Organisationseinheiten

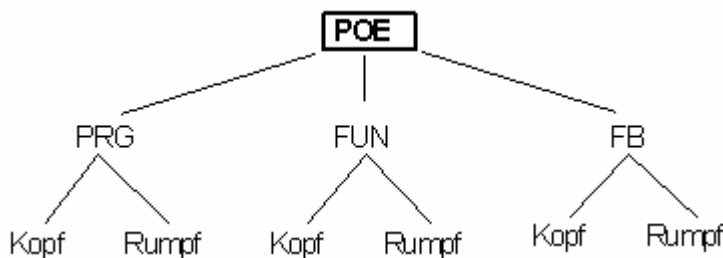
4.1 Programm-Organisationseinheiten

Programm-Organisations-Einheiten (POEs) sind die Bausteine eines Programms in FFWIN Pro. Sie enthalten das SPS-Steuerprogramm. Im Gegensatz zur herkömmlichen Programmierung besteht ein Programm in FFWIN Pro nicht aus einem einzelnen Programm, das gegebenenfalls Unterprogramme aufruft, sondern aus mehreren Teilprogrammen. Jedes Teilprogramm ist in sich abgeschlossen und bearbeitet eine Aufgabe. Je nachdem, um welche Aufgabe es sich handelt, wird der entsprechende POE-Typ ausgewählt.

Wir unterscheiden zwischen drei POE-Typen (Programmklassen):

- Programm (siehe Seite 35) (PRG)
- Funktion (siehe Seite 36) (FUN)
- Funktionsbausteine (siehe Seite 37) (FB)

Jede POE, gleich welchen Typs, ist unterteilt in einen Kopf und einen Rumpf. Beide werden in einem gemeinsamen Programmierfenster angezeigt.



Die Aufteilung in Kopf und Rumpf ergibt sich aus der IEC-Philosophie, nach der alle Variablen in einer Liste deklariert werden und statt physikalischer Adressen diese Deklarationen (symbolische Namen) im Programm verwendet werden.

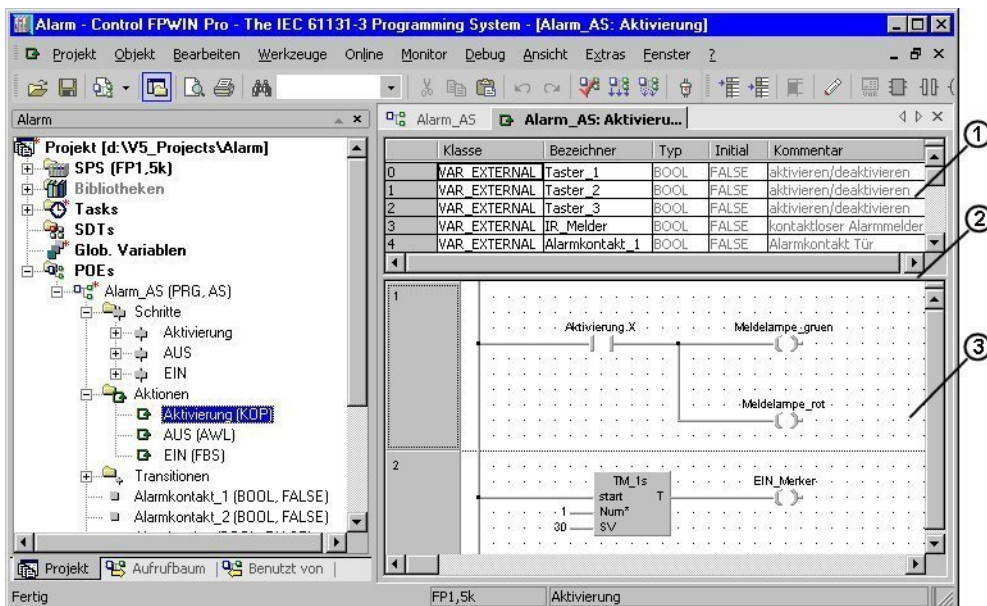
Deklarationen haben den Vorteil, dass der Compiler die Adressverwaltung übernimmt und Sie Adressänderungen nur einmal im POE-Kopf vornehmen müssen (sofern Sie eine Adresse definiert haben). Die Programme bleiben unberührt.

In die Liste der globalen Variablen werden alle Variablen eingetragen, die eine Ein-/Ausgangsadresse bzw. eine SPS-Adresse haben und die im gesamten Projekt (in allen POEs) verwendet werden können. Im POE-Kopf werden ausschließlich lokale Variablen deklariert, die Sie nur in dem zugehörigen POE-Rumpf verwenden können.


Im POE-Rumpf steht die Programmlogik. Die Programmlogik kann in einer dieser Programmiersprachen geschrieben werden:

- Kontaktplan (KOP) Funktionsbausteinsprache (FBS) (siehe Seite 76)
- Strukturierter Text (ST) (siehe Seite 84)
- Anweisungsliste (AWL) (siehe Seite 95)
- Ablaufsprache (AS-Editor) (siehe Seite 97)

Im POE-Kopf werden die lokalen Variablen eingegeben, das heißt, die Verbindung zu den Ein-/Ausgängen und den internen Speichern wird definiert. Im POE-Rumpf steht die Programmlogik. Sowohl POE-Kopf als auch der Rumpf werden in einem gemeinsamen, teilbaren Programmierfenster angezeigt.



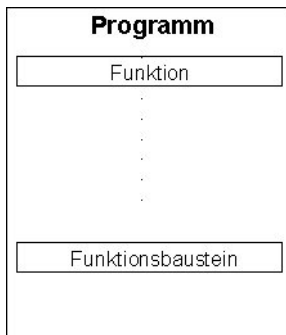
Komponenten einer POE

①	Kopf der POE
②	Ziehen Sie diese Leiste mit der Maus (Zeiger ) , um die Höhe der Fenster für Kopf und Rumpf festzulegen.
③	Rumpf der POE

4.1.1 Programme

Bei der Steuerungsaufgabe kann es sich z.B. darum handeln, eine Temperatur zu messen, sie weiterzuverarbeiten, (z.B. mit einem Sollwert zu vergleichen) und entsprechende Ausgangsdaten auszugeben, um Peripheriegeräte, wie z.B. eine Heizung, zu steuern.

Ein Programm stellt die höchste Ebene in der POE-Hierarchie dar. In einem PRG können Sie POEs vom Typ Funktion und Funktionsbausteine aufrufen.



◆ Hinweis

Ein Programm kann ausschließlich von einer Task (siehe Seite 62) aufgerufen werden. Funktionen und Funktionsbausteine können wiederum Funktionen (FUN) und Funktionsbausteine (FB) aufrufen, aber keine Programme.

4.1.2 Funktion (FUN)

Funktionen (FUN) sind Programmorganisationseinheiten (POEs), die nach dem Ausführen ein Datenelement als Funktionsergebnis und beliebig viele Ausgangswerte der Klassen VAR_OUTPUT und VAR_IN_OUT liefern. Wird als Ergebnistyp **VOID** festgelegt, hat die Funktion kein Ergebnis.

Ein Zugriff auf globale Variablen ist mit VAR_EXTERNAL, VAR_EXTERNAL_RETAIN oder VAR_EXTERNAL_CONSTANT möglich.

Funktionen enthalten keine internen Zustandsinformationen, das heißt, der Aufruf einer Funktion mit gleichen Eingangswerten der Klassen VAR_INPUT und VAR_IN_OUT liefert immer das gleiche Funktionsergebnis und die gleichen Ausgangswerte der Klassen VAR_OUTPUT und VAR_IN_OUT.

FPWINPro unterscheidet zwei Arten von Funktionen:

1. Funktionen der Systembibliotheken

- FP Library
- FP Pulsed Library
- FP Tool Library
- IEC Standard Library

2. Anwenderfunktionen (siehe Seite 127)

Sie können auch eigene Anwenderfunktionen schreiben und diese in eine Benutzerbibliothek einfügen. Wenn die Benutzerbibliothek installiert ist, können Sie deren Funktionen in allen Projekten verwenden.

Anwenderfunktionen können in folgenden Programmiersprachen geschrieben werden:

- Kontaktplan (KOP) Funktionsbausteinsprache (FBS) (siehe Seite 76)

- Strukturierter Text (siehe Seite 84) (ST)
- Anweisungsliste (siehe Seite 95) (IL)



◆ Hinweis

- **Funktionen können einer Task nicht direkt zugeordnet werden, da sie immer nur von einem Programm oder einem Funktionsbaustein aufgerufen werden können.**
- **Rekursive Aufrufe einer Funktion sind nicht möglich.**
- **Es sind maximal 5 geschachtelte Funktionsaufrufe zulässig, z. B. Fun1 (Fun2 (Fun3 (Fun4 (Fun5 (x))))).**
- **Anwenderfunktionen benötigen keine Variablen als Funktionsergebnis und an den Ausgängen.**
Ausnahme:
In den Programmiereditoren ST und AWL beim Aufruf der Funktion ohne Formelparameter.
- **Beim Aufruf einer Funktion mit Formalparameter im ST-Editor gilt:**
 - Die Reihenfolge der Parameter ist ohne Bedeutung.
 - Bei Anwenderfunktionen mit EN/ENO kann der EN-Eingang weggelassen werden; in diesem Fall wird der Eingang mit TRUE initialisiert.

4.1.3 Funktionsbaustein (FB)

Auch bei Funktionsbausteinen handelt es sich um kleine Programmteile. Im Gegensatz zu Funktionen verfügen Funktionsbausteine über einen eigenen Speicherbereich, in dem Werte zwischengespeichert werden können. Gleiche Eingangswerte liefern je nach zwischengespeichertem Wert, der addiert, subtrahiert usw. werden kann, unterschiedliche Ergebnisse. Sie können ein und denselben FB in Ihrem PRG beliebig oft verwenden. Aus diesem Grund wird jedes Mal, wenn Sie einen FB aufrufen, eine Kopie angelegt. Dieser Kopie (Instanz) geben Sie einen Namen, damit nicht die Werte des zuvor verwendeten FB vom gleichen Typ überschrieben bzw. weiterverarbeitet werden.

FPWINPro unterscheidet zwei Arten von Funktionsbausteinen:

1. Funktionsbausteine der Systembibliotheken
 - Standard-Funktionsbausteine
 - FP-Funktionsbausteine
2. Anwender-Funktionsbausteine (siehe Seite 128)

Diese können in folgenden Programmiersprachen geschrieben werden:

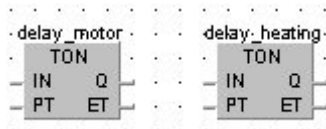
- Kontaktplan (KOP) Funktionsbausteinsprache (FBS) (siehe Seite 76)
- Strukturierter Text (siehe Seite 84) (ST)

- Anweisungsliste (siehe Seite 95) (IL)



◆ BEISPIEL

Wenn Sie den Funktionsbaustein "E_TON" (Zeitgeber mit Einschaltverzögerung) aus der IEC-Standardbibliothek zum ersten Mal aufrufen, um einen Motor verzögert einzuschalten, nennen Sie ihn "delay_motor". Den gleichen Funktionsbaustein nennen Sie beim nächsten Aufruf z. B. "delay_heating" usw.



◆ Hinweis


- FB können einer Task nicht direkt zugeordnet werden, da sie immer nur von einem Programm oder einer Funktion aufgerufen werden können.
- Rekursive Aufrufe von Funktionsbausteinen sind nicht möglich.
- Es sind maximal 5 geschachtelte Funktionsaufrufe zulässig, z. B. Fun1 (Fun2 (Fun3 (Fun4 (Fun5 (x))))).
- Je FB dürfen Sie insgesamt bis zu 40 Ein- und Ausgangsvariablen definieren.

4.2 POE neu anlegen

Die Art und Weise, wie POEs für ein Programm, eine Funktion und einen Funktionsbaustein erstellt werden, unterscheidet sich nur geringfügig.



◆ Vorgehensweise

1. Bearbeiten → Neu → POE oder 

Das Dialogfeld "POE anlegen (Projekt)" wird geöffnet.

2. Wählen Sie "Programm", "Funktion" oder "Funktionsbaustein"
3. Befolgen Sie die Anweisungen in den Dialogfeldern, um die POE zu erstellen.

Kapitel 5

Variablen

5.1 Variablen

Variablen sind symbolische Namen für Ein- und Ausgänge sowie für Speicherbereiche innerhalb der SPS. Symbolische Namen werden im Programm anstatt der physikalischen Adressen verwendet.

Bei den Variablen wird unterschieden zwischen globalen (siehe Seite 42) und lokalen (siehe Seite 44) Variablen:

- Globale Variablen sind die direkte Verbindung zur Außenwelt, (Ein-/Ausgänge zum Prozess, sonstige Variablen zur IOP oder zur Visualisierung). Globale Variablen werden in der "Globalen Variablenliste" deklariert.
- Lokale Variablen sind z. B. Platzhalter für Zwischenergebnisse, die das System an beliebiger Stelle speichert. Sie werden im POE-Kopf der jeweiligen POE deklariert.

Einmal deklarierte Variablen können Sie später ändern, im Programm direkt dem Ein-/Ausgang usw. zuweisen oder zu so genannten Variablengruppen, wie z. B. Arrays oder strukturierten Datentypen, zusammenfassen. Für komplexe Steuervorgänge können Sie auch Rezepturen erstellen.

Mit der so genannten Querverweisliste haben Sie die Möglichkeit, sich alle Variablendeklarationen und deren Abhängigkeiten auflisten zu lassen und sich so eine Übersicht über die Variablen zu verschaffen.

5.1.1 Globale Variablen

Die globalen Variablen sind symbolische Namen, die den Ein-/Ausgängen zugeordnet werden. Sie sind global in dem Sinne, dass Sie in allen POEs vom Typ PRG oder FB verwendbar sind. Weitere Informationen zum Thema "Globale Variablen finden Sie im Kapitel über Variablen. Sie deklarieren eine globale Variable immer dann,

- wenn die SPS auf einen bestimmten Ein-/Ausgang zugreifen soll, z. B. X0 oder Y0,
- wenn die SPS auf einen fest definierten internen Speicherbereich zugreifen soll (Bediengeräte sollen z. B. auf das Datenregister DT0 zugreifen) oder wenn eine Variable in mehreren Programm-Organisationseinheiten verwendet wird.
- Globale Variablen können Sie nämlich in jeden POE-Kopf übernehmen (über VAR_EXTERNAL).



◆ Hinweis

Bei globalen Variablen, die keinen Zugriff auf Ein-/Ausgänge bzw. Speicherbereiche haben, sondern in der Globalen Variablenliste deklariert werden, weil Sie sie in anderen POEs verwenden möchten, empfiehlt es sich, keine Adressen anzugeben. Die Adressvergabe wird dann automatisch vom Compiler übernommen. Sie vermeiden somit Fehler durch doppelt belegte Adressen. Vom Compiler vergebene Adressen werden außerdem beim Wechsel des SPS-Typs automatisch angepasst.

Weitere Informationen zu den Formaten der Adressen finden Sie in der Online-Hilfe unter dem Stichwort "Adressformate".

Bei den globalen Variablen wird zwischen drei Klassen unterschieden:

- **VAR_GLOBAL**
 - Der Wert kann vom SPS-Programm geändert werden; diese Klasse ist voreingestellt. Die Variable wird mit dem Wert initialisiert, der in der globalen Variablenliste definiert ist, nachdem Sie die SPS neu gestartet oder vom PROG- in den RUN-Modus geschaltet haben, oder nach einem Stromausfall.
- **VAR_GLOBAL_RETAIN**
 - wie VAR_GLOBAL, aber **selbsthaltend**, d.h., der Wert bleibt bei einem Neustart der SPS, bei einem Stromausfall oder beim Umschalten vom PROG- in den RUN-Modus erhalten. Initialisierung mit dem Wert aus der globalen Variablenliste nur nach einem Kaltstart, das heißt, wenn ein Programm auf die Steuerung geladen oder der Schalter INTIAL im PROG-Modus aktiviert wurde.

Hinweis:

Wenn Sie unter Extras → Optionen → Compiler-Optionen → Code-Erzeugung das Kontrollfeld "Haltende Variablen im Anwender-Adressbereich nicht initialisieren" aktiviert haben, werden haltende Variablen, denen der Anwender Adressen zugewiesen hat, nicht erneut initialisiert.

- **VAR_GLOBAL_CONSTANT**
 - Der Wert der globalen Variablen enthält eine Konstante und kann **nicht** vom SPS-Programm geändert werden. Sie belegt keine Adresse in der SPS, sondern die Konstante wird in den Programmcode eingefügt.
 -

5.1.1.1 Adressen

Mit Adressen legen Sie fest, welcher Eingang oder interne Speicherbereich gelesen und welcher Ausgang gesteuert werden soll. Für die Adressierung von Panasonic Steuerungen werden sowohl Wort- als auch Bitadressen verwendet.

Unter FPWIN Pro können Sie Adressen im FP Format, die so genannte Hardware-Adresse, oder im IEC-Format eingeben. Beide Adresstypen werden in der Hilfe unter dem Stichwort "Adressdefinition" erläutert.

5.1.1.2 Globale Variablen deklarieren

Bevor Sie mit dem Deklarieren von globalen Variablen beginnen, möchten wir Ihnen kurz die Felder in der Liste der globalen Variablen erläutern.

Ein Beispiel zur Deklaration der globalen Variablenliste finden Sie in der Hilfe unter dem Stichwort "Globale Variablen deklarieren".

Die Felder in der globalen Variablenliste haben folgende Bedeutung:

	1	2	3	4	5	6	7	8
	Klasse	Bezeichner	FP-Adresse	IEC-Adresse	Typ	Initial	Autoextern	Kommentar
0	VAR_GLOBAL	Taster_1	X0	%IX0.0	BOOL	FALSE	✓	aktivieren/ deaktivieren
1	VAR_GLOBAL	Taster_2	X1	%IX0.1	BOOL	FALSE	✓	aktivieren/ deaktivieren
2	VAR_GLOBAL	Taster_3	X2	%IX0.2	BOOL	FALSE	✓	aktivieren/ deaktivieren

①	Klasse gibt die Variablenklasse an, z.B. VAR_GLOBAL, VAR_GLOBAL_CONSTANT usw. an, nähere Informationen finden Sie unter Globale Variablen (siehe Seite 42)
②	Bezeichner ist der symbolische Name, der im Programm verwendet wird. Bezeichner dürfen nicht mit einer Zahl beginnen!
③ ④	FP Adresse und IEC-Adresse ist die physikalische Adresse, die Sie der Variablen zuweisen. <ul style="list-style-type: none"> Geben Sie nur für Ein- und Ausgänge der SPS eine Adresse ein oder, wenn für die Variable ein bestimmtes Datenregister zwingend ist. In allen anderen Fällen geben Sie keine Adresse ein. Es ist gleich, welches Adressfeld (FP oder IEC) Sie verwenden, das andere Feld wird automatisch ausgefüllt, wenn Sie zum nächsten Feld gehen. Wenn Sie kein Adressfeld ausfüllen, werden die Adressen vom Compiler zugewiesen. Hinweis: Ausführliche Informationen zu diesen Adressformaten finden Sie in der Hilfe unter dem Stichwort IEC-Adressformate ⇔ FP Adressen.
⑤	Typ ist der Datentyp (BOOL, WORD, ...). Sie können bei Bedarf einen anderen Typ aus der Liste der Datentypen wählen.
⑥	Initial ist der Anfangswert, den die Variable beim Start der SPS zugewiesen bekommt. Sie können diesen Wert bei Bedarf ändern.
⑦	Autoextern fügt die globale Variable automatisch in den Kopf aller neu zu erstellenden POE ein. Aktivieren Sie das Kontrollfeld <input checked="" type="checkbox"/> Autoextern-Feld kopiert Deklaration sofort in POE-Köpfe unter Extras → Optionen → Programm-Optionen → Editoren → Deklarations-Editoren , um die Variable automatisch in alle im Projektnavigator angezeigten POE einzufügen.
⑧	Kommentar z.B. für die Beschreibung der Variablen



◆ Hinweis

- **Bezeichner dürfen nicht mit einer Zahl beginnen.**
- **FP Adressen (X0, Y1 usw.) sind feststehende Begriffe und dürfen nicht als Bezeichner verwendet werden.**

5.1.2 Lokale Variablen

Lokale Variablen werden im POE-Kopf deklariert und können auch nur in dem zugehörigen POE-Rumpf verwendet werden. Einen Austausch mit anderen POE erlauben lokale Variablen nicht. Im POE-Kopf unterscheidet man zwischen den Variablen, die Sie aus der globalen Variablenliste in den POE-Kopf übernehmen und denen, die Sie speziell für einen bestimmten POE-Rumpf im dazugehörigen POE-Kopf deklarieren.

Welchen Variablentyp Sie im POE-Kopf definieren können, hängt vom POE-Typ ab.



◆ Hinweis

Mit Extras → Optionen → Compiler-Optionen → Adressbereiche (siehe Seite 106) können Sie die Adressbereiche angeben, die der Compiler für lokale Variablen reserviert. Der verbleibende Adressbereich wird für globale Variablen genutzt.

5.1.2.1 Globale Variablen in den POE-Kopf übernehmen

Wenn Sie beim Deklarieren der globalen Variablen im Feld "Autoextern" ein Häkchen eingetragen haben, werden alle globalen Variablen automatisch in jeder neu erstellten POE in den POE-Kopf übernommen.

Sollten Sie dieses Feld nicht aktiviert haben, können Sie alle oder auch nur bestimmte globale Variablen nachträglich mit **Extras → Externe Variablen deklarieren** übernehmen.



◆ Vorgehensweise

1. Im Projektnavigator auf "Globale Variablen" doppelklicken

Die Liste der globalen Variablen wird geöffnet.

2. Gewünschte Variablen selektieren

3. Extras → Externe Variablen deklarieren

Eine Bestätigungsaufforderung wird angezeigt.

4. [OK] wählen

Die selektierten globalen Variablen werden als VAR_EXTERNAL in die Köpfe aller aktuell im POE-Pool vorhandenen POE vom Typ PRG und FB eingefügt.



◆ Hinweis

Bearbeiten → Rückgängig **ist für** Extras → Externe Variablen deklarieren nicht möglich. Sie können extern deklarierte Variablen nur mit Extras → Nicht verwendete Variablen löschen (siehe Seite 46) aus den POE-Köpfen entfernen.

Variablen, die Sie aus der Liste der globalen Variablen in einen POE-Kopf übernehmen haben drei Klassen:

- **VAR_EXTERNAL**

Variable vom Typ VAR_GLOBAL, die aus der "Globalen Variablenliste" übernommen worden ist. Eine Variable vom Typ VAR_EXTERNAL kann ausschließlich in den Kopf einer POE vom Typ PRG oder FB übernommen werden. Sie werden auf den Initialwert gesetzt, nachdem Sie die SPS neu gestartet oder vom PROG- in den RUN-Modus geschaltet haben, oder nach einem Stromausfall.

- **VAR_EXTERNAL_RETAIN**

Variable vom Typ VAR_GLOBAL_RETAIN, die aus der "Globalen Variablenliste" übernommen worden ist. Eine Variable vom Typ VAR_EXTERNAL_RETAIN ist **selbsthaltend** und wird nur nach einem Kaltstart auf den Initialwert gesetzt, das heißt, wenn ein Programm auf die Steuerung geladen oder der Schalter INTIAL im PROG-Modus aktiviert wurde. Wenn Sie unter **Extras → Optionen → Compiler-Optionen → Code-Erzeugung** die Option "Haltende Variablen im Anwender-Adresssbereich nicht initialisieren" gewählt haben, werden Variablen, denen der Anwender Adressen zugewiesen hat, allerdings nicht initialisiert.

- **VAR_EXTERNAL_CONSTANT**

Eine Variable vom Typ VAR_GLOBAL_CONSTANT, die aus der "Globalen Variablenliste" übernommen worden ist. Eine Variable vom Typ VAR_EXTERNAL_CONSTANT kann ausschließlich in den Kopf einer POE vom Typ PRG oder FB übernommen werden. Sie belegt keine Adresse in der SPS, sondern die Konstante wird in den Programmcode eingefügt.

Sollten Sie überflüssige Variablen aus der Liste der globalen Variablen übernommen haben, können Sie diese auch wieder aus dem POE-Kopf löschen.

5.1.2.2 Überflüssige globale Variablen im POE-Kopf löschen

Mit **Extras → Nicht verwendete Variablen löschen** können Sie nicht verwendete Variablen aus den Köpfen der im POE-Pool vorhandenen POEs und/oder aus der globalen Variablenliste löschen, wenn diese Variablen in den zugehörigen POE-Rümpfen nicht benutzt werden. Nicht benutzte Variablen können Sie in POE-Köpfen und/oder der globalen Variablenliste nur löschen, wenn sie nicht in den jeweiligen POE-Rümpfen verwendet worden sind.

Sie haben folgende Möglichkeiten:

Löschen	Erklärung
Lokale Variablen	Alle lokalen Variablen in den POE-Köpfen werden gelöscht, wenn sie nicht in den zugehörigen POE-Rümpfen verwendet werden.
externe Variablen	Alle Variablen, die in den POE-Köpfen als externe Variablen deklariert wurden, werden gelöscht, wenn sie nicht in den zugehörigen POE-Rümpfen verwendet werden.
Globale Variablen	Alle Variablen aus der globalen Variablenliste werden gelöscht, wenn sie nicht in POE-Köpfen verwendet werden.



◆ Hinweis

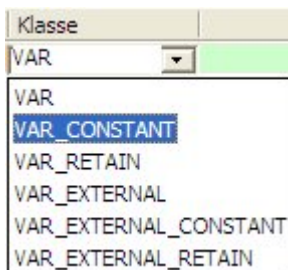
POE in Benutzerbibliotheken werden nicht berücksichtigt.

Standardeinstellung:

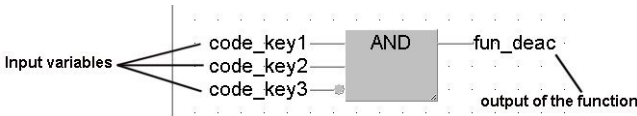
Alle Variablen in allen POE löschen.

5.1.2.3 Lokale Variablen deklarieren

Lokale Variablen deklarieren Sie ähnlich wie globale Variablen, jedoch im POE-Kopf:



Adressen für lokale Variablen werden immer automatisch vom System vergeben. Bei den lokalen Variablen, werden sieben Klassen angeboten. Welche Klasse Sie auswählen können, hängt vom POE-Typ ab.

Variablentyp	Erklärung																								
VAR	Variable, die Sie für jeden POE-Typ deklarieren können, z. B. zum Speichern von Zwischenergebnissen. Die Werte von VAR bleiben von einem Aufruf zum nächsten erhalten. VAR wird erst auf den Initialwert gesetzt, nachdem Sie die SPS vom PROG- in den RUN-Modus geschaltet haben, oder nach einem Stromausfall.																								
VAR_CONSTANT	wie VAR, enthält aber eine Konstante. VAR_CONSTANT belegt keine Adresse, sondern die Konstante wird in den Programmcode eingefügt.																								
VAR_RETAIN	<p>wie VAR, aber selbsthaltend, d.h., der Wert der Variablen bleibt auch bei einem Stromausfall erhalten. VAR_RETAIN wird nur nach einem Kaltstart auf den Initialwert gesetzt, das heißt, wenn ein Programm auf die Steuerung geladen oder Schalter INITIAL im PROG-Modus aktiviert wurde.</p> <p>Wenn Sie unter Extras → Optionen → Compiler-Optionen → Code-Erzeugung das Kontrollfeld "Haltende Variablen im Anwender-Adressbereich nicht initialisieren" aktiviert haben, werden haltende Variablen, denen der Anwender Adressen zugewiesen hat, nicht erneut initialisiert.</p>																								
VAR_INPUT	<p>Eingangsvariable, die für die Parameterübergabe verwendet wird. Die aufrufende POE übergibt einen Wert an eine Funktion oder einen Funktionsbaustein (nicht an ein Programm). Eingangsvariablen müssen im zugehörigen Kopf der Funktion/des Funktionsbausteins deklariert werden. Sie können den Wert einer Eingangsvariablen lesen, aber nicht setzen (Ausnahme: "Forcen", siehe Online-Hilfe).</p> <table><tr><th></th><th>Klasse</th><th>Bezeichner</th><th>Typ</th><th>Initial</th><th>Kommentar</th></tr><tr><td>0</td><td>VAR_EXTERNAL</td><td>Taster_1</td><td>BOOL</td><td>FALSE</td><td>aktivieren/deaktivieren</td></tr><tr><td>1</td><td>VAR_EXTERNAL</td><td>Taster_2</td><td>BOOL</td><td>FALSE</td><td>aktivieren/deaktivieren</td></tr><tr><td>2</td><td>VAR_EXTERNAL</td><td>Taster_3</td><td>BOOL</td><td>FALSE</td><td>aktivieren/deaktivieren</td></tr></table> <p>Der Ausgang ist implizit in der Funktion enthalten und wird nicht im POE-Kopf deklariert.</p> <p>Programcode im POE-Rumpf der Funktion "Deak_fun"</p> 		Klasse	Bezeichner	Typ	Initial	Kommentar	0	VAR_EXTERNAL	Taster_1	BOOL	FALSE	aktivieren/deaktivieren	1	VAR_EXTERNAL	Taster_2	BOOL	FALSE	aktivieren/deaktivieren	2	VAR_EXTERNAL	Taster_3	BOOL	FALSE	aktivieren/deaktivieren
	Klasse	Bezeichner	Typ	Initial	Kommentar																				
0	VAR_EXTERNAL	Taster_1	BOOL	FALSE	aktivieren/deaktivieren																				
1	VAR_EXTERNAL	Taster_2	BOOL	FALSE	aktivieren/deaktivieren																				
2	VAR_EXTERNAL	Taster_3	BOOL	FALSE	aktivieren/deaktivieren																				

Variablentyp	Erklärung
VAR_OUTPUT	<p>Ausgangsvariable, die Sie nur in Funktionsbausteinen verwenden können. VAR_OUTPUT wird erst auf den Initialwert gesetzt, nachdem Sie die SPS vom PROG- in den RUN-Modus geschaltet haben, oder nach einem Stromausfall. Bitte beachten Sie folgendes:</p> <ul style="list-style-type: none"> - In dem Funktionsbaustein, in dem die Ausgangsvariable deklariert worden ist, kann sie gesetzt werden, andere POEs können die Ausgangsvariable nur lesen. - Im Gegensatz zu Funktionsbausteinen haben Funktionen immer nur einen Ausgang, der genauso heißt wie die Funktion selbst, z. B. Deak_fun. Der Datentyp kann bei der Neuanlage der Funktion oder über Objekt → Eigenschaften eingegeben werden.
VAR_OUTPUT_RETAIN	<p>wie VAR_OUTPUT, aber selbsthaltend, d. h., der Wert der Variablen bleibt auch bei einem Stromausfall erhalten. VAR_OUTPUT_RETAIN wird nur nach einem Kaltstart auf den Initialwert gesetzt, das heißt, wenn ein Programm auf die Steuerung geladen oder Schalter INITIAL im PROG-Modus aktiviert wurde.</p> <p>Wenn Sie unter Extras → Optionen → Compiler-Optionen → Code-Erzeugung das Kontrollfeld "Haltende Variablen im Anwender-Adressbereich nicht initialisieren" aktiviert haben, werden haltende Variablen, denen der Anwender Adressen zugewiesen hat, nicht erneut initialisiert.</p>
VAR_IN_OUT	<p>Bei Ein-/Ausgangsvariablen (VAR_IN_OUT) werden die Aktualparameter vor dem Sprung in das Funktionsprogramm auf die Formalparameter kopiert und nach dem Rücksprung wieder von den Formalparametern auf die Aktualparameter zurückkopiert. Im Funktionsprogramm können die Formalparameter gelesen und beschrieben werden.</p>

Deklariieren Sie lokale Variablen im POE-Kopf wie im folgenden beschrieben.



◆ Hinweis

- **Bezeichner dürfen nicht mit einer Zahl beginnen.**
- **FP Adressen (X0, Y1 etc.) sind feststehende Begriffe und dürfen nicht als Bezeichner für globale oder lokale Variablen verwendet werden.**



◆ Vorgehensweise

1. Im Projektnavigator auf POE-Namen doppelklicken
2. Im Feld "Klasse" auf den Pfeil klicken
3. Variablenklasse auswählen <Tab>
4. Bezeichner eingeben <Tab>

Wenn Sie eine FB-Instanz deklarieren, wählen Sie hier den gewünschten Funktionsbaustein aus.

5. Im Feld "Typ" anklicken



6. Unter "Datentypen" gewünschten Datentyp auswählen

7. Bibliothek auswählen

8. Unter "Typen" gewünschten Datentyp anklicken

Wenn Sie eine FB-Instanz deklarieren, wählen Sie hier den gewünschten Funktionsbaustein aus.

9. [OK] wählen <Tab>

Der für den gewählten Datentyp voreingestellte Initialwert wird automatisch angezeigt. Sie können diesen bei Bedarf überschreiben.

10. <Tab>

11. Kommentar eingeben <Eingabe>

12. <Umsch> + <Eingabe>

Eine neue Deklarationszeile wird nach der aktuellen Zeile erstellt, wenn dies die letzte Zeile im POE-Kopf ist.

	Klasse	Bezeichner	Typ	Initial
0	VAR	Schalter_AUS	BOOL	FALSE
1	VAR			

Tipps:

Im FBS- und KOP-Editor können Sie lokale Variablen auch direkt im POE-Rumpf deklarieren, wenn Sie unter **Extras** → **Optionen** → **Programm-Optionen** → **Editoren** → **KOP/FBS Editoren** die Option "Neue Bezeichner deklarieren" ausgewählt haben. In diesem Fall wird automatisch das Dialogfeld "Variablen-Auswahl (Modus Neue


Variable)" geöffnet, sobald Sie im POE-Rumpf den Namen einer noch nicht im POE-Kopf deklarierten Variablen eingeben und <Eingabe> drücken. Sie können die Variable dann sofort deklarieren.

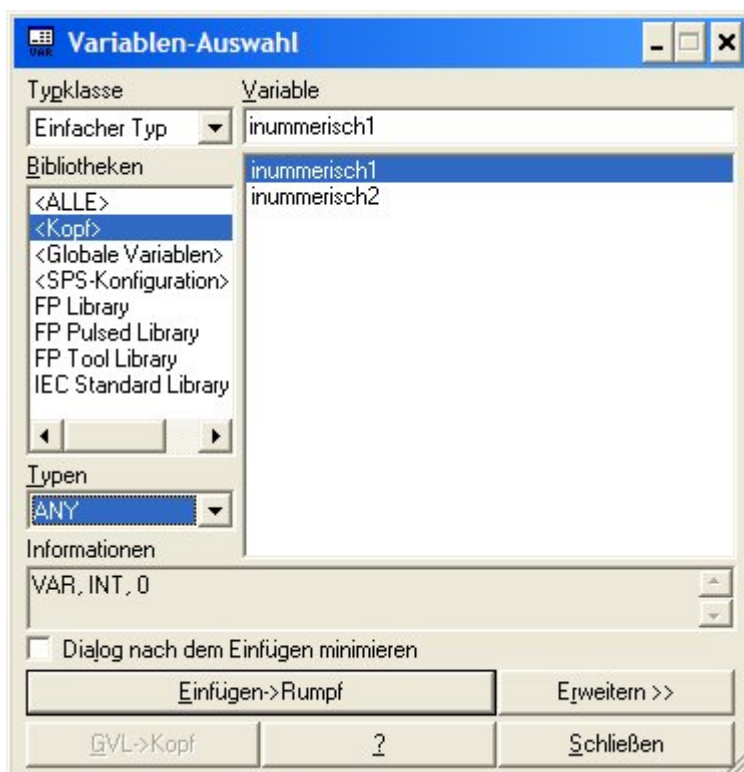
5.1.2.4 Variablen im Programm zuweisen

Nachdem Sie Variablen in der "Globalen Variablenliste" oder im POE-Kopf deklariert haben, können Sie sie im SPS-Programm mit Hilfe des Dialogfeldes "Variablen-Auswahl" zuweisen. Bei der folgenden Vorgehensweise wird davon ausgegangen, dass der POE-Rumpf bereits im Programmierfenster angezeigt wird.



♦ Vorgehensweise

1. **KOP und FBS: Auf das Namefeld klicken**
AWL: Cursor in die Operandenspalte stellen
2. Auf  klicken, <F2> drücken oder aus dem Kontextmenü "Variablen-Auswahl" wählen.



3. **Unter "Bibliotheken" Deklarationsort anklicken**

Die Variable kann im Kopf der aktuellen POE "<Kopf>", in der globalen Variablenliste "<Globale Variablen>" oder in einer Benutzerbibliothek deklariert sein. Wenn Sie "<Alle>" anklicken, werden alle diese Deklarationsorte durchsucht.

4. Typklasse auswählen

Sie können hier eine Klassifizierung für die Datentypen vornehmen, die dann im Auswahlfeld "Typen" angezeigt werden: Einfacher Typ (INT, WORD usw.), Strukturierter Datentyp oder Funktionsbaustein.

5. Datentyp auswählen

Im Auswahlfeld "Typen" legen Sie den Datentyp fest, für den die vorhandenen Variablen angezeigt werden. Wenn Sie z.B. unter "Bibliotheken" "<Globale Variablen>", als Typklasse "Einfacher Typ" und als Typ "Array" gewählt haben, werden unter "Variablen" alle Variablen vom Datentyp Array angezeigt, die in der Globalen Variablenliste deklariert sind.

6. Gewünschte Variable anklicken

Unter "Informationen" werden die Parameter der markierten Variable angezeigt.

7. Auf markierte Variable doppelklicken oder [Einfügen -> >Rumpf]

Die selektierte Variable wird an der aktuellen Position in den POE-Rumpf eingefügt. Wenn Sie eine globale Variable markiert haben, wird diese automatisch auch in den Kopf der aktuellen POE kopiert, falls sie dort noch nicht vorhanden ist.

Tipps:

Beim Einfügen der Variablen in das SPS-Programm können Sie auch globale Variablen verwenden, die noch nicht im POE-Kopf vorhanden sind.



♦ Vorgehensweise

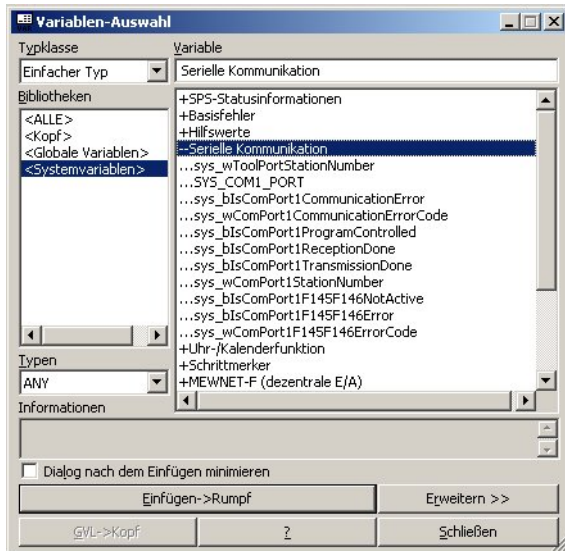
1. Im Dialogfeld "Variablen-Auswahl" unter "Bibliothek "Globale Variablen" auswählen
2. In der rechten Spalte die globale Variablen auswählen, die eingefügt werden soll
3. Auf [Einfügen -> Rumpf] klicken

Die Variable wird automatisch als VAR-EXTERNAL in die lokale Variablenliste eingetragen.

Mit [Erweitern >>] können Sie das Dialogfeld "Variablen-Auswahl" um den Bereich "Modus Neue Variable" erweitern, in dem Sie Variablen ändern oder neue deklarieren können.

5.1.3 Systemvariablen

Es empfiehlt sich, mit den SPS-unabhängigen Systemvariablen aus FPGWIN Pro auf die Sonderdatenregister und Sondermerker zuzugreifen. Sie können die Systemvariablen über den Dialog "Variablen-Auswahl" direkt in den Rumpf einfügen, ohne die Variablen im POE-Kopf zu deklarieren.



Detaillierte Informationen zur Verwendung der Systemvariablen finden Sie in der Online-Hilfe (Schlüsselwort "Datenübertragung von und zu Sonderdatenregistern").

Siehe auch:

Tabelle der Systemvariablen, ihrer Adressen und Verfügbarkeiten in der Online-Hilfe

5.2 Variablen ändern

Variablenparameter wie z. B. Name, Typ usw. ändern Sie für globale Variablen in der "Globalen Variablenliste" und für lokale Variablen in den jeweiligen POE-Köpfen. Sie können festlegen, dass die Änderungen automatisch in alle POE-Köpfe und -Rümpfe des aktuellen Projekts übernommen werden, in denen die Variablen verwendet werden. Oder übernehmen Sie die Änderungen mit **Extras** → **Variablenname aktualisieren**, wobei Sie die Übernahme für jeden POE-Kopf und jeden POE-Rumpf einzeln bestätigen oder verwerfen können.

Bestimmte Parameter der globalen und lokalen Variablen können Sie auch direkt im Dialogfeld "Variablen-Auswahl" vom POE-Rumpf aus ändern.

Weitere Informationen zu diesem Thema finden Sie in der Online-Hilfe unter dem Stichwort "Variablenname aktualisieren".

5.3 Variablen exportieren

Mit **Extras** → **CSV-Datei exportieren** erstellen Sie eine Textdatei im Format CSV (**C**omma **S**eparated **V**alues). Dabei werden die einzelnen Einträge durch Kommas ',' oder Strichpunkte ';' getrennt. Sie können das Trennzeichen unter **Extras** → **Optionen** → **Programmooptionen** → **CSV-Export** einstellen. Mit dieser Textdatei können später Informationen zu Variablen in externe Programme z.B. Prozeßvisualisierung importiert werden.



♦ Vorgehensweise

1. **Extras** → **CSV-Datei exportieren**
2. **Verzeichnis auswählen, in dem die CSV-Datei gespeichert werden soll**
3. **Dateinamen eingeben**
4. **Dateityp auswählen**
5. **[OK]**

Haben Sie **CSV Projekt** ausgewählt, wird eine Textdatei in der alle Variablen aus Control FPWin Pro mit Name und Adresse enthalten sind, erstellt. Zusätzlich werden allgemeine Informationen zum Projekt wie z.B. Projektname, Compilierzeit und Control FPWin Pro Version exportiert.

Jede Variable enthält folgende Einträge in der Textdatei: Klasse, Name, IEC-Adresse, FP-Adresse, Typ, Anfangswert, Kommentar.

Mit **CSV-Export (konfigurierbar)** wird ein Dialogfeld geöffnet, in dem Sie den Exportvorgang genauer festlegen können.

[Zuordnen]

Mit einem Klick auf Zuordnen wird der markierte Eintrag aus der linken Liste dem markiertem Eintrag aus der rechten Liste zugeordnet und innerhalb der Klammern angezeigt. In der rechten Liste sind auch mehrfache Markierungen möglich. Die Einträge der rechten Liste können in dem Textfeld unterhalb der Liste bearbeitet werden. Sie können es dort bearbeiten. So kann man die Einträge Feld x durch symbolische Bezeichner ersetzen.

[Zurücksetzen]

Setzt einen markierten Eintrag aus der Liste rechts zurück. In Klammern erscheint der Hinweis "nicht zugeordnet".

[Export-Konfiguration speichern]

Speichert die Konfiguration des Exports in einer eigenen Datei.

[Export-Konfiguration laden]

Speichert die Konfiguration des Exports in einer eigenen Datei.

Spaltenüberschrift exportieren

Ist dieses Kästchen angekreuzt, wird eine Überschrift in die CSV-Datei geschrieben. Diese Überschrift enthält, durch das Trennzeichen getrennt, die Einträge Feld1; Feld2, ... bzw. die symbolischen Bezeichner.

Array-Elemente als einzelne Variablen exportieren

Dieses Kontrollfeld aktivieren, um alle Elemente des Array in der Exportdatei aufzulisten.

[Exportieren]

Starten den Exportvorgang

[Abbrechen]

Schließt den Dialog, ohne daß der Exportvorgang gestartet wird.



◆ Hinweis

Um strukturierte Datentypen oder Arrays Systemen zugänglich zu machen, die diese nicht unterstützen, werden die einzelnen Elemente zusätzlich als Variablen in die CSV-Datei exportiert.

5.4 Variablen importieren

Verwenden Sie **Extras** → **CSV-Datei importieren**, um aus einer Textdatei Variablen in die globale Variablenliste zu importieren. Die Textdaten müssen im Format CSV (**C**omma **S**eparated **V**alues) sein. Dabei werden die einzelnen Einträge durch Kommas ‘,’ oder Strichpunkte ‘;’ getrennt. (Sie können das Trennzeichen unter **Extras** → **Optionen** → **Programmooptionen** → **CSV-Export** einstellen.)



♦ Vorgehensweise

1. **Extras** → **CSV-Datei importieren**
2. **Verzeichnis auswählen, in dem die CSV-Datei gespeichert ist**
3. **Dateinamen eingeben**
4. **Dateityp auswählen**
5. **[OK]**

Wählen Sie **CSV-Import (Projekt)** nur, wenn die CSV-Datei mit Control FPWIN Pro erstellt worden ist und der Dateityp **CSV-Export (Projekt)** gewählt wurde. Es werden alle globalen Variablen der CSV-Datei in die globale Variablenliste importiert.

Haben Sie **CSV allgemein** ausgewählt, können Sie CSV-Dateien importieren, welche nicht mit Control FPWIN Pro erstellt wurden. Ein Dialogfeld wird geöffnet, in dem Sie den Importvorgang genauer festlegen können. Erläuterungen zum Dialogfeld "CSV-Import konfigurieren":

"Importiere ab Zeile"

Ab dieser Zeile wird der Importvorgang gestartet. Die ausgewählte Zeile wird in der Liste darüber angezeigt.

[Zuordnen]

Mit einem Klick auf Zuordnen wird der markierte Eintrag aus der linken Liste dem markiertem Eintrag aus der rechten Liste zugeordnet und innerhalb der Klammern angezeigt. In der rechten Liste sind auch mehrfache Markierungen möglich.

[Zurücksetzen]

Setzt einen markierten Eintrag aus der Liste rechts zurück. In Klammern erscheint der Hinweis "nicht zugeordnet".

Trennzeichen

In dieser Gruppe wird das Trennzeichen ausgewählt, welches die einzelnen Spalten in der zu importierenden CSV-Datei trennt.

[Export-Konfiguration speichern]

Speichert die Konfiguration des Imports in einer eigenen Datei.

[Import-Konfiguration laden]

Speichert die Konfiguration des Imports in einer eigenen Datei.

[Import]

Startet den Importvorgang.

[Abbrechen]

Schließt den Dialog, ohne daß der Importvorgang gestartet wird.

5.5 Array und strukturierter Datentyp

ARRAYs

Ein Array ist eine Gruppe von Variablen, die alle den **gleichen** elementaren Datentypen haben und hintereinander in einem zusammenhängenden Block angeordnet sind. Diese Variablengruppe stellt selbst wieder eine Variable dar und wird daher auch deklariert. Im Programm können Sie dann das gesamte Array oder nur einzelne Array-Elemente verwenden.



◆ Hinweis

Ein Array kann nicht als Variable eines anderen Arrays verwendet werden.

Gültige Datentypen für Arrays sind:

- BOOL
- INT
- DINT
- REAL
- WORD
- DWORD
- TIME
- STRING

Arrays können 1-, 2- oder 3-dimensional sein. In jeder Dimension kann ein Array mehrere Felder haben.

Strukturierter Datentyp

Ein strukturierter Datentyp (SDT) besteht nur aus einer Gruppe von Variablen, die sich aus mehreren elementaren Datentypen (BOOL, WORD usw.) zusammensetzt. Solche Gruppen werden z. B. verwendet, wenn Tabellen verarbeitet werden wie für die Bitmuster Ausgabe beim Befehl F164_SPD0 (FP1, FP-M) der "FP-Library" (siehe Online-Hilfe). Die Bitmuster Ausgabe dieses Befehls können Sie z.B. zur Drehzahlregelung eines Motors über einen Drehzahlregler verwenden. Einen SDT definieren Sie zunächst im SDT-Pool. Sie können den SDT dann im Feld "Typ der globalen Variablenliste oder eines POE-Kopfs ähnlich verwenden wie die Datentypen Integer, BOOL etc. Im Programm können Sie dann entweder die gesamten SDT oder einzelne Variablen des SDT verwenden.



◆ Hinweis

Ein SDT kann nicht als Variable eines anderen SDT verwendet werden.

Weitere Informationen zu Arrays und SDTs finden Sie in der Online-Hilfe.

5.6 Querverweislisten

In FPWIN Pro können Sie sich mit Hilfe einer Querverweisliste eine gute Übersicht über alle im Projekt verwendeten Variablen, deren Parameter, Deklarationen, Abhängigkeiten und Referenzen verschaffen. Über ein Dialogfeld mit verschiedenen Such- und Filterkriterien können Sie die zu suchenden Querverweise genauer bestimmen und sich für jede Variable alle Deklarationen und POE-Rümpfe, in denen sie vorkommt, direkt anzeigen lassen.

Die Querverweisliste wird aus den Querverweisobjekten (*.SCT) --die zuerst für jede POE getrennt getestet werden--aus der Liste der globalen Variablen und aus jedem strukturierten Datentyp (SDT) erstellt. Diese Querverweisobjekte werden dann vom System verknüpft und in einer Querverweisdatei (*.SCX) zusammengefasst.



◆ Hinweis

- **Querverweisobjekte für die Querverweisliste werden nur erstellt, wenn Sie die Objekte (POE, GVL, SDT) oder das gesamte Projekt mit Objekt → Kontrolle geprüft und die Option zum Erzeugen von Querverweisobjekten bei Kontrolle/Kompilieren unter Extras → Optionen → Programmooptionen → Querverweis aktiviert haben. Bei jeder Kontrolle eines Objekts wird das zugehörige Querverweisobjekt aktualisiert.**
- **Wenn das Kontrollfeld "Querverweisliste aus Querverweisobjekten erzeugen" markiert ist, wird die Querverweisliste automatisch beim Kompilieren des Programms erstellt. Da sich dadurch jedoch der Kompiliervorgang verlängert, empfiehlt es sich, die Querverweisliste nach dem Kompilieren zu erstellen.**

Weitere Informationen zur Querverweisliste finden Sie in der Online-Hilfe unter dem Stichwort "Querverweisliste erzeugen".

5.7 Rezeptureditor

Wenn Sie mit Ihrem SPS-Programm z. B. einen Reaktorbehälter für chemische Substanzen mit Zu- und Ablaufventilen, Heizung und Rühranlage zum Durchmischen beim Dosiervorgang usw. steuern wollen, kann es sehr hilfreich sein, die Steuerungsvariablen dieser Komponenten für eine bestimmte Substanz zu einer Rezeptur zusammenzufassen.

Wählen Sie **Monitor → Rezeptureditor → Rezeptureditor öffnen**, um Datensätze zusammenzustellen, die alle erforderlich Variablen für eine bestimmte Rezeptur enthalten. Mit anderen Worten, Sie können die Variablen eintragen, ändern oder ihnen neue Werte zuweisen. Dann lässt sich die Rezeptur in einer Datei (*.rez) speichern und mit Monitor → Rezeptureditor → Rezeptur zur SPS übertragen im Online-Modus von Ihrem PC an die SPS übertragen.

Mit **Monitor → Rezeptureditor → Rezeptur von der SPS rückübertragen** können Sie die aktuellen Variablenwerte einer Rezeptur im Online-Modus aus der SPS in Ihren PC übertragen (upload).

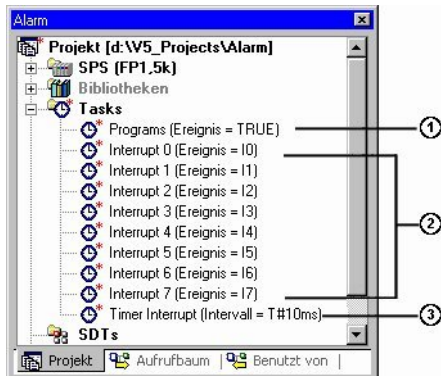
Weitere Informationen finden Sie in der Online-Hilfe unter dem Stichwort "Rezeptureditor öffnen".

Kapitel 6

Tasks

6.1 Tasks

Tasks stehen an der Spitze der Projekthierarchie. Sie steuern alle POEs vom Typ PRG (Programm). Die Tasks befinden sich im **Task-Pool** des Projektnavigators.



Geöffneter Task-Pool

FPGWIN Pro bietet drei Arten der Ausführungssteuerung an:

①	Zyklisch (Programs) Das Programm wird einmal vollständig abgearbeitet und anschließend wiederholt, Beispiel: Aus einem Tank wird z. B. kontinuierlich Flüssigkeit abgepumpt und die Temperatur der Flüssigkeit wird gemessen.
②	Ereignisgesteuert (Interrupt) Das Programm wird aufgerufen, wenn ein bestimmtes Ereignis eintritt. Beispiel: Wenn z. B. die Mindesttemperatur im Tank unterschritten wird, wird das Programm "Heizen" aufgerufen.
③	Zeitgesteuert (Timer Interrupt) Das Programm wird in regelmäßigen Zeitabständen aufgerufen. Beispiel: Alle 10 Sekunden wird z. B. das Programm "Synchronisieren" aufgerufen.



◆ Hinweis

Jede POE vom Typ PRG muss einer Task zugewiesen sein. Andernfalls wird sie nicht abgearbeitet.

Sie können einer Task eine oder mehrere POEs zuweisen. Wenn Sie mehrere POEs zuweisen, werden diese entsprechend der Reihenfolge ihrer Eingabe abgearbeitet. Während der Kompilierung werden die Programme als ein einzelnes Programm behandelt und auf die SPS heruntergeladen.

Sie können ein Programm innerhalb einer Task für die Codegenerierung deaktivieren bzw. wieder aktivieren, indem Sie das Programm markieren und im Pop-up-Menü oder im Menü Bearbeiten den Eintrag **Deaktivieren/Aktivieren** auswählen.

6.2 Programm einer Task nachträglich zuweisen

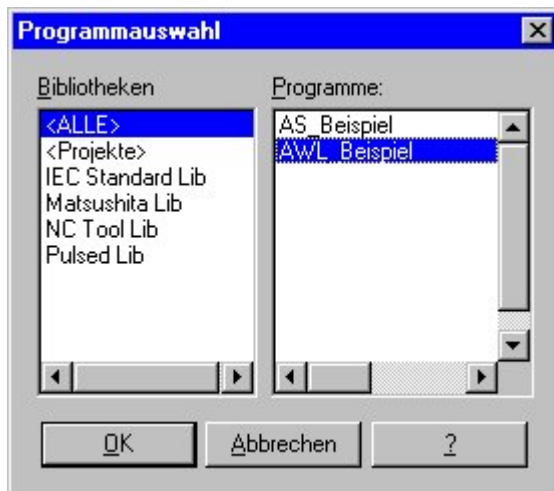
Sie können ein Programm einer Task direkt beim Anlegen eines neuen Programms zuweisen oder nachträglich über die Task-Liste.

In dieser Vorgehensweise soll das Programm zyklisch gesteuert werden und wird daher der Task "Programs" zugewiesen.

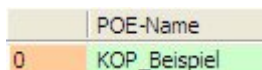


♦ Vorgehensweise

1. Im Projektnavigator auf "Tasks" doppelklicken
2. Unter "POE-Name" auf klicken
3. Unter "Programme" POE auswählen



4. [OK]



Wenn Sie mehrere Programm erstellt haben, können Sie diese einer Task zuordnen. Alle zugewiesenen Programme werden dann von dieser einzelnen Task gesteuert.

5. <Tab>
6. Kommentar eingeben <Eingabe>
7. Objekt → Speichern

8. Objekt → Schließen



◆ Hinweis

- Wenn ein Projekt mehrere AS-Programme enthält, müssen diese hintereinander in den Task-Pool eingetragen werden. Dazwischen darf sich keine andere POE befinden.

	POE-Name
0	KOP_Beispiel
1	AS_Beispiel1
2	AS_Beispiel2
3	AS_Beispiel3
4	FBS_Beispiel

- Sie können ein Programm innerhalb einer Task für die Codegenerierung deaktivieren bzw. wieder aktivieren, indem Sie das Programm markieren (oder die POE-Nummer auswählen) und im Pop-up-Menü oder im Menü Bearbeiten den Eintrag Deaktivieren/Aktivieren auswählen.

Mit **Objekt** → **Eigenschaften** oder <Alt>+<Eingabe>.

Task-Information

Ereignis

Dieses Feld zeigt das zugeordnete Ereignis an (oder TRUE bei zyklischer Abarbeitung). Das Ereignis ist fest mit der Task verbunden.

Intervall

Dieses Feld zeigt bei zeitgesteuerten Tasks (Timer Interrupt) das Intervall an, nach dem die POE wieder aufgerufen wird. Das Intervall können Sie ändern, indem Sie in das Intervall-Feld klicken und eine Zeit eingeben.

**◆ Hinweis**

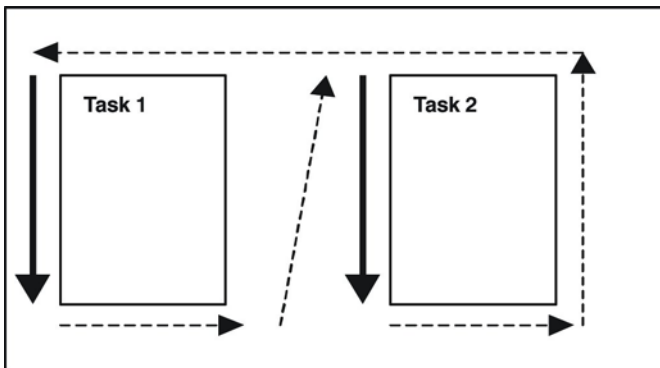
Das Intervall muss im IEC-Format eingegeben werden: Beispiel: T#10s.

Priorität

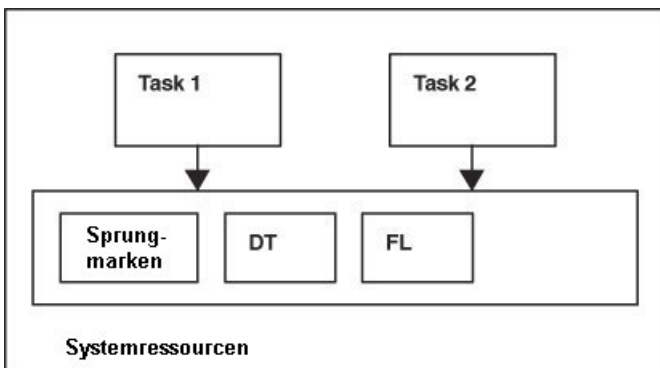
In Control FPWIN Pro ist die Priorität für alle Tasks fest auf den Wert 31 eingestellt und nicht veränderbar. Interrupt-Tasks werden in der Reihenfolge abgearbeitet, die im Task-Pool angegeben ist, das heißt, Interrupt 0 hat eine höhere Priorität als Interrupt 1 usw.

6.3 Zweite Task für FP10SH/120K und FP2SH 120k

Für die FP10SH 120K und die FP2SH 120k ist der Programmspeicher in zwei Tasks aufgeteilt. Jede Task kann maximal 60K Programmschritte umfassen. Wird eine POE in die erste Task eingetragen, fügt der Compiler den erzeugten Programmcode automatisch in die ersten 60K des Programmspeichers ein. Wird eine POE in die zweite Task eingetragen, fügt der Compiler den erzeugten Programmcode automatisch in die zweiten 60K des Programmspeichers ein. Für die beiden Programmspeicher sind zwei Programm-Code-Editoren im Projektnavigator vorhanden (Programm-Code und Programm-Code_2). Die Initialisierung von Variablen, die in Programmen (POE) verwendet werden, geschieht immer in der ersten Task. Die Ausführung der beiden Tasks wird von der SPS wechselweise vorgenommen (Task 1, Task 2, Task 1, Task 2, ...).



Die Systemressourcen, wie z.B. Labels, Datenregister oder Fileregister werden von beiden Tasks gemeinsam genutzt und zentral verwaltet.



◆ Hinweis

- Programme, die im Task-Pool unter Programs 1 und den Interruptroutinen (Interrupt 0 bis Interrupt 23 und Timer Interrupt) eingetragen sind, werden in der ersten Task abgelegt.
- Programme, die im Task-Pool unter Programs 2 eingetragen sind, werden in der zweiten Task abgelegt.
- Programme, die in Ablaufsprache erstellt sind können nur mit der ersten Task verwendet werden.

- **Wird eine Funktion in verschiedenen POEs verwendet die nicht gemeinsam einer Task zugeordnet sind, wird der Programmcode der Funktion dupliziert. Das heißt, der Programmcode der Funktion wird jeweils in der ersten und der zweiten Task abgelegt.**
- **Auch wenn unter Extras → Optionen → Compiler-Optionen → Code-Erzeugung das Kontrollfeld "Anwender-Funktionsbausteine indizieren" gewählt ist, wird der Programmcode des Funktionsbausteins dupliziert, falls der Funktionsbaustein von POEs in beiden Tasks verwendet wird.**

Kapitel 7

Netzwerke

7.1 Netzwerke Einführung

Netzwerke sind Programmabschnitte, die eine abgeschlossene Teilaufgabe beinhalten. Sie werden in allen Editoren gleich erzeugt. Netzwerke bestehen aus zwei Spalten. In der linken Spalte befindet sich das Netzwerk-Infofenster und in der rechten das Programmierfenster. Im Netzwerk-Infofenster befinden sich fortlaufende Netzwerknummern. Sie können hier eine Sprungmarke (auch Label genannt) eingeben.

Im Programmierfenster geben Sie das Programm ein. Die nachstehende Abbildung zeigt zwei Netzwerke mit den Sprungmarken "start" und "stop".

1	(*) Programmteil zum Starten des Motors *)
Start:	LD Schalter_Start
	DF (*) Flankenauswertung des
	ANDN Fehler
	S Motor_Start
2	(*) Programmteil zum Stoppen des Motors *)
Stop:	LD Fehler
	OR Schalter_Stop
	R Motor_Start

Sprungmarken im AWL Netzwerk

①	Netzwerk-Infofenster mit fortlaufender Netzwerknummer
②	Marke (z. B. start) und Titel (z.B. Motor_Start)
③	Programmierfenster

Die Gesamtheit aller Netzwerke ergibt eine POE. In einer POE können Sie Netzwerke einfügen, löschen, mit Marken versehen und kommentieren. Nachdem Sie Netzwerke erzeugt haben, können Sie sich diese Netzwerke **Werkzeuge** → **Netzwerkliste** (siehe Seite 71) auflisten lassen.



◆ Hinweis

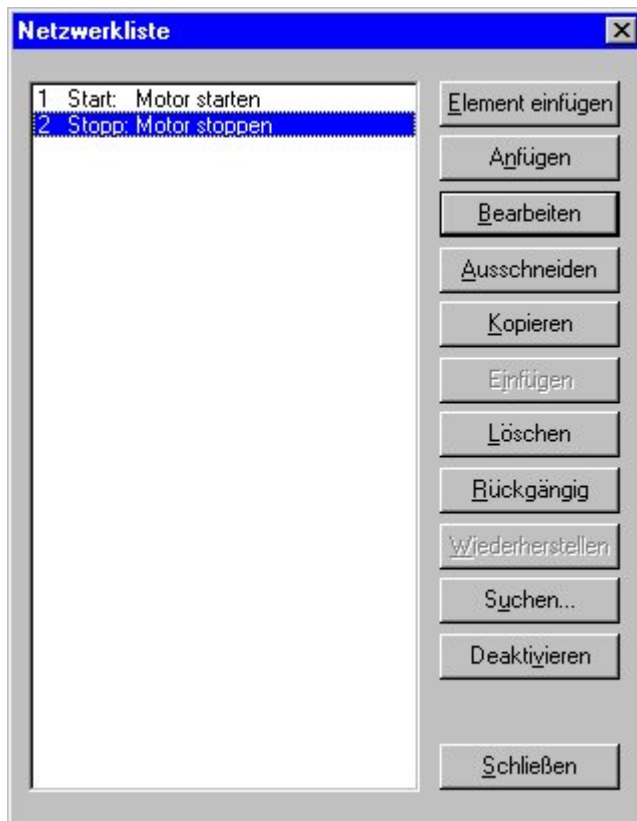
- In der Ablaufsprache dürfen in den Netzwerken von Aktionen keine Marken definiert werden!
- Vergeben Sie keine Marken, wenn Sie kein Sprungziel benötigen. Andernfalls wird überflüssiger Programmcode erzeugt.

7.2 Netzwerkliste



♦ Vorgehensweise

1. Im Navigator auf beliebigen Rumpf doppelklicken
2. Werkzeuge → Netzwerkliste



Im obigen Beispiel werden 2 Netzwerke mit den Sprungmarken "Start" und "Stopp" und ihren Titeln (Kommentaren) aufgelistet.

Im Dialogfeld Netzwerkliste befinden sich auf der rechten Seite mehrere Schaltflächen, mit denen Sie die Netzwerke bearbeiten können.

Markieren Sie das gewünschte Netzwerk, bevor Sie es bearbeiten. Sie können einzelne, mehrere aufeinander folgende oder beliebige nicht aufeinander folgende Netzwerke markieren.

7.3 Marken und Titel definieren

Sie können eine Marke für ein Netzwerk (siehe Seite 70), auch Sprungmarke genannt, definieren, um z.B. von einer Programmstelle aus in ein bestimmtes Netzwerk zu springen.



◆ Hinweis

- **Vergeben Sie keine Marken, wenn Sie kein Sprungziel benötigen. Andernfalls wird überflüssiger Programmcode erzeugt.**
- **In der Ablaufsprache dürfen in den Aktionen keine Marken definiert werden!**
- **Marken müssen immer mit einem Doppelpunkt enden, da sonst der Compiler einen Programmierfehler meldet.**

Wenn Sie eine Marke definieren, können Sie gleichzeitig einen Namen (Titel) für das Netzwerk eingeben. Der Titel eines Netzwerks wird nur in der Netzwerkliste angezeigt. Die folgende Vorgehensweise gilt auch für das Ändern einer Marke oder eines Netzwerknamens.

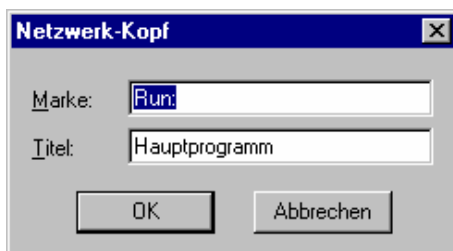


◆ Vorgehensweise

In der Netzwerkliste:

1. **Netzwerk auswählen**
2. **[Bearbeiten]**

Es erscheint folgendes Dialogfeld:



3. **Marke und/oder Titel eingeben**

Wenn keine Marken erforderlich sind, genügt es eventuell, den Titel einzugeben.

4. **[OK]**

Im Netzwerk-Infofenster:

1. **Auf das Netzwerk-Infofenster doppelklicken**
Das Dialogfeld "Netzwerk-Kopf" erscheint.
2. **Marke und/oder Titel eingeben**
3. **[OK]**

7.4 Netzwerk deaktivieren/aktivieren

Mit Deaktivieren/Aktivieren wird das selektierte Netzwerk für die Codegenerierung deaktiviert bzw. wieder aktiviert. Der Zustand des Netzwerkes wird im Netzwerk-Infofenster bzw. in der Netzwerkliste durch ein X gekennzeichnet.

In der Netzwerkliste:

1. **Netzwerk(e) auswählen**
2. **[Deaktivieren]**

Im Programmierfenster:

1. **Netzwerk(e) auswählen**
2. **Bearbeiten→ Netzwerk deaktivieren/aktivieren oder im Kontextmenü "Netzwerk deaktivieren/aktivieren" wählen**

Führen Sie den Mauszeiger in das Netzwerk-Infofenster, um das Kontextmenü zu öffnen.

Kapitel 8

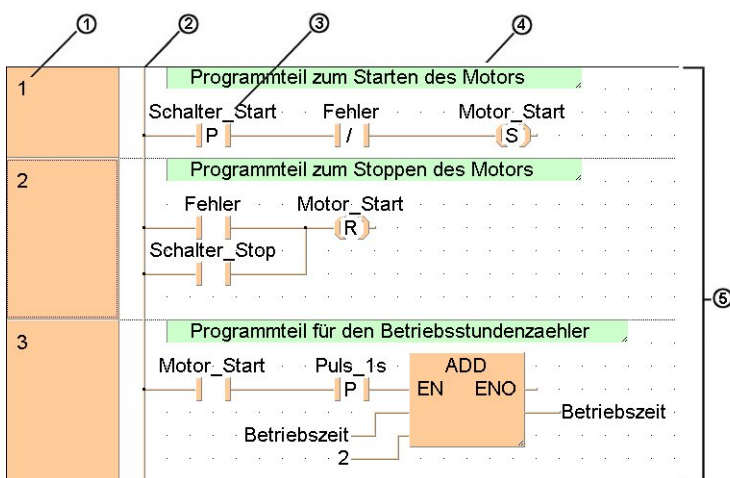
Programmiereditoren

8.1 KOP- und FBS-Editor

Im Kontaktplan- und im Funktionsbaustein-Editor werden Programme grafisch dargestellt. Beim Kontaktplan-Editor werden Kontakte, Spulen, Eingangsvariablen, Ausgangsvariablen, Sprünge und Rücksprünge in Form von Kontaktplansymbolen angezeigt.

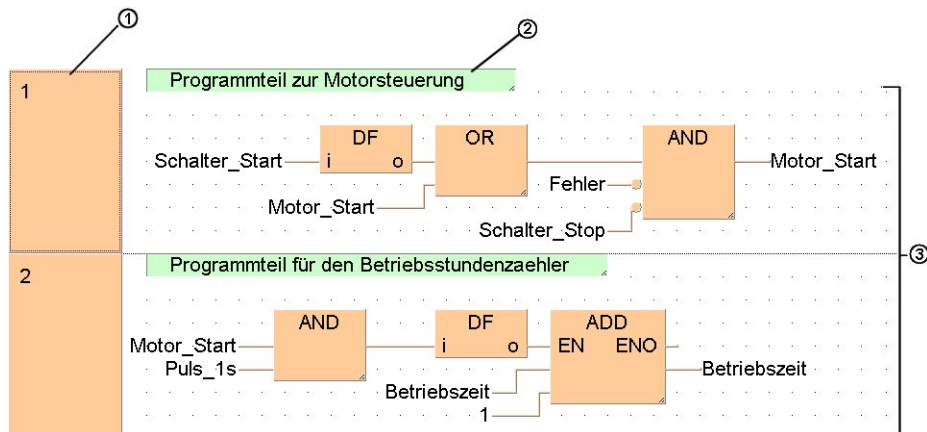
- Der Kontaktplan-Editor bietet außerdem
- die Möglichkeit, boolesche Variablen in Form von Kontakten und Spulen einzubinden

eine Stromschiene, die die Netzwerke untereinander verbindet und an die alle Kontakte, Spulen usw. innerhalb eines Netzwerks angeschlossen sind. Ein FBS-Programm besteht aus Bausteinen, Sprüngen, Eingangsvariablen und Ausgangsvariablen. Die Abbildung zeigt das Programmierfenster mit drei Netzwerken, die im Kontaktplan erstellt wurden:




KOP-Programm mit Stromschiene und booleschen Kontakten

①	Netzwerknummer im Netzwerk-Infofenster
②	Stromschiene
③	Netzwerk mit booleschen Kontakten
④	Kommentarfeld
⑤	Programmierfenster



FBS-Programm

①	Netzwerknummer im Netzwerk-Infofenster
②	Kommentarfeld
③	Programmierfenster

Jeder POE-Rumpf besteht aus einem oder mehreren Netzwerken. Im Netzwerk-Infofenster werden die Netzwerknummern, Sprungmarken und Haltepunkte angezeigt. Hier finden Sie die Netzwerknummer und ggf. Sprungmarken oder Stati, z.B. bei Haltepunkten (Debug), Netzwerkauswahl und Fehlermeldungen. Das Programm wird rechts im Programmierfenster angezeigt. Kommentare können Sie mit Hilfe der Schaltfläche  einfügen.




◆ Hinweis

Innerhalb eines Netzwerks dürfen Sie bis zu 160 Elemente platzieren. Unter einem Element versteht man alles, was Sie am Bildschirm erzeugen, auch Linien. Achten Sie daher beim Erzeugen von Linien darauf, sie in einem Stück zu zeichnen. Wenn Sie eine Linie aus mehreren Stücken zusammensetzen, zählt jedes Stück als Element.

8.1.1 Objekte verbinden

In KOP und FBS gibt es zwei Editiermodi: Platzierungsmodus (Cursor = Pfeil) Verbindungsmodus (Cursor = Zeichenstift). Im Platzierungsmodus markieren Sie die Programmiersymbole, wie z. B. Kontakte, und positionieren bzw. bearbeiten sie im Programmierfenster. Im Verbindungsmodus zeichnen Sie die Linien, die die Programmiersymbole miteinander verbinden.

Der Platzierungsmodus ist voreingestellt. Sie wechseln zwischen den beiden Modi mit

- **Bearbeiten** → **Linie zeichnen** (der Befehl ist im Verbindungsmodus mit einem Häkchen versehen) oder
- klicken in der Symbolleiste auf , oder

- wählen den Befehl "Linie zeichnen" aus dem Kontextmenü.

Mit <ESC> oder einem Doppelklick auf eine freie Fläche im Programmierfenster verlassen Sie den Verbindungsmodus.

Programmiersymbole, die keine Verbindungspunkte besitzen, können im Verbindungsmodus nicht ausgewählt werden. Logische Elemente, Eingangs- und Ausgangsvariablen, Sprünge und Rücksprünge können auch im Verbindungsmodus platziert werden.

KOP- und FBS-Editor bieten Ihnen folgende Hilfen, die Ihnen das Programmieren in Kontaktplan und Funktionsbausteinsprache erleichtern:

- Wenn Sie Funktionen, Funktionsbausteine und Operatoren in den POE-Rumpf einfügen, werden diese Programmiersymbole automatisch mit leeren E-/A-Variablen und Namefelder ("?",) versehen, wenn das Kontrollfeld "Ein-/Ausgangsvariablen automatisch hinzufügen" unter **Extras** → **Optionen** → **Programm-Optionen** → **Editoren** → **KOP/FBS-Editor** aktiviert ist.
- Halten Sie <Umsch> gedrückt, während Sie Elemente verbinden, so wird die Verbindungslinie automatisch berechnet und wandert beim Verschieben der Kontakte mit.
- Halten Sie <Umsch> gedrückt, während Sie die Elemente verschieben, die bereits verbunden sind, z.B. eine Spule; die Verbindungslinien werden automatisch zusammen mit dem Element verschoben.
- Beim Kopieren von Programmiersymbolen (Kontakten, Funktionen usw.) werden mit **Bearbeiten** → **Kopieren/Einfügen** die zugehörigen Verbindungslinien mit kopiert. Wenn Sie ein Programmiersymbol kopieren, indem Sie es selektieren und <Strg> gedrückt halten, während Sie es an die neue Position ziehen, wird das Programmiersymbol mit allen zugehörigen Verbindungslinien dupliziert und automatisch mit Kontakten versehen.

8.1.1.1 Nützliche Tipps

Folgende Tipps erleichtern Ihnen die Arbeit im KOP-Editor:



◆ TASTATURBEFEHLE















- Mit <Strg> + <a> markieren Sie alle Netzwerke im POE-Rumpf.
- Mit <Tab> springen Sie innerhalb eines Netzwerkes von einem Namensfeld (Variablenname, FB-Instanzname) zum nächsten. Mit <Umsch> + <Tab> springen Sie rückwärts.
- Mit **Werkzeuge** → **Netzwerkhöhe optimieren** werden die markierten Netzwerke auf die minimale Höhe verkleinert.
- Mit einem Doppelklick auf eine beliebige freie Stelle im Netzwerk wechseln Sie zwischen Auswahl- und Verbindungsmodus.
- Mit <ESC> verlassen Sie den Verbindungsmodus.
- **Schnelle Tastaturbefehle:** Setzen Sie den Mauszeiger an der gewünschten Position auf das Raster. Drücken Sie jetzt eine Zahlentaste (1 bis 9), um das gewünschte Element zu erhalten.

Tipp: Wenn ein schneller Tastaturbefehl vorhanden ist, wird die zugehörige Zahl direkt auf dem Symbol angezeigt (links unten in kleiner blauer Schrift).

Eine vollständige Liste aller verfügbaren Tastaturbefehle finden Sie in der Online-Hilfe (Schlüsselwort "Tastaturbefehle").

8.1.1.2 Schaltflächen in der Symbolleiste

Wenn Sie ein KOP- oder FBS-Programm öffnen, erscheinen in der Symbolleiste folgende Schaltflächen:

Schaltfläche	Beschreibung
	Neues Netzwerk vor dem markierten Netzwerk einfügen
	Neues Netzwerk nach dem markierten Netzwerk einfügen
	Markiertes Netzwerk deaktivieren. Deaktivierte Netzwerke werden wie Kommentare behandelt, also nicht kompiliert.
	Zwischen Verbindungsmodus und Platzierungsmodus umschalten
	Variablen-Auswahl-Dialogfeld aufrufen, wenn Sie ein Namensfeld für eine Ein-/Ausgangsvariable ausgewählt haben
	Dialogfeld "OP/FUN/FB-Auswahl" öffnen, wenn Sie einen Funktionsbaustein/Operator oder eine Funktion auswählen und im Programmierfenster einfügen möchten.
	Namensfeld für eine Variable im Programmierfenster einfügen. Linker Mausklick an gewünschter Position.
	Sprunganweisung einfügen
	Rücksprung aus FUN/FB einfügen
	Kommentar einfügen
	Vertikalen Abstand ändern
	Horizontalen Abstand ändern
In der Symbolleiste des Kontaktplan Editors finden Sie zusätzlich folgende Symbole:	
	Kontakt mit einem Mausklick an der gewünschten Position im Programmierfenster einfügen Linker Mausklick an gewünschter Position.
	Spule in das Programmierfenster einfügen. Linker Mausklick an gewünschter Position.

8.1.1.3 Kontextmenü

Wenn Sie im Programmierfenster mit der rechten Maustaste klicken, wird ein Kontextmenü geöffnet. Dieses Menü enthält viele häufig gebrauchte Befehle.

Der untere Teil des Kontextmenüs enthält eine Liste der 5 zuletzt verwendeten Operatoren, Funktionen oder Funktionsbausteine.



◆ Hinweis

Für alle im POE-Rumpf eingefügten Funktionen, Funktionsbausteine oder Operatoren können Sie sich direkt die entsprechende Hilfeseite anzeigen lassen, indem Sie das gewünschte Programmiersymbol selektieren und <F1> drücken.

8.1.1.4 Programmierbeispiel

Programmierbeispiele finden Sie in der Online-Hilfe.

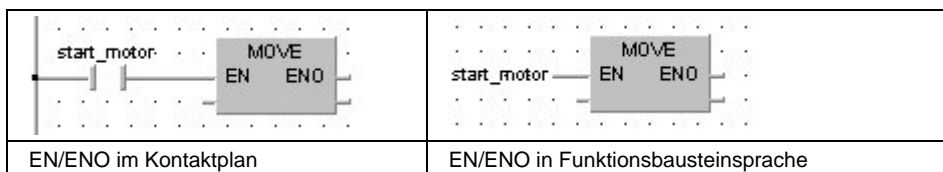
8.1.1.5 Enable Input und Enable Output

In FBS und KOP können Sie mit den EN/ENO-Funktionen und Funktionsbausteinen Bedingungen programmieren. EN (enable input) steht für Freigabe-Eingang, ENO (enable output) für Freigabe-Ausgang.


Alle IEC-Funktionen und -Funktionsbausteine gibt es sowohl mit als auch ohne EN/ENO.

Wenn Sie einen Freigabe-Eingang (EN) und einen Freigabe-Ausgang (ENO) benötigen, können Sie diesen Befehl für die Editoren KOP, FBS und AWL im Dialog OP/FUN/FB-Auswahl mit der Schaltfläche [Einfügen mit EN/ENO] auswählen. Die Funktion mit Enable (E_) erscheint dann in der Liste "Zuletzt verwendet" und im Kontextmenü und kann somit ohne erneute Auswahl wieder verwendet werden.

EN und ENO sind boolesche Variablen. Eine Eingangsvariable vom Typ BOOL oder das Ergebnis eines logischen Ausdrucks vom Typ BOOL können am EN-Eingang angeschlossen werden. Wenn EN auf TRUE gesetzt ist, wird die Funktion bzw. der Funktionsbaustein abgearbeitet. Nachdem die Funktion/der Funktionsbaustein erfolgreich abgearbeitet worden ist, wird der dazugehörige Ausgang ENO auf TRUE gesetzt. An den Ausgang ENO einer Funktion/eines Funktionsbausteins können Sie den EN-Eingang der nächsten POE anschließen, die nur dann abgearbeitet wird, wenn der Ausgang ENO der ersten POE auf TRUE gesetzt ist.



Anwenderfunktionen (siehe Seite 127) und Funktionsbausteine (siehe Seite 128) lassen sich mit oder ohne EN/ENO-Eingang bzw. Ausgang erstellen. Sie haben folgende Möglichkeiten:

- beim Erstellen einer POE (**Bearbeiten** → **Neu** → **POE** oder ) , wenn Sie die Option "Mit EN/ENO Kontakten" gewählt haben
- nachträglich mit **Objekt** → **Eigenschaften...**

Setzen des ENO-Ausgangs explizit im Rumpf während des Programmierens:

- Wird der ENO-Ausgang im Rumpf der benutzerdefinierten Funktion oder des Funktionsbausteins nicht explizit verwendet, hat er den gleichen Wert wie der EN-Eingang.
- Wird der ENO-Ausgang im Rumpf der benutzerdefinierten Funktion oder des Funktionsbausteins auf FALSE gesetzt, dann werden die Werte der Ausgangsvariablen nicht an die entsprechenden Ausgänge weitergeben.



◆ Hinweis

Setzen Sie EN auf TRUE, bevor die POE abgearbeitet wird. Nachdem die POE erfolgreich abgearbeitet worden ist, wird der dazugehörige ENO auf TRUE gesetzt. Wenn ENO nicht gesetzt wird, ist wahrscheinlich ein Fehler aufgetreten.


8.1.1.6 Programmiersymbole bearbeiten

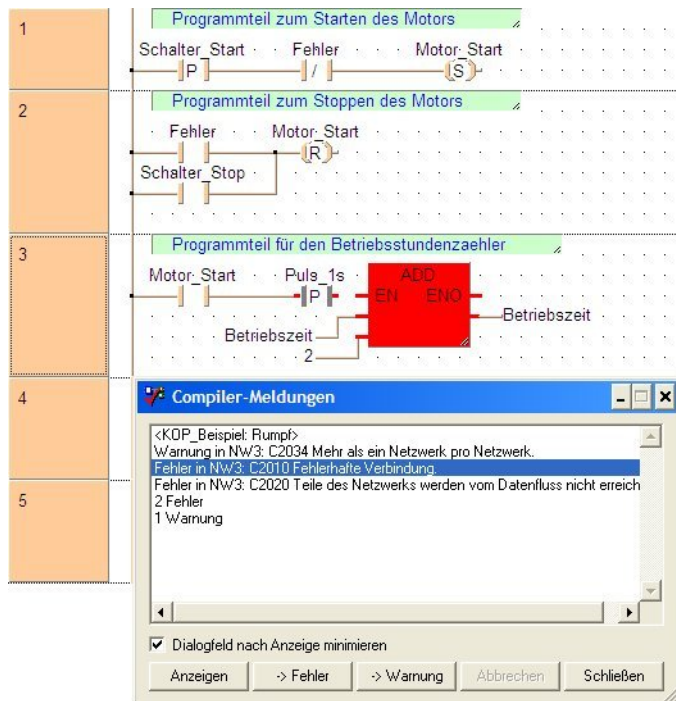
Im Kontaktplan und in der Funktionsbausteinsprache können Sie die Programmiersymbole in Ihrem Programm im Auswahlmodus wie folgt bearbeiten:

- selektieren und deselektieren
- ausschneiden/kopieren und einfügen
- verschieben (mit und ohne Verbindungslinien)
- Abstand zwischen Programmiersymbolen ändern
- Variablennamen eingeben/ändern
- Funktionen und Operatoren erweitern
- Kontakte negieren, Flanken, Set und Reset definieren
- Grafische Makros definieren

Eine detaillierte Beschreibung zu den Programmiersymbolen und deren Verwendung finden Sie in der Hilfe zu FPWIN Pro.

8.1.1.7 KOP oder FBS-Programm überprüfen

Sie können Ihr Programm jederzeit mit **Objekt → Kontrolle** oder  überprüfen. Das gesamte bis dahin eingegebene Programm wird dann auf Syntaxfehler und Deklarationsfehler (z. B. Verwendung nicht deklarierter Variablen) überprüft. Jeder Fehler wird einzeln in der Fehlerliste angezeigt. Wenn Sie auf einen Fehler in der Fehlerliste doppelklicken, wird er im POE-Rumpf in der eingestellten Fehlerfarbe markiert.



Oder wählen Sie den Fehler aus und klicken Sie auf [Anzeigen]. Das Programm, das den Fehler enthält wird auf dem Bildschirm dargestellt und der Fehler wird markiert.

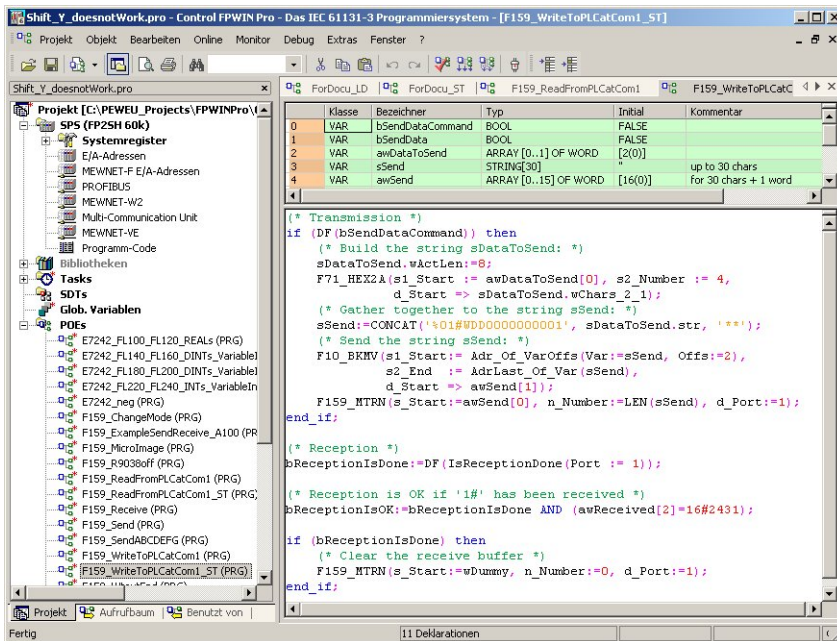


◆ Hinweis

Beheben Sie immer zuerst den ersten Fehler und wiederholen Sie anschließend Objekt → Kontrolle. Alle anderen Fehler sind unter Umständen Folgefehler. Deaktivieren Sie ein Netzwerk, wenn Sie es von der Kontrolle ausschließen möchten.

8.2 Structured Text Editor (ST)

Der ST-Editor ist ein textbasierender, syntaxfreier Editor, mit dem komplexe Ausdrücke und Kontrollstrukturen in Hochsprache programmiert werden können. Mit ST können Sie komplexe Programme schreiben und Strukturen mit einer optimierten Programmiersprache steuern. Er ist für alle Steuerungen verfügbar und belegt dabei nicht mehr Ressourcen (d.h. Schritte, Labels oder Calls) als andere Editoren bei vergleichbarer Programmierung.



8.2.1 Ausdrücke

Ausdrücke sind Verknüpfungen von Operanden (siehe Seite 85) durch Operatoren (siehe Seite 85) gemäß ihrer Rangfolge.

Bei gleicher Rangfolge erfolgt die Auswertung von links nach rechts:

Mit den Werten A:=1.0; B:=2.0; C:=3.0; und D:=4.0; erhält man für

$X := A + B - C * \text{SQRT}(D);$ das Ergebnis -3.

Durch Klammersetzen kann die Reihenfolge der Auswertung geändert werden und so erhält man für

$X := A + (B - C) * \text{SQRT}(D);$ das Ergebnis -1.

Boolesche Ausdrücke werden immer komplett ausgewertet:

```

IF a<100 AND UserFun1(a) THEN
  a:=a+1;
END_IF;
  
```

```
IF a<100 AND UserFun1(a) THEN
```

Hier wird UserFun1 auch ausgewertet wenn a>=100 ist.

Wenn das Abarbeiten von UserFun1 unerwünscht ist, weil es sehr zeitaufwendig oder weil für a>=100 ein Operationsfehler auftreten kann oder weil Speicherbereiche überschrieben werden können, kann man z.B. schreiben:

```
IF a<100 THEN
    IF UserFun1(a) THEN
        a:=a+1;
    END_IF;
END_IF;
```

Ausdrücke können auch Elemente eines Arrays bezeichnen:

```
X:=Array1[i+2];
```

8.2.2 Operanden

Mögliche Operanden innerhalb des ST-Editors sind:

Bezeichnung	Typ	Beispiel
Literal	Numerisch Zeichenkette Zeit	49 oder 3,14159 'Das ist ein Text' T#8d_3h_23m
Variable	Einzelvariable Element eines Arrays Element eines SDTs (siehe Seite 22) Element eines Arrays eines SDTs	Var1 Array1[5] Dut1.Var1 Dut1.Array1[i+5]
Funktion	Funktionsaufruf	Fun1(a,b,c)

Diese Operanden können untereinander mit Operatoren (siehe Seite 85) verknüpft werden. Die Kombination von Operatoren in Verbindung mit Operanden werden als Ausdrücke (siehe Seite 84) bezeichnet.

8.2.3 Operatoren

Mögliche Operatoren innerhalb des ST-Editors sind:

Operator	Beschreibung	Rangfolge
()	Klammern, Funktionsaufruf	am höchsten
-	Negation	
NOT	Komplement	

Operator	Beschreibung	Rangfolge
**	Potenzierung	am niedrigsten
*	Multiplikation	
/	Division	
MOD	Modulo	
+	Addition	
-	Subtraktion	
>,<,>=,<=	Vergleich	
=	Gleichheit	
<>	Ungleichheit	
&, AND	Boolesches UND	am niedrigsten
XOR	Boolesches Exklusiv ODER	
OR	Boolesches ODER	

Operatoren können Operanden (siehe Seite 85) miteinander verknüpfen. Die Kombinationen von Operatoren in Verbindung mit Operanden werden als Ausdrücke (siehe Seite 84) bezeichnet.

8.2.4 Anweisungen

Anweisungen des ST-Editors sind:

- die Zuweisungsanweisung :=
- die Auswahlanweisungen IF, CASE
- die Wiederholungsanweisungen FOR, WHILE, REPEAT mit der Abbruchanweisung EXIT
- die Rücksprunganweisung RETURN ST-Anweisungen müssen durch Strichpunkte getrennt werden.

Schlüssel wort	Beschreibung	Beispiel	Erklärung
:=	Zuweisung	a := 87; b := b + 1; c := SIN (x) ;	der rechts stehende Wert wird dem links stehenden Bezeichner zugewiesen

Schlüssel wort	Beschreibung	Beispiel	Erklärung
	Aufruf von Funktionen	<pre>Y := SIN(x); Y := LIMIT(MN := 0, IN := X, MX := 100);</pre>	<p>Funktionsargument in kurzer Schreibweise</p> <p>Funktionsargumente mit Formalparametern</p> <p>Hinweis:</p> <p>Bei Argumentübergabe mit Formalparametern ist die Reihenfolge beliebig.</p> <p>Bei Anwenderfunktionen können auch die EN-Ein- bzw. Ausgänge weggelassen werden.</p> <p>Ein- bzw. Ausgänge, die weggelassen werden, werden auf TRUE gesetzt.</p>
		<pre>Y := LIMIT(0, X, 100);</pre>	<p>Funktionsargumente ohne Formalparametern</p> <p>Hinweis:</p> <p>Bei Argumentübergabe ohne Formalparametern ist die Reihenfolge der Parameter zu beachten.</p> <p>Die Groß- und Kleinschreibung muss nicht berücksichtigt werden.</p>
	Aufruf von FBs	<pre>TON1(IN := Start1, PT := T#300ms, Q => End1, EV => EV_1);</pre>	<p>Funktionsbaustein-Argumente mit Formalparametern</p> <p>Hinweis:</p> <p>Bei Argumentübergabe mit Formalparametern ist die Reihenfolge beliebig.</p>
		<pre>Ton1(IN := Start1, PT := T#300ms); End1 := Ton1.Q; Ev1 := Ton1.EV;</pre>	<p>Funktionsargumente ohne Formalparametern</p> <p>Hinweis:</p> <p>Bei Argumentübergabe ohne Formalparametern ist die Reihenfolge der Parameter zu beachten.</p> <p>Die Groß- und Kleinschreibung muss nicht berücksichtigt werden.</p>

Schlüssel wort	Beschreibung	Beispiel	Erklärung
		<pre> Ton1.IN:=Start1; Ton1.PT:=T#300ms; Ton1(); IF Ton1.Q THEN END_IF; </pre>	Beliebige Verwendung der Formalparameter im aufrufenden Programm
IF	Bedingte Verzweigung	<pre> IF a>=0 AND a<=10 THEN b:=0; ELSIF a>=100 THEN b:=1; ELSE b:=2; END_IF; </pre>	Verzweigung abhängig vom booleschen Wert eines Ausdrucks
CASE	Mehrfachauswahl	<pre> CASE a OF 0: b:=0; 1,2: b:=1; 3,4,10...20: b:=2; 100...110: b:=3; ELSE b:=4; END_CASE; </pre>	Mehrfachauswahl in Abhängigkeit von einer Variablen
FOR	Schleifenanweisung	<pre> FOR i:=0 TO 100 DO SUM:=SUM + a[i] END_FOR; FOR i:=0 TO 100 BY 10 DO IF a[i]>=100 THEN EXIT; END_IF; END_FOR; </pre>	<p>Festgelegte Anzahl von Schleifendurchläufen mit voreingestellter Schrittweite 1</p> <p>oder mit benutzerdefinierter Schrittweite</p> <p>Hinweis: Der Wert der Steuervariablen (hier i) kann nach Ende der Schleifendurchläufe beliebige Werte annehmen und sollte deswegen nicht abgefragt werden.</p>

Schlüsselwort	Beschreibung	Beispiel	Erklärung
WHILE	Schleifenanweisung	<pre> i:=0; WHILE i<=100 AND a[i]<100 DO i:=i+10; END_WHILE; </pre>	Schleifendurchläufe mit Prüfung der Schleifenbedingung vor der Schleife
REPEAT	Schleifenanweisung	<pre> i:=0; REPEAT i:=i+10; UNTIL i>100 OR a[i]>=100 END_REPEAT; </pre>	Schleifendurchläufe mit Prüfung der Schleifenbedingung nach der Schleife
EXIT	Abbruchanweisung	<pre>EXIT;</pre>	Unbedingter Schleifenabbruch
RETURN	Rücksprung	<pre>RETURN;</pre>	Das Programm kehrt in die aufrufende POE zurück

8.2.5 Kommentare

Sie können Kommentare an jeder beliebigen Position innerhalb des ST-Editors eingeben. Kommentare werden durch Klammern mit Sternchen '(' und ') begrenzt. Sie können sich über mehrere Zeilen erstrecken.

```
(* dies ist ein
    mehrzeiliger Kommentar *)
```

Ungültiger Kommentar, da verschachtelt:

```
(* dies ist ein ungültiger Kommentar
```

Verschachtelte Kommentare der Form (* Stufe 1 (* Stufe 2 *) *) sind möglich, in der IEC 61131-3 allerdings nicht vorgesehen und generieren in Abhängigkeit der Compiler-Optionen eine Fehlermeldung zur Compile-Zeit, siehe **Extras** → **Optionen** → **Compiler-Optionen** → **Zusätzliche Fehler**.

8.2.6 Eingabehilfen

Die Vorgehensweise entspricht der Vorgehensweise im KOP- und FBS-Editor. Siehe daher KOP/FBS-Programm überprüfen (siehe Seite 83).

8.2.7 Eingabeschablonen für ST-Editor

Die folgenden Eingabeschablonen, die mit **<Strg> + <F1>** verfügbar sind, ersparen beim Programmieren Zeit. Die Tastenkombination **<Strg> + <F1>** ist eine Alternative zur Eingabe eines Begrenzers.

Eingabeschablonen werden automatisch erzeugt nach:

- Einfügen einer Funktion (siehe Seite 36) aus dem Dialogfeld OP/FUN/FB-Auswahl
- Einfügen eines Instanznamens für einen Funktionsbaustein aus dem Dialogfeld Variablen-Auswahl

Hinweis: Nach dem Einfügen eines Funktionsbausteins wird das Auswahldialogfeld für Variablen automatisch angezeigt, und Sie werden aufgefordert, einen Namen für den Funktionsbaustein einzugeben.

- Einfügen einer Funktion oder deren Kurzform in das Programmierfenster.



♦ Vorgehensweise

1. **Geben Sie dazu zunächst den Befehl (z.B. F0) oder dessen Kurzform (z.B. MV) in das Programmierfenster ein.**
2. **Drücken Sie dann <Strg> + <F1> oder geben Sie einen Begrenzer ein**
Bei der Funktion MV erscheint z. B.:
`F0_MV(?s_ANY16?, ?d_ANY16?);`
3. **Markieren Sie die zwischen den Fragezeichen stehenden Datentypen (z. B. durch Doppelklick)**
4. **Operanden eingeben**

Mit <F1> erhalten Sie Hilfe zu der Funktion oder dem Funktionsbaustein, auf dem sich der Cursor befindet.



♦ TASTATURBEFEHLE

Funktionen und ihre Kurzformen

Eingabe im ST-Editor	Ergebnis nach <Strg> + <F1> oder Eingabe eines Begrenzers
f0	<code>F0_MV(?s_ANY16?, ?d_ANY16?);</code>
f0_	
_mv	

mv	
f0_mv	
p0	P0_MV(?s_ANY16?, ?d_ANY16?);
p0_	
bkmv	F10_BKMV(s1_Start := ?ANY16?, s2_End := ?ANY16?, d_Start => ?ANY16?);
bkmvnumberoffset	F10_BKMV_NUMBER_OFFSET(s1_Start := ?ANY16?, s2_Number := ?ANY16?, d_Offset := ?ANY16?);
adrofvar	Adr_Of_Var(?Var_ANY?);
Inttoword	INT_TO_WORD(?a_Int_INT?);

Spezifikation oder sich wiederholende Befehle, z.B. IF, CASE, FOR, WHILE...

Für die Befehle IF, CASE, FOR, WHILE, REPEAT, EXIT, RETURN steht Ihnen eine Eingabeschablone zur Verfügung, die die Programmierung mit dem ST-Editor erleichtert.

Eingabe im ST-Editor	Ergebnis nach <Strg> + <F1> oder Eingabe eines Begrenzers
IF	IF (?BOOL?) THEN END_IF;
case	case ?any_int? of x,y: x..y: else end_case;
for	for ?any_int? := ?any_int? to ?any_int? by ?any_int? do end_for; while (?bool?) do end_while;
While	While (?Bool?) Do End_While;
repeat	repeat until (?bool?) end_repeat;

Klammern

[]
---	---

Automatische Rechtschreibkorrektur für Schlüsselworte

Falsch geschriebene Schlüsselworte werden automatisch nach der Eingabe von <Strg> + <F1> oder einem Begrenzer korrigiert.

Eingang	Korrektur
endif	end_if
endcase	end_case
endfor	end_for
endwhile	end_while
endrepeat	end_repeat
elseif	elsif
else_if	
els_if	

Kommentare

Zum Auskommentieren geben Sie am Anfang der Zeile '/' ein. Stellen Sie sicher, dass der Einfügemodus auf der Tastatur aktiviert ist.


// For i:=0 to 5 do	(* For i:=0 to 5 do *)
// for i:=0 to 5 do (* comment *)	(* for i:=0 to 5 do *) (* comment *)

8.2.7.1 OP/FUN/FB-Auswahl

Sie können den Namen eines Operators oder einer Funktion (siehe Operand einfügen (siehe Seite 93)) direkt im Editor bearbeiten. FB-Instanzen werden über das Dialogfeld Variablen-Auswahl eingegeben.



♦ Vorgehensweise

1. Auf  oder **Werkzeuge → OP/FUN/FB-Auswahl** oder **<Umsch> + <F2>** klicken
Das Dialogfeld OP/FUN/FB Selection wird geöffnet.
2. **OP bzw. FUN oder FB auswählen**
3. Mit **[Insert ->Body]** oder Doppelklick in den Editor einfügen
4. **Parameterliste mit der Eingabeschablone anlegen**


oder
Parameter direkt in den Editor eingeben.

8.2.7.2 Operanden eingeben

Sie können den Namen einer Variablen (siehe Seite 85) oder Funktionsbaustein (siehe Seite 37) instanzen direkt in den Editor eingeben oder das Dialogfeld "Variablen-Auswahl" verwenden:



◆ Vorgehensweise

1. **Klick in das Programmierfenster**
2. **Klicken Sie auf  oder Werkzeuge → Variablen-Auswahl oder <F2>**
Das Dialogfeld Variablen-Auswahl wird geöffnet.
3. **Variable auswählen**
4. **Mit [Insert ->Body] oder Doppelklick in den Editor einfügen**

Für Funktionsbausteininstanzen gilt:

Parameterliste mit der Eingabeschablone anlegen oder Parameter direkt in den Editor eingeben.

8.2.8 Besonderheiten des ST-Editors

Beachten Sie beim Programmieren mit ST folgende Besonderheiten:

- Innerhalb des ST-Editors gibt es keine EN/ENO-Funktionen und Funktionsbausteine der IEC-Standard-Bibliothek.
- Als boolesche Konstanten können ebenso wie TRUE und FALSE die boolesche Null (0) und Eins (1) verwendet werden.
- Die EN-Eingangs- und ENO-Ausgangsfunktionen der F-Befehle aus der FP Library und der P-Befehle aus der FP Pulsed Library spielen keine Rolle. Diese Funktionalität wird stattdessen durch die Verwendung einer Auswahlanweisung festgelegt.

Beispiel:

```
IF start THEN
    F10_BKMV( s1_Start:= source_Array[1],
             s2_End:= source_Array[3],
             d_Start=> target_Array[0]);
END_IF;
```

- Die Adressfunktionen der FP-Tool-Bibliothek unterscheiden nicht zwischen Eingangs- und Ausgangsfunktionen, das heißt, im ST-Editor existiert jeweils nur noch eine Funktion:

Funktionen des ST-Editors	Entsprechende Funktionen der FP-Tool-Bibliothek	
Adr_Of_Var	Adr_Of_Var_I	Adr_Of_Var_O
AdrLast_Of_Var	AdrLast_Of_Var_I	AdrLast_Of_Var_O
Adr_Of_VarOffs	Adr_Of_VarOffs_I	Adr_Of_VarOffs_O
AdrDT_Of_Offs	AdrDT_Of_Offs_I	AdrDT_Of_Offs_O
AdrFL_Of_Offs	AdrFL_Of_Offs_I	AdrFL_Of_Offs_O

- Beim Debuggen einer IF- oder CASE-Kontrollstruktur wird auch dann der Programm-Code innerhalb der Kontrollstruktur durchlaufen, wenn die Kontrollbedingung unwahr ist. Die einzelnen Befehle werden jedoch nicht ausgeführt.
- Der Wert einer Laufvariablen in einer Schleife nach Ende der Schleifendurchläufe hängt von der Implementierung ab und sollte deswegen nicht abgefragt werden.
- Insbesondere bei der Verwendung von Schleifen mit vielen Durchläufen kann die Zykluszeit der SPS überschritten werden. Sie können dann versuchen diese höher zu setzen (Systemregister 30) oder die Programmabarbeitung auf mehrere Zyklen zu verteilen.
- Der Datentyp eines booleschen oder numerischen Literals kann durch Setzen eines Postfixes festgelegt werden, das aus dem Namen des Datentyps und dem Zeichen "#" besteht. Beispiel: INT#2 oder REAL-3.2.
Um Literale eindeutig identifizieren zu können, verlangen andere Programmiersysteme diese Nomenklatur. Deswegen wird sie auch von FPWIN Pro unterstützt. Dadurch, dass der Compiler intern automatisch die richtigen Datentypen den Literalen zuweist, muss die vorher beschriebene Nomenklatur nicht unbedingt eingehalten werden.
- Groß- und Kleinschreibung wird nicht berücksichtigt bei:
 - Ausdrücken (z. B. wird abcd, ABCD und aBCd gleich interpretiert)
 - Anweisungen (z. B. sind "FOR" und "for" syntaktisch gleichwertig).



◆ Hinweis

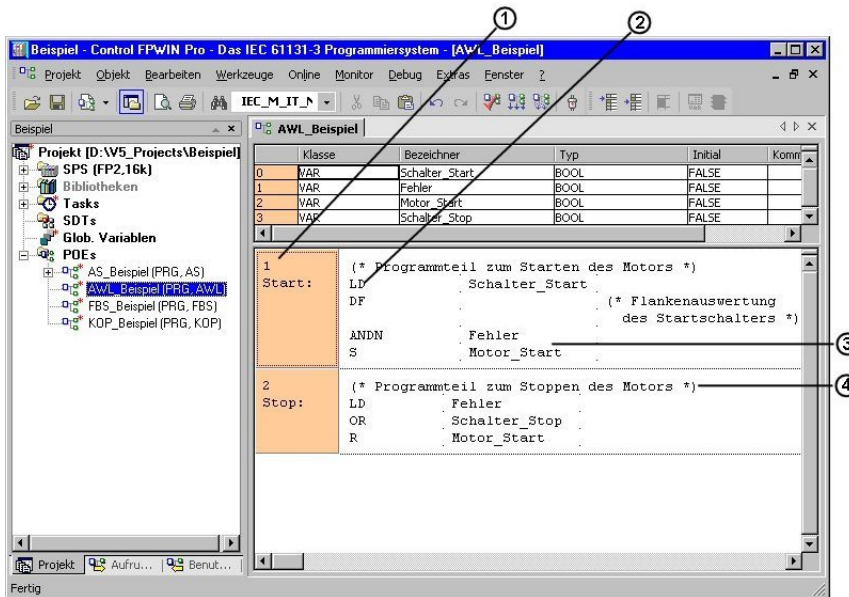
Frühere Versionen von FPWIN Pro unterscheiden zwischen Groß- und Kleinschreibung. Wenn Sie diese Syntax verwenden möchten, kann der Code mit früheren Versionen von FPWIN Pro nicht kompiliert werden.

8.2.9 Programmierbeispiel

Programmierbeispiele finden Sie in der Online-Hilfe.

8.3 AWL-Editor

Der Anweisungslisten-Editor ist ein textbasierender, syntaxfreier Editor. Hier geben Sie AWL-Befehle nach der Norm IEC 61131 (siehe Hilfe "Operatoren") oder mit dem Basisbefehlssatz ein. Die folgende Abbildung zeigt eine Anweisungsliste (AWL) mit zwei Netzwerken:



AWL-Editor Netzwerk-Infofenster

①	Kommentare
②	Operanden
③	Operatoren

Jeder POE-Rumpf besteht aus einem oder mehreren Netzwerken. Im Netzwerk-Infofenster werden die Netzwerknummern, Sprungmarken und Haltepunkte angezeigt. Außerdem sehen Sie hier, welches Netzwerk ausgewählt ist oder - im Fehlerfall - in welchem Netzwerk ein Fehler aufgetreten ist.

Das Programm wird rechts im Programmierfenster angezeigt. Es ist in drei Spalten unterteilt: **Operatoren**, **Operanden** und **Kommentare**. Der Kommentar wird durch Klammern und Sternchen (* *) begrenzt. Kommentare können mehrzeilig sein und beliebig im Programmierfenster positioniert werden. Leere Zeilen im Rumpf sind erlaubt. Jeder Rumpf kann ASCII-Quelltexte mit einer Größe von max. 60 kB enthalten (je nach SPS-Typ).

Ein AWL-Netzwerk (siehe Seite 70) muss immer mit einem Ladebefehl beginnen (LD). Verknüpfungsergebnisse werden im Bitspeicher (Akkumulator) abgelegt. Sie gehen allerdings beim Übergang von einem Netzwerk zum anderen verloren.



◆ Hinweis

- **Jedes Ergebnis, das Sie später noch benötigen, müssen Sie zwischenspeichern (Variable), bevor ein anderes Netzwerk abgearbeitet wird.**
- **Erstellen Sie keine zu großen Netzwerke, da dies die Fehlersuche und das Verfolgen von Sprüngen innerhalb des Programms sehr zeitaufwändig macht. Nutzen Sie die Vorteile einer strukturierten Programmierung.**

Weitere Informationen und Beispiele für das Arbeiten mit dem AWL-Editor finden Sie in der Hilfe zu FPWIN Pro unter dem Stichwort "Anweisungsliste (AWL)".

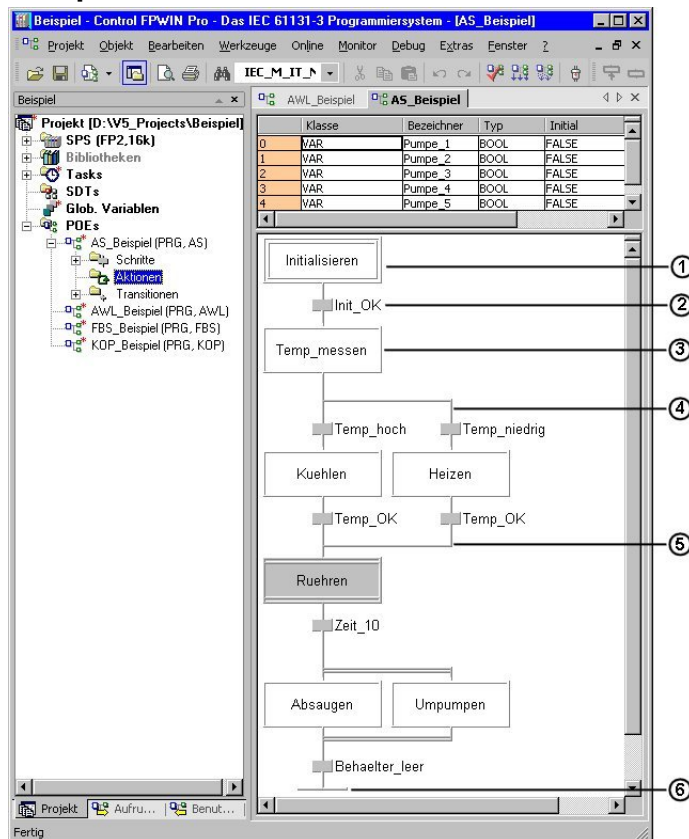
8.3.1 Eingabehilfen

Die Vorgehensweise entspricht der Vorgehensweise im KOP- und FBS-Editor. Siehe daher KOP/FBS-Programm überprüfen (siehe Seite 83).

8.4 Ablaufsprache (AS-Editor)

In der Ablaufsprache können Sie komplexe Programme übersichtlich darstellen. Dabei wird die Gesamtaufgabe in Teilaufgaben unterteilt und der Ablauf Schritt für Schritt dargestellt.

Beispiel:



AS-Editor

①	Initialisierungsschritt
②	Transition
③	Schrittsymbol
④	Verzweigung
⑤	Verknüpfung
⑥	Abschlusschritt

Folgende Symbole können verwendet werden:

Schrittsymbol

Ein Schritt ist eine Teilaufgabe, z.B. Motor einschalten.

Step1

Wenn Sie die SPS vom PROG. in den RUN-Modus schalten, ist der Initialisierungsschritt der erste Schritt, der aktiviert wird. Nacheinander werden alle Schritte abgearbeitet. Nachdem der Abschlußschritt abgearbeitet worden ist, wird der Initialisierungsschritt wieder aktiviert usw.

Jedem Schritt können Sie eine oder mehrere Aktionen zuordnen. Wenn Sie einem Schritt keine Aktion zuordnen, hat dieser Schritt eine Warte-Funktion, bis die nachfolgende Transition erfüllt ist. Aktionen werden im Projektnavigator im Aktions-Pool eingetragen und können boolesche Variablen oder Programme in AWL, FBS oder KOP sein. Sie können einem oder mehreren Schritten zugeordnet werden. Immer wenn ein Schritt aktiv ist, werden die ihm zugeordneten Aktionen ausgeführt.


Aktionen

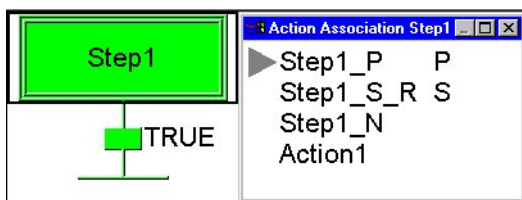
So erstellen Sie eine neue Aktion:

1. Im Navigator das SFC-Programm oder dessen Aktions-Pool auswählen




2. Bearbeiten → Neu → Aktion oder
3. Name eingeben Programmier-Editor wählen
4. [OK]

Markieren Sie einen Schritt, dem Sie eine Aktion zuweisen wollen, und klicken Sie auf  um das zugehörige Aktionsfenster zu öffnen.



◆ Hinweis

In Netzwerken von Aktionen dürfen keine Sprungmarken (siehe Seite 72) definiert werden.

Mit **Monitor** → **Monitor Kopf** oder  überwachen Sie den Status eines Schrittmerkers. Der Name dieses Merkers besteht aus dem Schrittnamen und der Erweiterung **.X**, z.B. **Step1.X**.

Makroschrittsymbol

Mehrere Schritte können in einem Makroschritt zusammengefasst werden. Ein Makroschritt wird durch zwei zusätzliche horizontale Linien gekennzeichnet. Hinter dem Makroschritt "Lueftung" befinden sich z.B. mehrere Schritte, die der Lüftungssteuerung dienen.

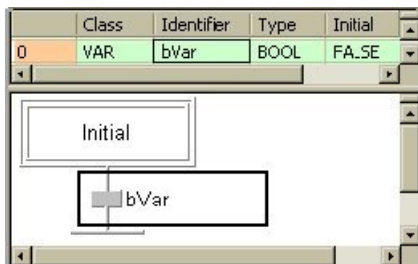


Transitionen

Eine Transition ist eine Weiterschaltbedingung. Ist die Transition erfüllt, wird der nächste Schritt aktiv.

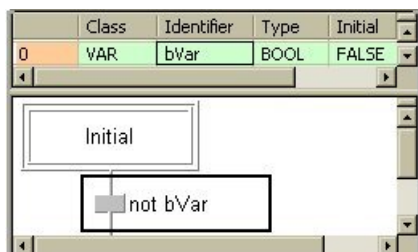
Eine Transition kann z.B. sein:

- Eine boolesche Variable (bVar oder R0)



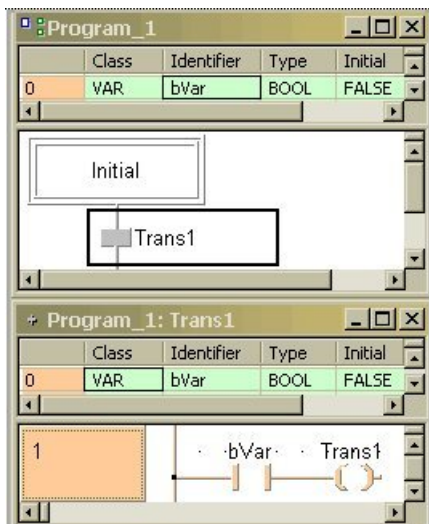
Bei einer booleschen Variablen gilt die Transition als erfüllt, wenn die zugewiesene Variable den Wert TRUE hat.

- eine Transitionsbedingung in ST-Syntax (z. B. "NOT bVar" oder "bVar1 & bVar2")



Bei einer Transitionsbedingung in ST-Syntax gilt die Transition als erfüllt, wenn der ST-Ausdruck den Wert TRUE hat.

- ein AWL-, FBS-, KOP- oder ST-Programm



Bei einem Programm gilt die Transition als erfüllt, wenn die Variable mit dem Namen der Transition den Wert TRUE hat. Die Variable mit dem Transitionsnamen wird automatisch von Control FPGWIN Pro deklariert.



◆ Hinweis

Wenn Sie einer Transitionen ein Programm zuordnen, gelten folgende Einschränkungen:

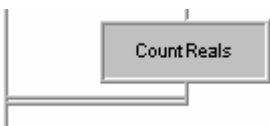
- **Im Transitionsrumpf ist nur ein Netzwerk erlaubt.**
- **EN-Funktionen dürfen nicht verwendet werden.**

Parallele Verzweigung

Eine parallele Verzweigung wird durch eine horizontale Doppellinie gekennzeichnet. Wenn die Transition vor der parallelen Verzweigung erfüllt ist, werden zwei oder mehr Schritte parallel (gleichzeitig) ausgeführt.



Nach der Ausführung, werden alle Schritte wieder in einer Transition zusammengeführt.



◆ Hinweis

Die Transition hinter einer Simultankette wird erst abgearbeitet, nachdem alle vorhergehenden Schritte abgearbeitet worden sind.

Selektive Verzweigung

Eine selektive Verzweigung wird durch eine horizontale Linie gekennzeichnet. Je nachdem, welche Transition erfüllt ist (**GoToWords** oder **GoToDoubles**), wird die zugehörige Verzweigung ausgeführt. Wenn beide Transitionen gleichzeitig erfüllt sind, gilt die Ausführungspriorität von links nach rechts, das heißt, bei der folgenden Verzweigung wird nur der Schritt nach **GoToWords** ausgeführt.



Gleich, welcher Schritt ausgeführt wird, beide werden mit folgendem Symbol wieder zusammengeführt:



Beim Programmieren im AS-Editor werden im Navigator nicht nur die POEs, sondern auch alle Aktionen und Transitionen unter **Aktionen** aufgelistet.

Die wichtigsten Schritte zum Eingeben eines AS-Programms werden in der Online-Hilfe unter "Editieren im AS-Editor" beschrieben.

Kapitel 9

Projekt in die SPS übertragen

9.1 Projekt zur SPS übertragen: Einführung

Nachdem Sie Ihr SPS-Programm programmiert haben, können Sie es in die SPS übertragen, um es anschließend zu testen. Bevor Sie jedoch das Programm übertragen, nehmen Sie folgende Einstellungen vor:

1. SPS-Typ einstellen
2. Compiler-Optionen einstellen, z.B. Adressbereiche
3. Programm kompilieren
4. Etc.

In diesem Kapitel erfahren Sie Schritt für Schritt, wie Sie vorgehen.

9.2 SPS-Typ

Mit **Online** → **SPS-Typ** können Sie Ihren SPS-Typ auswählen. Bitte beachten Sie, dass Sie den SPS-Typ nur im Offline-Modus ändern können.



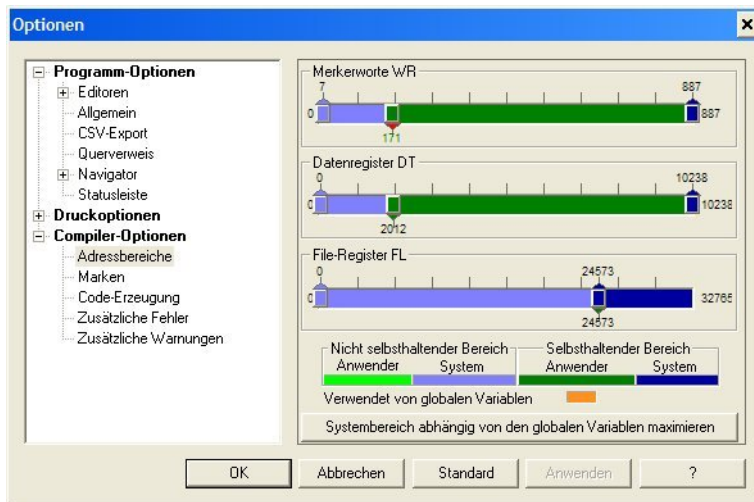
◆ Hinweis

- Wenn Sie [Standard] wählen, werden die Compiler-Optionen auf die Standardwerte zurückgesetzt. Variablen aus der Liste der globalen Variablen, die im Selbsthaltebereich gelegen haben, können danach im Nicht-Selbsthaltebereich liegen. Passen Sie daher die Compiler-Optionen und Nicht-Halte-/Haltebereiche nach einem SPS-Wechsel an. Mit Extras → Optionen → Compiler-Optionen → Code-Erzeugung "Haltende Variablen im Anwender-Adressbereich nicht initialisieren" können Sie festlegen, dass der Compiler die Variablen nicht initialisieren soll.
- Außerdem werden beim Wechsel des SPS-Typs alle Systemregister initialisiert und die Standardwerte geladen.

9.3 Adressbereiche Adressbereiche

Jede Variable, die nicht direkt einer physikalischen Adresse zugeordnet ist, muss vom Compiler einen Speicherplatz zugewiesen bekommen.

Mit **Extras** → **Optionen** → **Compiler-Optionen** → **Adressbereiche** können Sie selbsthaltende oder nicht selbsthaltende Speicherbereiche festlegen. Die Speicherbereiche sind nochmals unterteilt in System- und Anwenderbereiche. Benutzen Sie die Schieber, um die Speicherbereiche einzustellen, oder doppelklicken Sie auf einen Schieber und geben Sie den gewünschten Wert ein.



Zu den Speicherbereichen zählen:

- Merkerworte (WR)
- Datenregister (DT)
- File-Register (FL) (abhängig vom SPS-Typ).

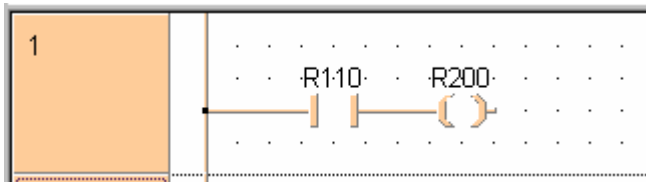
Mit **[Systembereich abhängig von den globalen Variablen maximieren]** wird der Adressbereich für das System auf den maximalen Wert eingestellt. Der Anwenderbereich wird auf den Bereich beschränkt, den der Anwender durch die Vergabe von expliziten Adressen für globale Variablen bestimmt hat.



◆ Hinweis

Verwenden Sie keine expliziten Adressen im Rumpf eines Editors. Diese Adressen werden, wenn Sie [Systembereich abhängig von den globalen Variablen maximieren] gedrückt haben, nicht berücksichtigt.

Wenn Sie zum Beispiel R110 und R200 direkt im Rumpf verwenden, werden WR11 und WR20 nicht berücksichtigt.



WR11 und WR20 werden nur dann berücksichtigt, wenn Sie globale Variablen mit expliziten Adressen verwenden, wie im folgenden Beispiel:

Global Variables

	Class	Identifier	Mats...	IEC_Address	Type	Initial
	VAR_GLOBAL	Bool_110	R110	%MX0.11.0	BOOL	FALSE
	VAR_GLOBAL	Bool_200	R200	%MX0.20.0	BOOL	FALSE

Compiler_max_user_area

	Class	Identifier	Type	Initial	Comment
0	VAR_EXTERNAL	Bool_110	BOOL	FALSE	
1	VAR_EXTERNAL	Bool_200	BOOL	FALSE	

Die Werte im nicht selbsthaltenden Bereich gehen bei einem Stromausfall oder beim Umschalten vom PROG- in den RUN-Modus verloren und werden initialisiert. Die Werte im selbsthaltenden Bereich bleiben erhalten. Sie werden erst initialisiert, wenn das Programm in die Steuerung heruntergeladen wird. Die Werte im Anwender-Adressbereich werden **nicht** initialisiert, wenn unter **Extras** → **Optionen** → **Compiler-Optionen** → **Code-Erzeugung** das Kontrollfeld "Haltende Variablen im Anwender-Adressbereich nicht initialisieren" markiert ist.



◆ Hinweis

- Die Größe der File-Register (FL) für die FP2, FP3, FP-C und FP5 stellen Sie im Projektnavigator unter SPS → Systemregister → Speicherbereich (0-3) ein.
- Die Größe der selbsthaltenden Bereiche stellen Sie im Projektnavigator unter SPS → Systemregister → Selbsthalteber. (5-18) ein.
- Wenn Sie die Compiler-Optionen ändern, müssen Sie das komplette Projekt neu kompilieren.
- Der Compiler weist globalen Variablen ohne explizite Adresse automatisch einen Adressbereich zu.

- Einige Speicherbereiche sind nicht für alle SPS-Typen verfügbar (hellgrau dargestellt).
- Für die Ein- und Ausgänge der SPS müssen Sie Adressen (X, Y) in der Liste der globalen Variablen angeben. Für alle internen Datenbereiche (DT, R usw.) vergibt der Compiler die Adressen automatisch, es sei denn, Sie haben eine Adresse eingetragen. Es wird empfohlen, alle Adressen vom Compiler verwalten zu lassen. Bei einem Austausch des SPS-Typs werden diese Adressen dann automatisch aktualisiert. Nur in zwingenden Fällen, wie bei der Adressvergabe für übergeordnete Geräte (z. B. Bediengeräte, Terminals) sollten Adressen vergeben werden.

Die folgende Tabelle zeigt, welche Klassen und Datentypen in den jeweiligen Speicherbereichen gespeichert werden:

Klasse	Datentyp	Speicherbereich
VAR, VAR_GLOBAL	BOOL	Merkerwort WR, nicht selbsthaltend
VAR_RETAIN, VAR_GLOBAL_RETAIN	BOOL	Merkerwort WR, selbsthaltend
VAR, VAR_GLOBAL	INT, DINT, WORD, DWORD, TIME, REAL, STRING	Datenregister DT, File-Register FL, Nicht selbsthaltend
VAR_RETAIN, VAR_GLOBAL_RETAIN	INT, DINT, WORD, DWORD, TIME, REAL, STRING	Datenregister DT, File-Register FL, selbsthaltend

Der Compiler erzeugt automatisch Sprungmarken (Labels), die für Schleifen benötigt werden.

Die Anzahl der Sprungmarken für das System (Compiler) können Sie unter **Extras** → **Optionen** → **Compiler-Optionen** → **Marken** festlegen.



◆ Hinweis

Mit [Standard] setzen Sie den Compiler zurück auf die Standardeinstellungen.

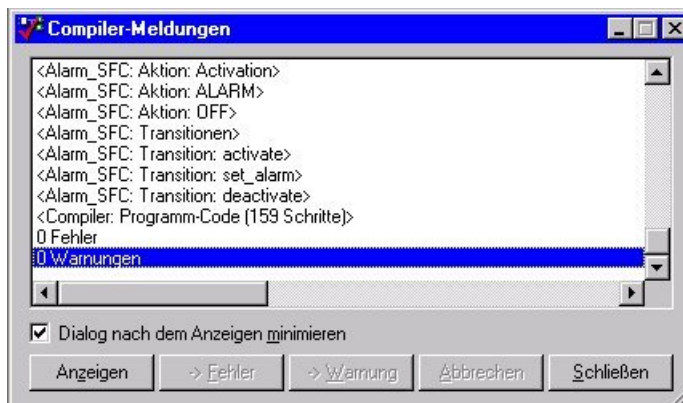
9.4 Programm kompilieren

Jedes Programm muss zuerst kompiliert werden, bevor es in die SPS übertragen werden kann. Mit **Projekt** → **Kompiliere Alles** wird das gesamte Projekt kompiliert. Dabei wird jede POE noch einmal auf Syntaxfehler überprüft und in Maschinensprache übersetzt.



◆ Vorgehensweise

1. **Projekt** → **Kompiliere Alles** oder  oder <Strg>+<Umsch>+<A>



Wenn beim Kompilieren keine Fehler auftreten, können Sie das kompilierte Projekt mit **Online** → **Programm-Code und SPS-Konfiguration übertragen** in die SPS übertragen.



◆ Hinweis

Kompiliert werden nur die POEs vom Typ Programm (PRG), die unter Tasks eingetragen sind.

9.4.1 Inkrementales Kompilieren

Alle Objekte, die im Projektnavigator mit einem Sternchen (*) gekennzeichnet sind, sind seit ihrer letzten Änderung nicht kompiliert worden. Mit **Projekt** → **Kompiliere inkremental** werden nur diese Objekte kompiliert.



◆ Vorgehensweise

1. **Projekt** → **Kompiliere inkremental** oder  oder <Strg>+<Umsch>+<I>

Wenn keine Fehler auftreten, können Sie das kompilierte Projekt mit

→Programm-Code und SPS-Konfiguration übertragen in die SPS übertragen.



◆ Hinweis

- **Mehrfache Verwendung des Befehls "Kompiliere inkremental" kann zur Fragmentierung der Speicherbelegung in der SPS führen. Es wird daher empfohlen, in regelmäßigen Abständen den Befehl Kompiliere alles (siehe Seite 109) auszuführen, um den Speicher zu defragmentieren.**
- **Eine POE ist eine inkrementale Einheit, das heißt, wenn in einer POE eine Änderung vorgenommen wird, wird die gesamte POE neu kompiliert. Um den Kompiliervorgang bei der Inbetriebnahme kurz zu halten, wird empfohlen, das komplette Programm auf möglichst viele POE zu verteilen (strukturiertes Programmieren).**
- **Kompiliert werden nur die POEs vom Typ Programm (PRG), die unter Tasks im Navigator eingetragen sind.**

9.5 Speicherbelegung prüfen

Mit **Projekt** → **Benutzer Speicher** können Sie sich anzeigen lassen, wie viel Speicherbereich verfügbar und wie viel davon belegt ist.

Benutzer Speicher		
	Verfügbar	Benutzt/Frei
<input checked="" type="radio"/> Eingang X	2048	0 / 2048
Eingangswort WX	128	0 / 128
<input type="radio"/> Ausgang Y	2048	0 / 2048
Ausgangswort WY	128	0 / 128
<input type="radio"/> Merker R	4048	31 / 4017
Merkerwort WR	253	5 / 248
<input type="radio"/> Koppelmerker L	2048	0 / 2048
Koppelmerkerwort WL	128	0 / 128
<input type="radio"/> Zeitgeberkontakt T	1000	3 / 997
Zeitgeberbaustein TM	1000	3 / 997
<input type="radio"/> Zählerkontakt C	24	0 / 24
Zählerbaustein CT	24	0 / 24
<input type="radio"/> Koppelregister Ld	256	0 / 256
<input type="radio"/> Datenregister DT	5998	3 / 5995
<input type="radio"/> File-Register FL	4093	0 / 4093
<input type="radio"/> Fehlermerker E	0	0 / 0
<input type="radio"/> Anweisung LBL	256	0 / 256
<input type="radio"/> Anweisung SUB Task 1	100	0 / 100
<input type="radio"/> Anweisung SUB Task 2	100	0 / 100
<input type="radio"/> Anweisung INT	24	0 / 24
<input type="radio"/> Anweisung SSTP	1000	3 / 997
<div> <div>Zeige Details</div> <div>Abbrechen</div> </div>		



◆ Hinweis

- **Schrittketten (SSTP)** stehen nur in der Ablaufsprache (AS) zur Verfügung. Sie können sich anzeigen lassen, wie viele Schritte zur Verfügung stehen bzw. bereits verwendet worden sind.
- Die Anweisung **SUB** ist der Speicher für benutzerdefinierte Funktionsbausteine (FB) und Funktionen (FUN).
- Die Anzahl der verfügbaren Relais und Register hängt vom jeweiligen SPS-Typ und der SPS-Konfiguration ab.

Wenn Sie **[Zeige Details]** wählen, erhalten Sie eine detaillierte Darstellung zu dem Speichertyp, den Sie im Dialogfeld "Benutzer Speicher" ausgewählt haben.

Ein-/Ausgänge, Merker, Koppelmerker, Zeitgeber, Zähler, Register, Anweisungen

Im folgenden Dialogfeld können Sie oben links die Wortadresse eintragen, die angezeigt werden soll. Im unteren Teil des Dialogfeldes werden die Zeichen erklärt, die anzeigen, ob ein Merker für den Anwender oder für das System reserviert ist und ob der Merker benutzt wird oder nicht.

Benutzer Speicher

0 Nummer Eingang X / Eingangswort WX

Nr	Rg	FEDCBA9876543210	Nr	Rg	FEDCBA9876543210	Nr	Rg	FEDCBA9876543210
0	-	-----						
1	-	uuuuuuuuuuuuuuuu						
2	-	uuuuuuuuuuuuuuuu						
3	-	uuuuuuuuuuuuuuuu						
4	-	uuuuuuuuuuuuuuuu						
5	-	uuuuuuuuuuuuuuuu						
6	-	-----						
7	-	-----						
8	-	-----						
9	-	-----						
10	-	-----						
11	-	-----						
12	-	-----						

-: Anwenderbereich nicht benutzt
 U: Anwenderbereich benutzt
 *: Systembereich nicht benutzt
 X: Systembereich benutzt

OK

Speicherbereiche für Eingänge/Ausgänge, Bitmerker und Koppelmerker

Je nach Speichertyp, den Sie unter "Benutzer Speicher" aktiviert haben, ändert sich die Anzeige.

9.6 Schnittstellen-Parameter

Überprüfen Sie die Schnittstellenparameter, bevor Sie ein Programm in die SPS übertragen. Control FPWIN Pro kann direkt über die RS232C-Schnittstelle, über eine Modemverbindung oder über Ethernet mit Ihrer SPS kommunizieren.


Die SPS kann auch in ein SPS-Netzwerk eingebunden werden. Die Einstellungen hängen vom jeweiligen SPS-Typ und der Verbindungsart ab. FPWIN Pro sucht automatisch nach den passenden Parametern, sofern diese Funktion nicht deaktiviert ist. Bitte beachten Sie auch die Hinweise in der entsprechenden Hardware-Dokumentation.

Online → **Schnittstellen-Parameter** wählen, Netzwerktyp einstellen und logisch fortfahren.

Hinweise zur Verbindung über Modem:



◆ Hinweis

- Um aus dem Dialogfeld "Kommunikationseinstellungen" direkt eine Verbindung herzustellen, klicken Sie auf [Verbinden]. Beachten Sie in diesem Fall, dass die Verbindung zum Modem NICHT mit **Online** → **Online** Mode oder  unterbrochen wird. Auch wenn Sie Control FPWIN Pro wieder offline schalten, bleibt die Modemverbindung bestehen. Wählen Sie [Trennen], um die Verbindung zur SPS abzubauen. Die Schaltfläche [Trennen] ist nur im Offline-Modus von FPWIN Pro verfügbar.
- Fügen Sie zur Statusleiste (siehe Seite 12) das Feld "Modemverbindung aktiv" hinzu, das eine bestehende Modemverbindung anzeigt.

Hinweis zur Verbindung über Ethernet:



◆ REFERENZ


Wenn Sie das ET-LAN-Koppelmodul verwenden, sollten Sie die Dokumentation zu diesem Modul gelesen haben, bevor Sie die Schnittstellen-Parameter einstellen.

9.6.1 Netzwerk-Parameter

Die Parameter für die Anbindung Ihrer SPS an ein SPS-Netzwerk, wie MEWNET-P, MEWNET-W oder C-NET, werden im Dialogfeld "Netzwerk-Parameter" eingestellt. Sie müssen angeben, auf welche SPS im Netzwerk Sie von Ihrem Rechner aus zugreifen wollen. Sie können auch auf eine SPS in einem Multi-Layer-Netzwerk zugreifen.

Mit **Online** → **Netzwerk-Parameter** öffnen Sie das Dialogfeld "Netzwerk-Parameter".

9.7 Online-Modus

Mit **Online** → **Online-Modus** oder **<Umsch> + <Esc>** oder  wechseln Sie zwischen Online- und Offline-Modus. Die Statuszeile (siehe Seite 12) zeigt an, ob der Online- oder Offline-Modus aktiv ist (falls die Statuszeile entsprechend eingerichtet wurde). Außerdem ist der Befehl **Online** → **Online-Modus** mit einem Häkchen versehen, wenn sich die Steuerung im Online-Modus befindet.

Sobald Sie vom Offline- in den Online-Modus wechseln, prüft die Software, ob der im Projekt eingestellte SPS-Typ mit der angeschlossenen SPS übereinstimmt. Sollte dies nicht der Fall sein, werden Sie aufgefordert, den erkannten SPS-Typ zu übernehmen. Wenn Sie diesen Vorgang abbrechen, bleibt Control FPWIN Pro im Offline-Modus.

Wenn Sie den SPS-Typ einstellen, müssen Sie auch die entsprechende SPS-Konfiguration vornehmen. Andernfalls schaltet FPWIN Pro nicht in den Online-Modus.

9.8 Sicherheitseinstellungen

Mit **Online** → **Sicherheitseinstellungen** zeigen Sie die aktuellen Sicherheitseinstellungen an und legen den Schutz für den Zugriff auf die SPS fest. Je nach SPS-Typ lässt sich ein Passwortschutz und ein Schutz für die Rückübertragung von der SPS festlegen.

The image displays two versions of the 'Security Settings' dialog box. The left version is for a device with no password set or a user logged in, requiring 4 hexadecimal digits for password protection. The right version is for a device where any character (8 digits) is used for password protection, upload protection is disabled, and the password retry count is 3. Both versions include fields for old and new passwords, a 'Change' button, a 'Clear' button, and a 'PLC access' section with a password field, 'Login', 'Logout', 'Help', and 'Close' buttons.

Dialogfeld für die Sicherheitseinstellung der SPS (Dialogfeld für Rückübertragungsschutz auf der rechten Seite, z.B. für FP-X)

Statusinformation

Die Statusinformation zeigt die aktuelle Zugriffsbedingung der SPS an, die entweder Passwort geschützt oder nicht geschützt sein kann.

Verbleibende Anmeldeversuche (nur für FP-X)

Im Feld Verbleibende Anmeldeversuche wird angezeigt, wieviele Gelegenheiten Sie noch haben, das richtige Passwort einzugeben. Wenn Sie dreimal hintereinander ein falsches Passwort eingeben, müssen Sie die SPS erneut ein- und ausschalten. Sie erhalten dann drei neue Versuche das richtige Passwort einzugeben.

Programmleseschutz (nur für FP-X)

Wenn der aktuelle SPS-Typ (z.B. FP-X) eine Leseschutzfunktionalität ermöglicht, aktiviert oder deaktiviert das Kontrollfeld "Programme nicht rückübertragen" diesen Schutz. Ist das Kontrollfeld aktiviert, kann keine Übertragung von der SPs auf den PC erfolgen.



◆ Hinweis

- Sobald der Rückübertragungsschutz aktiv ist, werden durch Ausschalten der Funktion der gesamte Programmcode, der Kommentarspeicher und die Passwordeinstellungen gelöscht.
- Im Kommentarspeicher werden die Projektinformationen gesichert.

Passwortfunktion

Hier können Sie ein neues Passwort einstellen oder ein vorhandenes löschen.



Wenn Sie das Passwort vergessen haben und nicht angemeldet sind, wird beim Klicken auf [Löschen] nicht nur das Passwort entfernt, sondern auch das Programm und andere Parameter der SPS, die im Kommentarspeicher aufbewahrt werden. Vorsicht! Der Vorgang kann sehr lange dauern.

Passwort einrichten

Geben Sie in das Feld "Neues Passwort" ein neues Passwort ein und wiederholen Sie dieses im darunter liegenden Feld "Passwort wiederholen". Klicken Sie auf [Ändern], um das neue Passwort einzurichten.

Passwörter ändern

Um ein Passwort zu bearbeiten, geben Sie ein neues Passwort in das Feld "Neues Passwort" ein, wie oben beschrieben. Um das vorhandene Passwort zu ändern, muss der Benutzer an der SPS angemeldet sein (siehe unten) oder das alte Passwort in das Feld "Altes Passwort" eingeben. Klicken Sie auf [Ändern], um das Passwort zu ändern.

Passwort löschen

Um das vorhandene Passwort zu entfernen, drücken Sie auf [Clear] oder ändern das Passwort (siehe oben) in ein leeres Passwort.

SPS-Zugriff

In diesem Bereich können Sie sich an der SPS an- oder abmelden.

Anmelden

Um sich an der SPS anzumelden, geben Sie das aktuelle Passwort in das Feld "Anmeldepasswort" ein und klicken auf [Anmelden]. Im Bereich "Statusinformation" sollte nun der neue Passwortstatus abzulesen sein.

Abmelden

Wenn ein Passwort eingerichtet ist und sich der Benutzer bereits an der SPS angemeldet hat, kann die SPS durch Anklicken von [Anmelden] geschützt werden. In diesem Fall ist weder eine Übertragung noch eine Rückübertragung möglich.

9.9 Projekt auf die SPS übertragen

Nachdem Sie Ihr Programm erstellt, kontrolliert und kompiliert haben, können Sie es auf die SPS übertragen (download).



◆ Vorgehensweise

1. Wenn der RUN-Modus aktiv ist: Online → SPS-Modus in PROG-Modus ändern

Auf  klicken, um zwischen PROG-Modus und RUN-Modus zu wechseln.

2. Online → Programm-Code und SPS-Konfiguration übertragen oder 

3. Online → SPS-Modus ändern (in RUN-Modus)



◆ Hinweis

Wenn Ihre SPS über den Konfigurationsspeicher verfügen kann (siehe Hardware-Beschreibung) können Sie nicht nur den Programm-Code, sondern das gesamte Projekt mit Online → Projekt auf der SPS speichern auf die Steuerung übertragen. Projektinformationen sind die Inhalte aller Editoren, die SPS-Konfiguration, die Compiler-Optionen und die Inhalte der Benutzerbibliotheken.

Die folgenden Steuerungen können über einen Konfigurationsspeicher verfügen:

SPS	Zusatzkarte erforderlich?	Wenn ja, welche?
FP-Sigma	nein	
FP2	ja	Expansion Memory Unit: FP2-EM1, FP2-EM2 oder FP2-EM3
FP2SH	nein	
FP10SH	ja	ROM Operation Board: AFP6208

9.9.1 Änderungen im RUN-Modus übertragen



Achtung:

Beim Übertragen von Änderungen mit angeschlossener Peripherie besteht bei Programmierfehlern Gefahr für Mensch und Gerät.

Sie können kleinere Änderungen in die Steuerung übertragen, ohne dass Sie in den PROG-Modus umschalten müssen. Dafür gelten folgende Beschränkungen:

- Die Variablen in der globalen Variablenliste, den SDT und POE-Köpfen sind nicht bearbeitet, hinzugefügt oder gelöscht worden.
- Es hat sich lediglich ein Programm oder eine Funktion geändert. Funktionsbausteine lassen sich nicht online ändern.
- Ein Programm darf hinzugefügt, aber nicht entfernt worden sein.
- Die SPS-Konfiguration darf sich nicht geändert haben.
- Wenn die SPS im RUN-Modus bleibt, sollte sich nur eine Anweisung geändert haben oder neu eingefügt worden sein.



◆ Vorgehensweise

1. Offline-Modus einstellen
2. Programmänderungen vornehmen
3. Online-Modus einstellen
4. Online → Änderungen im Programm-Code übertragen oder



Das Programm wird kompiliert und anschließend in die SPS übertragen. Nach erfolgreicher Änderung erhalten Sie eine entsprechende Meldung.



◆ Hinweis

- Bei den Steuerungen FP1 und FP-M vor Version 3.0 ist Online → Änderungen im Programm-Code übertragen nur im PROG-Modus verfügbar, bei allen anderen SPS-Typen auch im RUN-Modus.
- Nehmen Sie nur kleine Änderungen vor oder teilen Sie die gewünschten Änderungen in kleine Schritte ein. Im RUN-Modus können nur Programmpakete bis max. 128 Schritte kompilierter, zusammenhängender Code in die Steuerung geladen werden. Für die FP-X: 512 Schritte).


- Die FP-X bietet die Möglichkeit, den kompletten Programmcode im Run-Modus zu übertragen, wenn die Größe der Code-Änderungen 512 Schritte nicht überschreitet.
- Die Offline-SPS-Konfiguration muss mit der Online-Konfiguration übereinstimmen. Andernfalls erhalten Sie eine Fehlermeldung.

9.9.2 Programm im RUN-Modus online ändern



Achtung:

Beim Übertragen von Änderungen mit angeschlossener Peripherie besteht bei Programmierfehlern Gefahr für Mensch und Gerät.

Mit **Online** → **Online-Editieren-Modus** oder  schalten Sie in einen Modus um, in dem Sie kleine Änderungen im Rumpf Ihres SPS-Programms online vornehmen können, ohne dass Sie in den PROG-Modus umschalten müssen (Änderungen im POE-Kopf sind nicht möglich). Dazu muss das im Projektnavigator geöffnete Projekt bereits in die SPS übertragen worden sein (**Online** → **Programm-Code und SPS-Konfiguration übertragen**). Nachdem Sie die gewünschten Änderungen vorgenommen haben, wählen Sie **Online** → **Änderungen im**

Programm-Code übertragen (siehe Seite 118) oder klicken auf .






◆ Hinweis

- Bei SPS vom Typ FP1 und FP-M können Sie Online-Programmänderungen nur im PROG-Modus durchführen. Bei allen anderen SPS-Typen ist dies im PROG- und im RUN-Modus möglich.
- Nehmen Sie nur kleine Änderungen vor oder teilen Sie die gewünschten Änderungen in kleine Schritte ein. Im Online-Modus können nur Programmpakete bis max. 128 Schritte kompilierter, zusammenhängender Code in die Steuerung geladen werden. Für die FP-X: 512 Schritte.
- Die FP-X bietet die Möglichkeit, den kompletten Programmcode im Run-Modus zu übertragen, wenn die Größe der Code-Änderungen 512 Schritte nicht überschreitet.
- Neue Variablen können im Online-Modus nicht deklariert werden. Deklarieren Sie für diesen Fall einige Dummy-Variablen im Offline-Modus. Ferner können Sie auch jederzeit direkte Adressen im Rumpf eingeben.



◆ Vorgehensweise

1. **Online → Online Modus** oder 
2. **Rumpf des Programms öffnen, das Sie ändern wollen**
Dieses Programm muss sich bereits auf der SPS befinden.
3. **Online → Online-Editier-Modus** oder 
4. **Änderungen im Netzwerk vornehmen**
5. **Online → Änderungen im Programm-Code übertragen** oder auf  **klicken**
Die Änderungen werden automatisch kompiliert und in die SPS geladen.

9.9.3 Programm-Code und SPS-Konfiguration rückübertragen



Machen Sie von dieser Änderungsmöglichkeit bitte nur Gebrauch, wenn Sie kein Quellprogramm mehr zur Verfügung haben. Aus der SPS kann nur der kompilierte Code rückübertragen werden, nicht Ihr AS-, ST-, AWL-, KOP- oder FBS-Programm. Außerdem geht beim Rückübertragen die gesamte Dokumentation des Programms (wie z. B. Variablennamen) verloren.

FPWIN Pro bietet zwei Optionen an:

- Programm-Code und SPS-Konfiguration übertragen
- Programm-Code und SPS-Konfiguration übertragen und FPWIN Pro-Projekt erstellen.

Programm-Code rückübertragen

Sie können Ihr Programm aus der SPS in ein Projekt rückübertragen (Upload), am Bildschirm ansehen und ändern. Es kann jedoch nur der kompilierte Code rückübertragen werden. Das gesamte Programm wird unter "Programm-Code" im Navigator abgelegt. Sie können das geladene Programm im Monitor-Modus betrachten und gegebenenfalls auf Fehler untersuchen.

Sichern Sie das Programm auf jeden Fall (z.B. auf Diskette), **bevor** Sie es mit **Objekt → Programm-Code exportieren** ändern. Im Notfall können Sie dann wieder auf das unberührte Programm zurückgreifen.

Programm-Code rückübertragen und SPS-Programm erstellen

Ein FPWIN Pro-Projekt wird erstellt und der Programm-Code wird rückübertragen und in ein KOP-Programm konvertiert (siehe auch Import von FPWIN GR-Dateien). Sie können auf diese Weise Programme ändern, die nicht mit FPWIN Pro erstellt wurden oder keine Projektinformationen enthalten.

Verwenden Sie diese Möglichkeit nur in Notfällen, z.B. dann wenn der Quellcode aus Versehen gelöscht wurde.



◆ Hinweis

Wenn Sie ein vollständiges Projekt von der SPS in den PC übertragen möchten, verwenden Sie den Befehl **Projekt → Projekt von der SPS öffnen**. Dieser Befehl ist jedoch nur für bestimmte Steuerungen verfügbar.



◆ Vorgehensweise

1. Projekt anlegen oder öffnen
2. Online → Online-Modus oder
3. Online → Programm-Code und SPS-Konfiguration rückübertragen oder
4. Im Projekt-Navigator auf "Programm-Code" doppelklicken

Wenn Sie mit dem Programm-Code-Editor (siehe Seite 22) vertraut sind, können Sie das rückübertragene Programm prüfen und die Fehler beheben. Es kann dann kompiliert und auf die SPS übertragen werden.

Das geänderte Programm muss geprüft, kompiliert und mit **Objekt → Programm-Code und SPS-Konfiguration übertragen** auf die SPS geladen werden.



◆ Vorgehensweise

1. Online → Online-Modus oder
 2. Im Projekt-Navigator auf "Programm-Code" doppelklicken Der Programm-Code-Editor wird geöffnet.
 3. Objekt → Programm-Code und SPS-Konfiguration übertragen oder
- Der Programm-Code wird auf die Steuerung geladen.

9.9.4 Programm löschen und Systemregister zurücksetzen

Mit **Online → Programm löschen und Systemregister zurücksetzen** löschen Sie das aktuelle Programm auf der SPS und setzen die Systemregistereinstellungen der SPS auf die Herstellerwerte zurück.



Hinweis

- Um ein neues Programm auf die SPS zu übertragen, ist es nicht notwendig zuvor das alte Programm zu löschen.
- Die Herstellerwerte unterscheiden sich von den Standardeinstellungen der Systemregister in Control FPWIN Pro.

Programm und Systemregistereinstellungen der SPS werden beim Übertragen des Programms aus Control FPWIN Pro auf die SPS überschrieben.

9.9.5 Programm-Code und Systemregister vergleichen

Mit **Online** → **Programm-Code und Systemregister vergleichen** können Sie das Projekt auf Ihrem PC mit dem in der SPS vergleichen. Das Programm wird dabei aus der SPS in den PC geladen und byteweise mit dem **kompilierten** Projekt im PC verglichen. Das Ergebnis wird Ihnen in einem Meldungsfenster angezeigt.

9.9.6 Menüs / Online / SPS-Konfiguration


Mit **Online** → **SPS-Konfiguration** können Sie die SPS-Konfiguration (Systemregister, E/A Adressen, Dezentrale E/A Adressen) aus der SPS in den PC laden (Schaltfläche Rückübertragen von SPS bzw. E/A Adressen registrieren) und umgekehrt (Schaltfläche Übertragen auf SPS). Sie können die SPS-Konfiguration von Control FPWIN Pro aber auch auf die Standardwerte zurücksetzen: (**Voreinstellung** → **Projekt**).



Hinweis

Die SPS-Konfiguration kann nur im Online-Modus geladen werden.

9.9.7 SPS-Modus ändern

Mit **Online** → **SPS-Modus ändern** oder  können Sie zwischen RUN- und PROG-Modus umschalten, wenn der Betriebsarten-Wahlschalter der SPS auf REMOTE steht. Im PROG-Modus können Sie das Programm in den SPS-Speicher übertragen, im RUN-Modus läuft das Programm in der SPS ab.

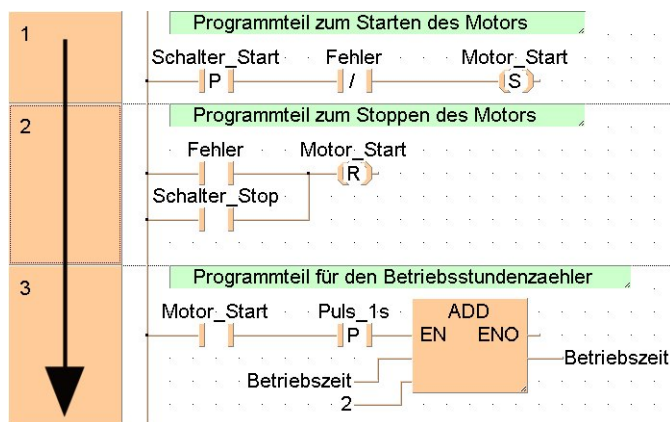
Wenn Sie das Feld "SPS-Modus" zur Statusleiste (siehe Seite 12) hinzugefügt haben, können Sie zwischen den beiden Modi auch mit einem Doppelklick auf dieses Feld umschalten.

9.10 Arbeitsweise des Compilers

Zur Vermeidung von Programmierfehlern ist es hilfreich, die Arbeitsweise des Compilers zu verstehen. Im folgenden wird deshalb erklärt, in welcher Reihenfolge der Compiler Netzwerke abarbeitet und wie Programmelemente, Sprünge, Variablen und Funktionen behandelt werden.

9.10.1 Reihenfolge beim Abarbeiten von Netzwerken in einer POE

Wenn eine POE aus **mehreren Netzwerken** besteht (AWL, KOP, FBS), werden sie nacheinander **von oben nach unten** abgearbeitet:



Abarbeitungsreihenfolge von Netzwerken

Wenn sich innerhalb eines SPS-Zyklus der Status von einem Ein- bzw. Ausgang ändert, der bereits abgearbeitet worden ist, wird diese Statusänderung erst beim nächsten Zyklus erfasst.

Innerhalb eines Netzwerks wird das Programm **von links nach rechts** und dann **von oben nach unten** abgearbeitet.

Im nächsten Abschnitt finden Sie Informationen zu der Reihenfolge, in welcher der Compiler die Elemente in einem Netzwerk auswertet und abarbeitet, die mit KOP oder FBS erstellt wurden.

9.10.2 Abarbeitungsreihenfolge in KOP- und FBS-Netzwerken

Beim Abarbeiten von Netzwerken im KOP- oder FBS-Editor gilt die folgende Reihenfolge:

1. Der Compiler beginnt links oben an der Stromschiene. Wenn Elemente direkt an die Stromschiene angeschlossen sind, wird der Strom durch alle diese Elemente geleitet. Wenn es keine direkt angeschlossenen Elemente gibt, beginnt der Compiler mit allen Eingängen. Eingänge werden von Control FPGWIN Pro so behandelt, als wären Sie zur Abarbeitung bereit.
2. Anschließend sucht der Compiler nach dem Element auf der höchsten Ebene. Wenn sich zwei Elemente auf der gleichen Ebene (Linie) befinden, arbeitet er zuerst das linke Element ab. Wenn es sich bei dem Element um einen Ausgang oder den Aufruf einer Funktion bzw. eines Funktionsbausteins handelt, erzeugt der Compiler den

entsprechenden Code. Hierbei können temporäre Variablen eingefügt werden, um das Signal zwischenspeichern bzw. im Stapelspeicher (Stack) abzulegen.

3. Der Compiler leitet den Ausgangsstrom des Elements, das im 2. Schritt abgearbeitet wurde (Zwischenergebnis), an alle direkt angeschlossenen Elemente. Dann wertet er das nächste Element aus. Anschließend beginnt er wieder bei Schritt 2, bis alle Elemente ausgewertet oder kein Element mehr zur Auswertung verfügbar ist.
4. Die Abarbeitung wird unterbrochen bzw. beendet.

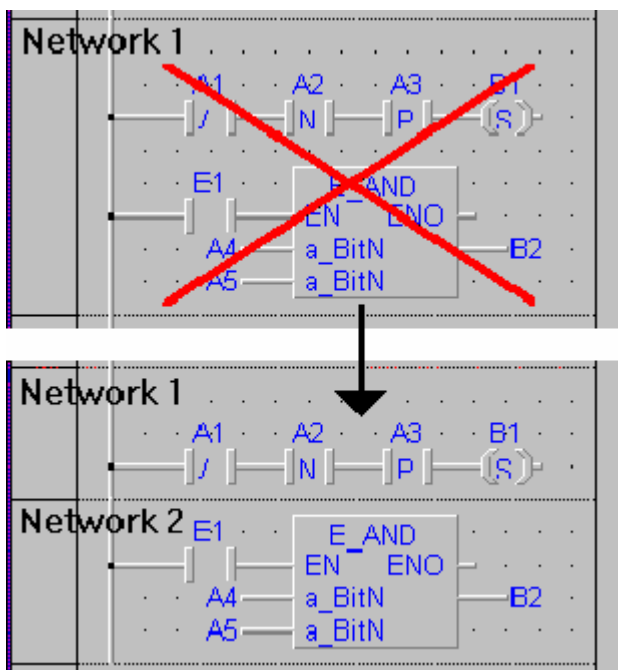
Programmbeispiele für die Abarbeitungsreihenfolge in FPGWIN Pro finden Sie in der Online-Hilfe unter dem Thema "Abarbeiten von Netzwerken im KOP-/FBS-Editor".

9.10.3 Aufteilen von Netzwerken

Es empfiehlt sich, jede einzelne Programmkette in ein eigenes Netzwerk zu stellen. Dies hat folgende Vorteile:

- Netzwerke sind klein und überschaubar
- Fehler können schneller gefunden werden, z.B. durch Netzwerk deaktivieren (siehe Seite 73)
- Die unbeabsichtigte Verwendung des "Late assignment (siehe Seite 125)" beim Zugriff auf Variablen wird vermieden.

Im oberen Teil der Abbildung befinden sich zwei Ketten in einem Netzwerk. Beim Kompilieren bzw. bei der Kontrolle erhalten Sie in einem solchen Fall eine Warnung.



Sinnvolles Aufteilen von Netzwerken

Control FPGWIN Pro behandelt Netzwerk 1 wie zwei hintereinander angeordnete Ketten, wie sie im unteren Teil der Abbildung dargestellt sind.

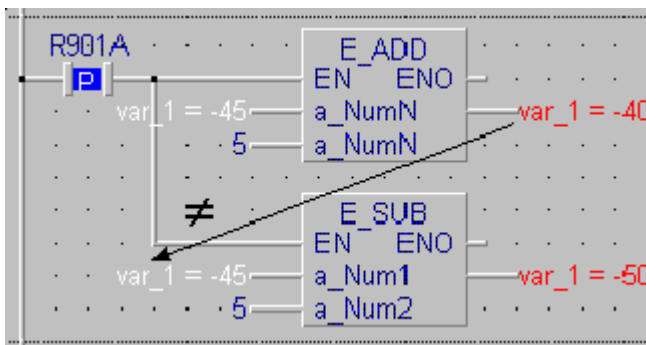
9.10.4 Behandlung von Variablen im Compiler

Wenn innerhalb eines Netzwerks der Wert einer Variablen oder Adresse gelesen wird, wird immer der Initialwert verwendet, den die Variable bzw. Adresse beim Eintritt in das Netzwerk hatte.

Das gilt auch, wenn der Wert einer Variablen bzw. Adresse geändert und anschließend gelesen wird (late assignment).

Beispiel 1:

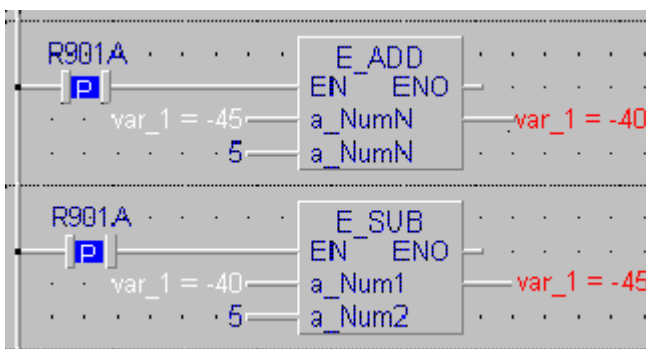
Der Wert von **var_1** wird von -45 zu -40 geändert. In der Variablen **var_1** der Funktion E_SUB wird aber der Wert -45 verwendet, den die Variable beim Eintritt des Compilers in das Netzwerk hatte. Dies wird auch als "late assignment" bezeichnet.



Das Ergebnis nach dem Abarbeiten des Netzwerks: **var_1 = -50**. Der Grund hierfür sind temporäre Variablen, die intern im Programmcode von Control FPGWIN Pro eingeführt werden. Am Ende eines Netzwerks und vor jedem Sprung (Jump) werden den Variablen die Werte dieser temporären Variablen zugewiesen.

Beispiel 2:

Hier ist das Programm auf zwei Netzwerke verteilt. In der Funktion E_SUB von Netzwerk 2 wird das Ergebnis von Netzwerk **var_1 = -40** verwendet. Nach dem Abarbeiten von Netzwerk 2 ist **var_1 = -45**.



9.10.5 Behandlung von Sprüngen im Compiler

Ganz gleich, an welcher Stelle Sie einen Sprung im Netzwerk positionieren, er wird immer erst am Ende des Netzwerks ausgeführt.

9.10.6 Behandlung von FUN/FB im Compiler

Bei der Behandlung von Funktionen (FUN) und Funktionsbausteinen (FB) unterscheidet Control FPGWIN Pro grundsätzlich zwischen den fest implementierten (FP-Anweisungen und IEC-Anweisung) und den benutzerdefinierten Funktionen und Funktionsbausteinen. Für benutzerdefinierte Funktionsbausteine können Sie bei einigen Steuerungen eine indizierte Codegenerierung aktivieren (**Extras** → **Optionen** → **Compiler-Optionen** → **Code-Erzeugung**, "Anwender-Funktionsbausteine indizieren").

9.10.7 Festimplementierte FUN/FB

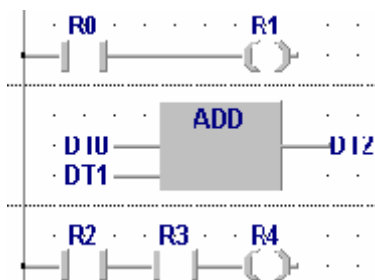
Die implementierten Funktionen (FUN) und Funktionsbausteine (FB) enthalten IEC-Operatoren, Funktionen und Funktionsbausteine aus der **IEC Standard Library** und **FP Tool Library** sowie die Befehle F und P aus der **FP Library** und **FP Pulsed Library**. Hierfür erzeugt der Compiler einen Inline-Code, das heißt, die Anweisungen (Assembler-Code) der jeweiligen Funktion werden direkt an der Aufrufstelle eingebaut und ausgeführt.

Zum besseren Vergleich des erzeugten Assembler-Codes werden im folgenden Beispiel FP-Adressen verwendet. Da das Programm bei Verwendung von FP-Adressen leicht unübersichtlich wird, empfiehlt es sich, Variablen mit sinnvollen Namen zu verwenden.



◆ BEISPIEL

Aufruf des IEC-Operators ADD im Kontaktplan-Editor:



Für die SPS wird folgender Assembler-Code erzeugt:

```
ST    R0
OT    R1
```

```

ST      R9010

F22      (*PLUS_S*)

        DT0

        DT1

        DT2      if (TRUE) DT2=DT0+DT1

ST      R2

AND      R3

OT      R4

```

9.10.8 Benutzerdefinierte Funktionen

Für die vom Benutzer im Projekt bzw. einer Benutzerbibliothek erstellten Funktionen (siehe Seite 36) und Funktionsbausteine werden im kompilierten Programm Unterprogramme angelegt. Bei jedem Aufruf werden zunächst die Eingangsparameter der Klassen VAR_INPUT und VAR_IN_OUT übergeben. Dann erfolgt ein Sprung ins entsprechende Unterprogramm. Anschließend werden die Ausgangsparameter der Klassen VAR_OUTPUT und VAR_IN_OUT sowie das Funktionsergebnis wieder gelesen. Die zulässige Anzahl von Unterprogrammen bzw. Funktionen/Funktionsbausteinen, die der Anwender definieren kann, ist vom Steuerungstyp abhängig.

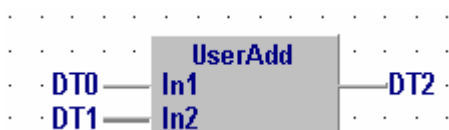
Für alle Aufrufe einer benutzerdefinierten Funktion wird nur ein Unterprogramm erzeugt, das von verschiedenen Stellen aufgerufen werden kann.

Zum besseren Vergleich des erzeugten Assembler-Codes werden im folgenden Beispiel FP-Adressen verwendet. Da das Programm bei Verwendung von FP-Adressen leicht unübersichtlich wird, empfiehlt es sich, Variablen mit sinnvollen Namen zu verwenden.

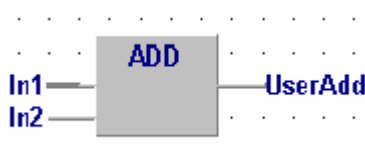


◆ BEISPIEL

Aufruf der benutzerdefinierten Funktion UserAdd im Kontaktplan-Editor



Sie hat folgenden Inhalt:



Assembler-Code für die Steuerung (Programm-Code), Erläuterungen rechts

ST	R9010		Schreiben der Eingangswerte
F0		(*MV*)	R9010 ist immer TRUE
	DT0		DT0 -> UserAdd.In1
	DT550		
F0		(*MV*)	
	DT1		DT1 -> UserAdd.In2
	DT551		
CALL	0		Aufruf des Unterprogramms UserAdd
ST	R9010		Ausgangswerte zurückladen
F0		(*MV*)	
	DT552		
	DT2		UserAdd -> DT2
...			
ED			Ende des Hauptprogramms
SUB	0		Unterprogramm UserAdd
ST	R9010		
F22		(*PLUS_S*	
)	
	DT550		
	DT551		
	DT552		if (TRUE)
			UserAdd=UserAdd.In1 + UserAdd.In2
RET			Rücksprung zum Hauptprogramm

9.10.9 Benutzerdefinierte Funktionsbausteine

Für die vom Benutzer im Projekt bzw. einer Benutzerbibliothek erstellten Funktionsbausteine wird im kompilierten Programm ein Unterprogramm angelegt. Bei jedem Aufruf werden zunächst die Eingangsparameter der Klassen VAR_INPUT und VAR_IN_OUT übergeben. Dann erfolgt ein Sprung ins entsprechende Unterprogramm. Anschließend werden die Ausgangsparameter der Klassen VAR_OUTPUT und VAR_IN_OUT sowie das Funktionsergebnis wieder gelesen. Die zulässige Anzahl von Unterprogrammen bzw. Funktionsbausteinen, die Sie definieren können, ist vom Steuerungstyp abhängig.

Funktionsbausteine besitzen ein "Gedächtnis", das heißt, jeder Instanz dieses Funktionsbausteins wird beim Aufruf ein Datenbereich und ein Unterprogramm zugeordnet. Bei einigen Steuerungen können Sie mit **Extras** → **Optionen** → **Compiler-Optionen** → **Code-Erzeugung** die Option "Anwender-Funktionsbausteine indizieren" aktivieren. Alle Instanzen eines Funktionsbausteins verwenden dann ein gemeinsames Unterprogramm. Über Indexregister wird auf den jeweils für die Instanz gültigen Datenbereich zugegriffen.

Damit lässt sich gerade bei häufiger Verwendung der Funktionsbausteine sehr viel Programmspeicher einsparen. Die Zugriffe auf den Datenbereich über Indexregister erhöhen allerdings die Zykluszeit.



◆ Hinweis

- Einige FP-Befehle arbeiten nicht korrekt, wenn sie mehrfach innerhalb eines Zyklus aufgerufen werden. Da dies bei indizierter Codegenerierung vorkommt, wenn mehrere verschiedene Instanzen eines Funktionsbausteins in einem Zyklus aufgerufen werden, erhalten Sie bei folgenden Befehlen eine Fehlermeldung:
 - Zähler: F118, F166, F167, F168
 - Schiebebefehle: SR, F119

9.10.9.1 Funktionsbausteininstanzen im Selbsthaltebereich

Instanzen von Funktionsbausteinen können im Selbsthaltebereich abgelegt werden, das heißt, alle Zustände der Funktionsbausteinvariablen bleiben nach einem Stromausfall erhalten. Dazu muß die Instanz im POE-Kopf als Klasse VAR_RETAIN oder VAR_EXTERNAL_RETAIN eingetragen sein. Im Fall von VAR_EXTERNAL_RETAIN muss die Instanz zuvor als Klasse VAR_GLOBAL_RETAIN in der globalen Variablenliste deklariert werden.

Wird für eine Instanz die Klasse VAR_RETAIN oder VAR_EXTERNAL_RETAIN verwendet, werden vom Compiler für die Nichtselbsthaltevariablen der Klassen VAR, VAR_INPUT, VAR_OUTPUT oder VAR_IN_OUT der Instanz Adressen aus dem Selbsthaltebereich vergeben und nur nach dem Runterladen des Programms initialisiert. Unterinstanzen der Klasse VAR werden wie Unterinstanzen der Klasse VAR_RETAIN behandelt. Die Adressen für sonstige Variablen oder Unterinstanzen der Klassen VAR_RETAIN, VAR_OUTPUT_RETAIN, VAR_EXTERNAL oder VAR_EXTERNAL_RETAIN werden wie für Instanzen der Klasse VAR oder VAR_EXTERNAL vergeben. Die Werte der Variablen aus den Klassen VAR_CONSTANT und VAR_EXTERNAL_CONSTANT werden als konstante Werte in den Programm-Code eingetragen.

Wird für eine Instanz die Klasse VAR oder VAR_EXTERNAL verwendet, hängt es von der Deklaration der Variablen oder Unterinstanzen des Funktionsbausteins ab, ob diese als selbsthaltend angenommen werden oder nicht.



◆ BEISPIEL

POE-Kopf des Programms 'Prog'

Ein Programm 'Prog' instanziert den Funktionsbaustein 'FB1' zweimal:

	Klasse	Bezeichner	Typ
0	VAR_RETAIN	aRetainInst1	FB1
1	VAR	alnst1	FB1

POE-Kopf des Funktionsbausteins 'FB1'

Der POE-Kopf des Funktionsbausteines 'FB1' hat die nachfolgenden Einträge: Im Funktionsbaustein 'FB1' wird wiederum auf zwei Instanzen des Funktionsbausteins 'FB2' zugegriffen.

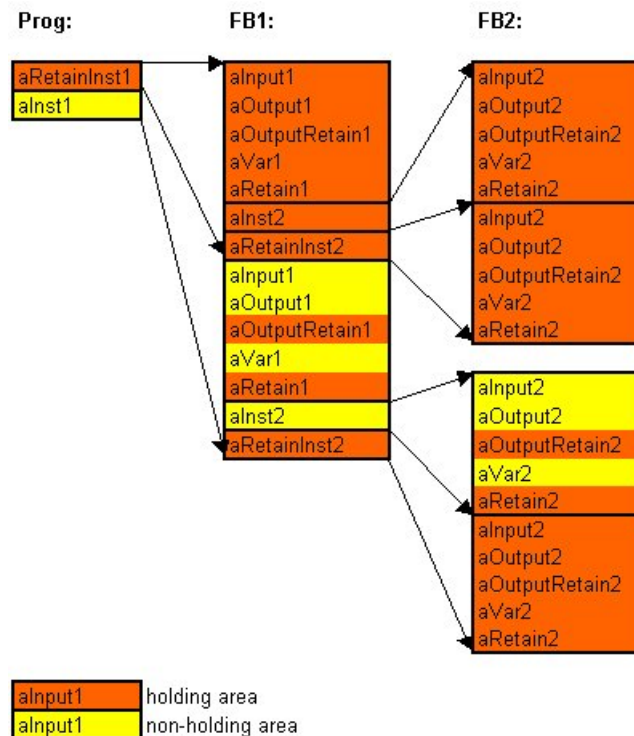
	Klasse	Bezeichner	Typ	Ir
0	VAR_INPUT	aInput1	INT	0
1	VAR_OUTPUT	aOutput1	INT	0
2	VAR_OUTPUT_RETAIN	aOutputRetain1	INT	0
3	VAR	aVar1	INT	0
4	VAR_RETAIN	aRetain1	INT	0
5	VAR	aInst2	FB2	
6	VAR_RETAIN	aRetainInst2	FB2	

POE-Kopf des Funktionsbausteins 'FB2'

Der POE-Kopf des Funktionsbausteines 'FB2' hat die nachfolgenden Einträge:

	Klasse	Bezeichner	Typ	Ir
0	VAR_INPUT	aInput2	INT	0
1	VAR_OUTPUT	aOutput2	INT	0
2	VAR_OUTPUT_RETAIN	aOutputRetain2	INT	0
3	VAR	aVar2	INT	0
4	VAR_RETAIN	aRetain2	INT	0

Speicherverteilung des Programms 'Prog':



9.10.10 Interrupt-Programme im Compiler

Die Interrupt-Programme unterbrechen das Abarbeiten der Programme des Hauptprogramms in der Taskliste "Programs". Beachten Sie folgendes:



◆ Hinweis

- Treten mehrere Interrupt-Anforderungen gleichzeitig auf, dann werden sie in der Reihenfolge ihrer Priorität, von der niedrigsten zur höchsten Nummer, ausgeführt.
- Treten während des Abarbeitens eines Interrupt-Programms weitere Interrupt-Anforderungen auf, wird erst nach dessen Abarbeitung das Interrupt-Programm mit der höchsten Priorität, d. h. mit der niedrigsten Nummer, ausgeführt.

9.10.10.1 Sichern der Indexregister in Interrupt-Programmen

Der Compiler muss dafür sorgen, dass Indexregister, die in Interrupt-Programmen bzw. in Funktionen oder Funktionsbausteinen verwendet werden, nicht überschrieben werden. Der Compiler muss dafür sorgen, dass Indexregister, die in Interrupt-Programmen bzw. in

Funktionen oder Funktionsbausteinen verwendet werden, nicht überschrieben werden. Hierzu wird z. B. folgender Code generiert:

Steuerungen mit Index-Registerbänken (FP2SH und FP10SH)

```
INT0
ST R9010
F411_CHGB K1      Index-Registerbank wird auf 1 gesetzt

ST R9010
F0_MV      DT458, I5    Eigentlicher Programmcode
F0_MV K5, I5DT455

ST R9010
F412_POPB      Zurückschalten der Index-Registerbank
```

Steuerungen ohne Index-Registerbänke

```
INT 0
ST R9010
F0_MV I5, DT461    Sichern des aktuellen Inhalts

ST R9010
F0_MV DT458, I5    Eigentlicher Programmcode
F0_MV K5, I5DT455

ST R9010
F0_MV DT461, I5    Zurückschreiben des aktuellen Inhalts
IRET
```

Kapitel 10

Fehlersuche

10.1 Fehlersuche Einführung



Beim Übertragen von Änderungen mit angeschlossener Peripherie besteht bei Programmierfehlern Gefahr für Mensch und Gerät.

Bei Panasonic-Steuerungen (außer FP-e, FP0, FP SIGMA, FP1 und FP-M) können Sie Ihr Programm im TEST-Modus Schritt für Schritt abarbeiten. Die Steuerung muss sich dabei im TEST/ RUN-Modus befinden. Stellen Sie zu diesem Zweck den Modus-Wahlschalter Ihrer SPS auf TEST (siehe Hardware-Beschreibung), den Betriebsarten-Wahlschalter auf REMOTE und wechseln Sie gegebenenfalls mit **Online** → **SPS-Modus ändern** in den RUN-Modus.

Bei der Fehlersuche (Debugging) können Sie zwischen Einzelschrittmodus und Haltepunktmodus wählen. Sie können bei der Fehlersuche den Programmablauf simulieren (das heißt, Sie haben keine Ausgänge angeschlossen).



◆ Hinweis

Wenn Sie Debug → Test Merker → Ausgänge aktivieren wählen, können Sie das Programm im Monitorbetrieb testen, ohne dass die Ausgänge Signale erhalten. Die SPS muss sich dafür im TEST/PROG-Modus befinden.

10.2 Einzelschrittmodus

Wenn Sie ihr Programm im Einzelschrittmodus testen möchten, können Sie die Auswirkung jedes Schritts im Datenmonitor beobachten und so gegebenenfalls Fehler aufdecken. Von einem Funktionsbaustein bzw. einer benutzerdefinierten Funktion aus können Sie auch in das Unterprogramm verzweigen und es schrittweise abarbeiten lassen.

Je nach Programmiersprache wird der Einzelschrittmodus wie folgt ausgeführt:

- AWL:** Eine Befehlszeile wird abgearbeitet.
- ST:** Eine Befehlszeile wird abgearbeitet.
- KOP:** Ein Netzwerk wird abgearbeitet.
- FBS:** Ein Netzwerk wird abgearbeitet.
- AS:** Ein ausgewählter Programmschritt wird abgearbeitet.



◆ Hinweis

Die POEs, die in der Task eingetragen sind, werden automatisch aufgerufen und im Einzelschrittmodus abgearbeitet.

Für die nachstehende Vorgehensweise wird vorausgesetzt, dass Sie Ihr Programm bereits auf die SPS heruntergeladen haben.



◆ Vorgehensweise

1. Projekt → Kompiliere alles **oder** Online → Online-Modus **oder** Online → Programm-Code und SPS-Konfiguration übertragen **oder in PROG-Modus wechseln**
2. **Debug → Testmerker → Einzelschritt-Modus**
3. **Modus-Wahlschalter auf TEST stellen**

4. **Online → SPS-Modus ändern auf RUN** **oder** 

5. **Debug → Schritt hinein** **oder** <F11> **oder** 

Bei Erreichen eines Funktionsbausteins bzw. einer benutzerdefinierten Funktion wird in das Unterprogramm verzweigt, das schrittweise abgearbeitet wird. Am Ende des Unterprogramms wird die Abarbeitung im aufrufenden Programm fortgesetzt.

6. **Debug → Schritt hinein** **oder** <F10> **oder** 

Ein Funktionsbaustein bzw. eine benutzerdefinierte Funktion wird nicht im Einzelschritt-Modus abgearbeitet. Verwenden Sie diesen Befehl, wenn Sie das Unterprogramm nicht mehr testen möchten.

7. **Einzelschrittmodus mit <F11> oder <F10> fortführen**

10.3 Haltepunktmodus



Beim Übertragen von Änderungen mit angeschlossener Peripherie besteht bei Programmierfehlern Gefahr für Mensch und Gerät.

Sie können mit Control FPWIN Pro in Ihrem Programm mehrere Haltepunkte (break points) setzen und das Programm dann von Haltepunkt zu Haltepunkt abarbeiten. Die SPS muss sich im Test-Modus befinden.



◆ Hinweis


- **Farben für gesetzte und aktive Haltepunkte mit Extras → Optionen → Programm-Optionen → Editoren → Format auswählen. Wählen Sie einen der Editoren aus und öffnen Sie das zugehörige Menü "Format". Wählen Sie eine Farbe für den gesetzten und aktiven Haltepunkt aus.**
- **Die POEs, die in der Task eingetragen sind, werden automatisch aufgerufen und von Haltepunkt zu Haltepunkt abgearbeitet.**

Für die nachstehende Vorgehensweise wird vorausgesetzt, dass Sie Ihr Programm bereits auf die SPS heruntergeladen haben.

Haltepunkte setzen:




◆ Vorgehensweise

1. In PROG-Modus wechseln
2. Debug → Testmerker → Haltepunkt gültig
3. AWL-Zeile oder Netzwerk im KOP bzw. FBS auswählen
4. Debug → Haltepunkt einfügen/löschen oder 

Die Zeile wird in der eingestellten Haltepunktfarbe markiert.
5. Schritte 3. und 4. wiederholen, bis alle Haltepunkte gesetzt sind
6. In RUN-Modus wechseln
7. Debug → Fortsetzen oder <F5>

<F5> Hiermit können Sie das Programm von Haltepunkt zu Haltepunkt abarbeiten.

Haltepunkt löschen:


1. In **PROG-Modus** wechseln
2. Haltepunkt auswählen
3. **Debug → Haltepunkt einfügen/löschen oder** 

Die Markierung der Zeile wird aufgehoben.

Mit **Debug → Haltepunkte** können Sie sich die aktuellen im Projekt gesetzten Haltepunkte anzeigen lassen, neue Haltepunkte einfügen oder Haltepunkte löschen.

Verwenden Sie **Debug → Testmerker → Ausgänge aktivieren**, um die Ergebnisse der Koppelungen an die Ausgänge zu senden. Beachten Sie, dass sich die Steuerung für diese Einstellung im TEST/PROG-Modus befinden muss.

10.4 Haltepunkte

Mit **Debug** → **Haltepunkte** oder  können Sie sich die im Projekt gesetzten Haltepunkte anzeigen lassen, neue Haltepunkte einfügen oder Haltepunkte löschen.

Die Felder im Dialogfeld haben folgende Bedeutung:

Task

keine Funktion

Aufrufpfad (POE)

Haltepunkt in Programmen:

Programmname eingeben

Haltepunkt in Funktionen:

Funktionsname eingeben

Haltepunkt in Funktionsbausteinen:

Programmname, FB-Instanzname

z.B.: **Programmname.FB-Instanzname1.FB-Instanzname2**

wenn z.B. in einem Funktionsbaustein ein anderer Funktionsbaustein aufgerufen wird:

Editor-Position

Netzwerknummer eingeben. Das erste Netzwerk hat die Nummer 1.

Haltebedingung

keine Funktion

Durchläufe

keine Funktion

Haltepunkte

Anzeige der gesetzten Haltepunkte.

10.4.1 Haltepunkte im AWL/ST-Editor

Sie können in jeder beliebigen Programmzeile einen Haltepunkt setzen, wenn Ihr AWL-Programm fehlerfrei kompiliert und in die SPS geladen wurde.



◆ Hinweis

Der Haltepunkt wird nur erreicht, wenn die Anweisung, auf die der Haltepunkt gesetzt ist, TRUE ist, d. h. auf der SPS ausgeführt werden kann:

In der folgenden Abbildung steht der Haltepunkt in der ersten Zeile des Netzwerks.

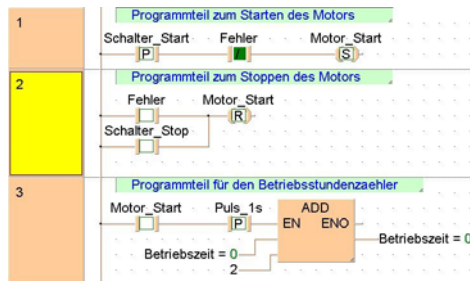
1	(* Programmteil zum Starten des Motors *)
Start:	LD <u>Schalter Start</u>
	DF <u>Fehler</u> (* Flankenauswertung des Startschalters *)
	ANDN <u>Schalter Start</u>
	S <u>Motor Start</u>
2	(* Programmteil zum Stoppen des Motors *)
Stop:	LD <u>Fehler</u>
	OR <u>Schalter Stop</u>
	R <u>Motor Start</u>

Haltepunkt im AWL-Editor

10.4.2 Haltepunkte im KOP-Editor

Im Kontaktplan- bzw. im Funktionsbaustein-Editor können beliebig viele Haltepunkte auf beliebige Netzwerke gesetzt werden.

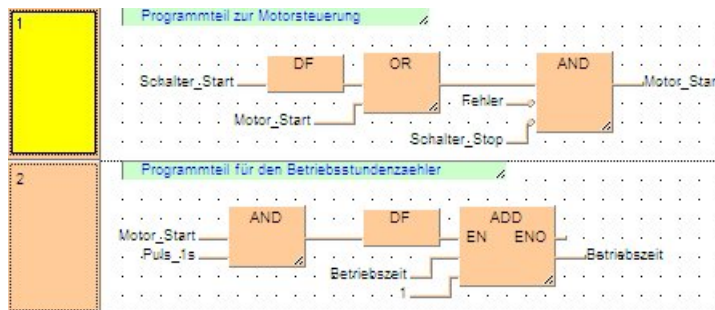
In der folgenden Abbildung wurde ein Haltepunkt im zweiten Netzwerk gesetzt und in der Haltepunkt-Farbe im Netzwerk-Infofenster markiert.



Haltepunkt in KOP-Editor

10.4.3 Haltepunkte im FBS-Editor

Im Funktionsbaustein-Editor können beliebig viele Haltepunkte auf beliebige Netzwerke gesetzt werden.

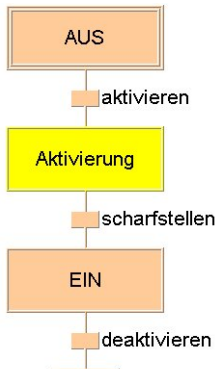


Haltepunkt in FBS-Editor

In der obigen Abbildung wird ein Haltepunkt im ersten Netzwerk gesetzt und in der Haltepunkt-Farbe im Netzwerk-Infofenster markiert.

10.4.4 Haltepunkte im AS-Editor

Im AS-Editor kann ein Haltepunkt auf einen Schritt innerhalb eines AS-Programms gesetzt werden, das erfolgreich kompiliert und übertragen worden ist.



Haltepunkt im AS-Editor

In der folgenden Abbildung wurde ein Haltepunkt auf den ersten Schritt nach dem Initialisierungsschritt gesetzt (farblich hervorgehoben).

Im AS-Programm wird der Haltepunkt nur abgearbeitet, wenn das Programm diesen Punkt auch erreicht. Sollte der Programmteil, in dem der Haltepunkt gesetzt worden ist, nicht abgearbeitet werden, wird auch der Haltepunkt nicht erreicht.





Kapitel 11

Überwachung

11.1 Monitorbetrieb Einführung

Wenn Sie Ihr Programm fehlerfrei kompiliert und in die SPS übertragen haben, gibt es mehrere Möglichkeiten, den Programmablauf zu überwachen. Im Monitor-Modus können Sie die Werte der Variablen ändern, die Wertänderung beobachten und teilweise auch Variablenwerte setzen (forcen).

Für diesen Zweck bietet Control FPWIN Pro verschiedene Monitortypen an:

Überwachte Daten	Schaltfläche
Werte (Datenmonitor)	
POE-Kopf	
Einzelwerte	
Leitdiagramme	

Außerdem können Sie folgendes prüfen:

- SPS-Status
- Koppelstatus
- Netzwerkstatus
- Status der Sondermerker
- Status der Sonderdatenregister
- Belegung des gemeinsamen Speichers



◆ Hinweis

Monitorbetrieb ist nur im Online-Modus (siehe Seite 114) möglich



◆ REFERENZ

Eine detaillierte Beschreibung der Vorgehensweisen für den Monitorbetrieb finden Sie in der Hilfe zu Control FPWIN Pro unter dem Stichwort "Monitor Menü".

Kapitel 12

Zusatzspeicher

12.1 IC-Karte, EEPROM, EPROM

IC-Karte

IC-Karten lassen sich in verschiedenen SPS-Typen als Zusatzspeicher verwenden. Mit dem IC-Karten-Manager können Sie die IC-Karte unter FFWIN Pro formatieren, beschreiben, lesen, löschen usw.

Control FFWIN Pro unterstützt IC-Karten vom Typ SRAM und F-EEPROM. Je nachdem, welcher Typ in der SPS gesteckt ist, sind bestimmte Funktionen (z.B. Formatieren) im Dialogfenster "IC-Karten-Manager" gesperrt.

EEPROM

Ein Programm kann zwischen dem RAM einer Panasonic SPS und einem EEPROM direkt ausgetauscht werden (steckbarer EEPROM bzw. EEPROM-Karte optional verfügbar, abhängig vom eingesetzten SPS-Typ).

EPROM

Ein kompiliertes Programm kann auf einer Festplatte oder einer Diskette im FP-Hex-Format, im Motorola-Hex-Format oder im Intel-Hex-Format gespeichert und mit Hilfe einer EPROM-Software in ein EPROM geladen werden.

Die Formate Motorola-Hex-Format und Intel-Hex-Format können mit einer EPROM-Software in ein EPROM geladen werden, das als Programmspeicher in die SPS gesteckt werden kann.

Bis auf die FP0 (10k), FP-Sigma, FP-M (0.9k), FP1 C14 (0.9k) und FP10/FP10S können alle Panasonic SPS-Typen ein EPROM zur Speicherung von Programmen verwenden.



◆ Hinweis

Wenn Sie ein EPROM stecken, achten Sie darauf, dass der Schreibschutzschalter der SPS auf "nicht schreibgeschützt" steht (siehe Hardware-Beschreibung) und auch die EPROM-Karte beim Beschreiben nicht schreibgeschützt ist.

Kapitel 13

Export/Import

13.1 Export/Import Einführung

Unter Control FPWIN Pro können Sie Projekte entweder vollständig oder teilweise (z. B. nur die Liste der globalen Variablen) exportieren und/oder importieren. Beim Exportieren wird eine ASCII-Datei erzeugt, die mit Control FPWIN Pro kompatibel ist und die Sie später wieder für den Import verwenden können.



◆ Hinweis

- **Der Export von Projekten eignet sich auch für die Datensicherung. Die erzeugte ASCII-Datei kann auf Diskette gespeichert werden.**
- **Bevor Sie ein Upgrade/Update von Control FPWIN Pro installieren, empfehlen wir, alle vorhandenen Projekte mit der alten Version zu exportieren und nach der Installation in die neue Version zu importieren. Damit stellen Sie sicher, dass Sie mit der aktuellen Datenbank arbeiten.**

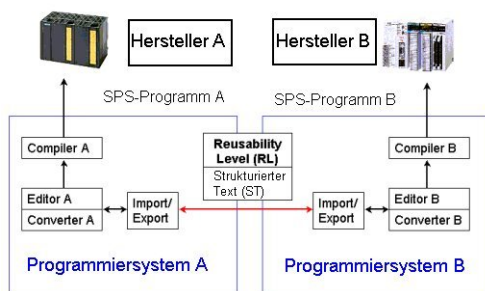
13.2 Einführung in Reusability Level ST

Control FPGWIN Pro ist nach dem Reusability Level Strukturierter Text der PLCOpen zertifiziert. Programmiersysteme mit dieser Zertifizierung garantieren, dass benutzerdefinierte Funktionen und Funktionsbausteine, die in ST programmiert sind, mit anderen zertifizierten Programmiersystemen ausgetauscht werden können.

Voraussetzung ist, dass die verwendeten Datentypen und Befehle von beiden Systemen unterstützt werden. Ein Funktionsbaustein des Herstellers A, der den Datentyp TIME verwendet, kann zum Beispiel nur dann auf dem System von Hersteller B ausgeführt werden, wenn dessen System diesen Datentyp ebenfalls unterstützt. Mit anderen Worten: Der Benutzer muss prüfen, welche Datentypen unterstützt werden.

Der Austausch der Anwenderprogramme erfolgt im ASCII-Format.

Die schematische Darstellung fasst zusammen, wie Anwenderprogramme zwischen Systemen unterschiedlicher Hersteller ausgetauscht werden können:



Sie haben die Wahl zwischen zwei verschiedenen Dateitypen:

- POE-Dateien (*.st)
Diese Dateien enthalten die POEs des ST-Editors. Eine POE kann eine Funktion, ein Funktionsbaustein oder ein Programm sein.
- Typ-Dateien (*.typ)
Diese Dateien enthalten Deklarationen, z. B. DUT-Deklarationen.

13.3 Projekte exportieren/importieren

Sie können ein vollständiges Projekt exportieren und, falls gewünscht, dieses Projekt auf einem anderen Datenträger sichern.

Projekt exportieren



♦ Vorgehensweise

1. **Projekt → Projekt exportieren**
2. **Laufwerk und Verzeichnis auswählen, in dem die Datei gespeichert werden soll**
3. **Dateityp auswählen**
Mögliche Dateitypen sind:
FPWIN-Pro-Dateien (*.asc), Reusability-Level-POE- (*.st) und Reusability-Level-Typ-Dateien (*.typ).
4. **Dateinamen eingeben**
5. **[OK]**

Falls der Dateiname bereits existiert, erscheint eine Abfrage. Klicken Sie auf [Ja], um den Dateinamen zu überschreiben oder auf [Nein], um mit dem Export fortzufahren.



♦ Hinweis

- Die Dateitypen "Reusability Level: POE-Dateien (*.st)" und "Reusability Level:
- Die Dateitypen "Reusability Level: Typ-Dateien (*.typ)" ist nur verfügbar, wenn strukturierte Datentypen im SDT-Pool angelegt wurden. Einführung in Reusability Level

Projekt importieren

Folgende Dateien können mit der Funktion "Projekt importieren" importiert werden:

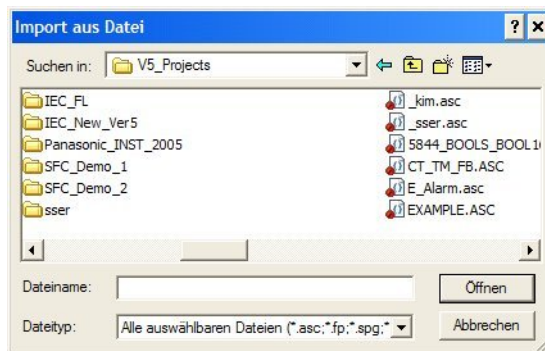
- FPWIN-Pro-Dateien (*.asc)
- FPWIN-GR-Projektdateien (*.fp)
- NPST-GR-Dateien (*.spg)
- Reusability-Level-POE-Dateien (*.st)
- Reusability-Level-Typ-Dateien (*.typ)



◆ Vorgehensweise

1. Projekt → Projekt importieren

Bevor Sie ein Projekt importieren, müssen Sie ein leeres Projekt mit Projekt → Neu anlegen oder ein Projekt mit Projekt → Öffnen öffnen.



2. Verzeichnis auswählen, in dem die Datei gespeichert ist

3. Dateityp auswählen

4. Dateiname auswählen

Benutzen Sie <Umschalt> oder <Strg> für die Mehrfachauswahl.

5. [Öffnen]

Das Projekt der ausgewählten Datei wird in Control FPWIN Pro importiert. Wählen Sie [Abbrechen], wenn Sie den Importvorgang abbrechen möchten.



◆ REFERENZ

Hinweise zum Import von FPWIN-GR-Dateien finden Sie in der Online-Hilfe (Schlüsselwort "FPWIN GR Datei").

Beispiele für den Import von FPWIN-GR-Dateien

13.4 Objekte exportieren/importieren

Objekte, wie z. B. POE, globale Variablenliste, Tasks etc. können Sie beliebig exportieren. Sie haben damit die Möglichkeit, z. B. nur bestimmte Programmteile auf einer Diskette/Festplatte zu sichern und in einem anderen Projekt weiterzuverwenden.

Objekte exportieren:



♦ Vorgehensweise

1. Im Projektnavigator Objekt auswählen, das Sie exportieren wollen

Benutzen Sie <Umschalt> oder <Strg> für die Mehrfachauswahl.

Öffnen Sie das Objekt nicht, denn sonst ist das Menü "Objekt exportieren" nicht aktiv.

2. Objekt → Objekt exportieren

3. Verzeichnis auswählen, in dem die Datei gespeichert werden soll

4. Dateityp auswählen

Mögliche Dateitypen sind: FPWIN Pro-Dateien (*.asc). Beim Export von SDTs oder globalen Variablen, sind auch die Dateitypen *.st und *.type verfügbar.

5. Name der Datei eingeben, in der das Objekt gespeichert werden soll

6. [Speichern]

Mit **Objekt → Objekt exportieren** können Sie einige oder alle Objekte in eine mit FPWIN Pro kompatible Datei exportieren.



♦ Hinweis

- Die Dateitypen "Reusability Level: POE-Dateien (*.st)" und "Reusability Level:
- Die Dateitypen "Reusability Level: Typ-Dateien (*.typ)" ist nur verfügbar, wenn strukturierte Datentypen im SDT-Pool angelegt wurden. Einführung in Reusability Level

Objekt(e) importieren

Objekte, wie z. B. POE, globale Variablenliste, Tasks usw. können Sie beliebig importieren. Sie haben damit die Möglichkeit, z. B. nur bestimmte Programmteile auf einer Diskette/Festplatte zu sichern und in einem anderen Projekt weiterzuverwenden. Mit **Objekt → Objekt importieren** können Sie auch einzelne Objekte aus einem vollständig exportierten Projekt importieren.

Folgende Dateien können mit der Funktion "Objekt importieren" importiert werden:

- FPWIN-Pro-Dateien (*.asc)

- FPWIN-GR-Projektdateien (*.fp)
- NPST-GR-Dateien (*.spg)
- Reusability-Level-POE-Dateien (*.st)
- Reusability-Level-Typ-Dateien (*.typ)



◆ Vorgehensweise

1. Objekt(e) im Navigator auswählen, z. B. POEs und/oder Tasks

Importiert werden Objekte, die der Auswahl entsprechen. Benutzen Sie die Tasten <Umschalt> und/oder <Strg> für die Mehrfachauswahl.

2. Objekt → Objekt importieren

3. Verzeichnis auswählen, in dem die zu importierende Datei gespeichert ist

4. Dateiname auswählen

Benutzen Sie die Tasten <Umschalt> und/oder <Strg> für die Mehrfachauswahl.

5. [OK] wählen

Die Informationen aus der gewählten Datei werden in das aktuelle Projekt importiert und unter den entsprechenden Objekten im Navigator eingetragen.

Sobald der Importvorgang abgeschlossen ist, erscheint eine Systemmeldung.

13.5 Programm-Code exportieren/importieren

Mit **Objekt → Programmcode exportieren** können Sie den kompilierten Programmcode im Programm-Code-Editor in einer Datei speichern.



◆ Vorgehensweise

1. Im Projekt-Navigator auf "Programm-Code" doppelklicken
2. Objekt → Programm-Code exportieren
3. Dateinamen *.asc eingeben und Verzeichnis wählen
4. Dateityp *.asc auswählen Verzeichnis auswählen [OK]

Mit **Objekt → Programmcode importieren** können Sie den kompilierten Programmcode aus einer Datei in den Programm-Code-Editor laden.



◆ Vorgehensweise

1. Programm-Code im Navigator öffnen
2. Objekt → Programm-Code importieren
3. Datei *.asc auswählen
4. [OK]

Kapitel 14

Tastaturbelegung

14.1 Tastaturbefehle für allgemeine Funktionen

In FPWIN Pro lässt sich komfortabel und einfach mit der Maus und/oder Tastatur programmieren. Eine vollständige Liste aller verfügbaren Tastaturbefehle finden Sie in der Online-Hilfe (Schlüsselwort "Tastaturbefehle").

Taste		<Umschalt>	<Strg>	<Alt>	<Umsch> + <Strg>
			<1>Navigator ein/aus <a>Alles markieren <c>Kopieren <f>Suchen <h>Ersetzen <k>Haltepunkte bearbeiten <n>Projekt neu <o>Projekt öffnen <p>Drucken <q>Druckvorschau <s>Speichern <v>Einfügen <x>Ausschneiden <y>Wiederholen <z>Widerrufen	<unterstrichenes Zeichen> Menü aktivieren <0> Navigator aktivieren	<c>Programm überprüfen (siehe Seite 32) <a>Compiliere alles (siehe Seite 109) <i>Inkrementell es Compilieren (siehe Seite 109)
<Esc>	Änderung in Eingabefeld widerrufen und Feld schließen	Online-Modus (siehe Seite 114)			
<Tab>	Nächstes Feld Nächste Zeile Zeile am Ende hinzufügen	Voriges Feld Letztes Feld der vorigen Zeile Zeile am Anfang hinzufügen	Fenster wechseln		
<Leerzeichen>	Feld öffnen oder Texteingabe, wenn Feld offen			Systemmenü öffnen	

Taste		<Umsch>	<Strg>	<Alt>	<Umsch> + <Strg>
<Eingabe> oder <Enter>	Feld schließen AWL: Neue Zeile im gleichen Netzwerk	Ins nächste Feld der gleichen Spalte Ins nächste Netzwerk	In Kommentar: Absatzzeichen einfügen Netzwerk schließen	Objekteigensch aften	Ins vorige Netzwerk
<Rücktaste>	Ausgewählte Zeichen löschen Zeichen links vom Cursor löschen		Wiederherstelle n	Rückgängig	
<Einf>		Einfügen	Kopieren		
<Entf>	Ausgewählte Zeichen löschen Zeichen rechts vom Cursor löschen	Ausgewählte Zeichen ausschneiden			
<Pos 1>	Zum Zeilenanfang	Von der Cursorposition bis zum Zeilenanfang markieren	Zum Textanfang		Von der Cursorposition bis zum Textanfang markieren
<Ende>	Zum Zeilenende	Von der Cursorposition bis zum Zeilenende markieren	Zum Textende		Von der Cursorposition bis zum Textende markieren
<Bild auf>	Seite nach oben		Seite nach links		
<Bild ab>	Seite nach unten		Seite nach rechts		
<↑>	Zeile nach oben				
<↓>	Zeile nach unten Auswahl "Klasse" oder "Typ" öffnen				
<←>	Zeichen nach links Feld nach links	Zeichen links vom Cursor markieren			

Taste		<Umschalt>	<Strg>	<Alt>	<Umsch> + <Strg>
<→>	Zeichen nach rechts Feld nach rechts	Zeichen rechts vom Cursor markieren			
<->	Unterelemente ausblenden				
<+>	Unterelemente einblenden				
<F1>	Hilfethema aufrufen		AWL: In Operandenspalte der Eingabeschablone einfügen		
<F2>	Auswahl für Variablen Funktionen Funktionsbausteine öffnen				
<F4>			Aktuelles Fenster in FPWIN Pro schließen	Beenden von Control FPWIN Pro	
<F5>	Debugging fortsetzen				
<F6>			Zum nächsten Fenster in FPWIN Pro wechseln		Zum vorigen Fenster in FPWIN Pro wechseln
<F9>	Haltepunkte einfügen/löschen				
<F10>	Schritt über (Debugging)				
<F11>	Schritt hinein (Debugging)				

Kapitel 15

Glossar

Ablaufsprache (AS-Editor)

Die Ablaufsprache ist eine Programmiersprache, die nur für den POE-Typ PRG verwendet werden kann. In der Ablaufsprache werden die verschiedenen Programmschritte über Weerschaltbedingungen miteinander verknüpft.

Die Elemente der Ablaufsprache sind: Schritte, Aktionen, Transitionen und Transitionsbedingungen. Mit der Ablaufsprache wird die Steuerungsaufgabe modularisiert.

Akkumulator

Ein Akkumulator ist ein Speicher, der das Verknüpfungsergebnis des zuletzt ausgeführten AWL-Befehls enthält. Ein Akkumulator kann auch Eingangsbedingungen für nachfolgende Operationen enthalten. Es wird immer mit dem aktuellen Inhalt des Akkumulators weitergearbeitet.

Aktion

Jedem Schritt im Ablaufdiagramm ist eine Aktion zugeordnet. Eine Aktion enthält Teile einer übergreifenden Logik. Die Aktion kann eine boolesche Ausgangsvariable oder ein anderes SPS-Programm sein.

Aktions-Pool

Der Aktions-Pool wird im Projektnavigator unter einer Programm-Organisationseinheit (POE) angezeigt, die in der Ablaufsprache erstellt wird. Im Aktions-Pool stehen alle Aktionen, die zu dieser POE programmiert werden.

Anweisungsliste (AWL)

Die Anweisungsliste ist eine Programmiersprache. In Control FPWIN Pro wird unterschieden zwischen der IEC-Anweisungsliste und der FP-Anweisungsliste.

Array

Ein Array ist ein benutzerdefinierter Datentyp, der aus mehreren gleichen elementaren Datentypen zusammengesetzt wird und maximal dreidimensional sein kann.

Ausgangsvariable

Die Funktionen und Funktionsbausteine schreiben Ergebnisse in Ausgangsvariablen.

Bezeichner

Der Bezeichner ist der symbolische Name einer Variablen.

Bibliotheken

Es stehen vier Hersteller-Bibliotheken und zahlreiche Benutzerbibliotheken zur Verfügung: Weitere Informationen finden Sie in der Online-Hilfe.

Datentyp

In FPWIN Pro wird zwischen elementaren und benutzerdefinierten Datentypen unterschieden. Zu den **elementaren Datentypen** gehören: BOOL, INT, DINT, WORD, DWORD, REAL, TIME und STRING.

Deklaration:

Eine Deklaration ist die Definition einer globalen oder lokalen Variablen. Globale Variablen werden in der Liste der globalen Variablen im Projektnavigator deklariert, lokale Variablen im jeweiligen Kopf der POE.

Eingangsvariable

Die Eingangsvariablen liefern einem Funktionsblock/einer Funktion die Werte zur Durchführung einer Rechenoperation.

F-Befehle

F-Befehle sind Befehle aus dem Befehlssatz von Panasonic, die immer ausgeführt werden, wenn am EN-Eingang der Wert TRUE anliegt. Die fast gleichlautenden P-Befehle werden dagegen nur bei steigender Flanke des EN-Eingangs ausgeführt.

Funktion

Eine Funktion ist eine Programm-Organisationseinheit (POE), die nach dem Ausführen ein Datenelement als Funktionsergebnis und beliebig viele Ausgangswerte liefert. Eine Funktion enthält keine internen Zustandsinformationen, d. h., der Aufruf einer Funktion mit gleichen Eingangswerten liefert immer die gleichen Ausgangswerte.

Funktionsbaustein (FB)

Ein Funktionsbaustein ist eine Programm-Organisationseinheit (POE), die im Gegensatz zur Funktion einen durch den Instanznamen festgelegten Zwischenspeicher besitzt. Bei jedem Aufruf eines Funktionsbausteins können Sie die Instanz angeben, für die ein eigener Datenbereich und ggf. ein eigener Codebereich reserviert wird.

Funktionsbausteinsprache (FBS)

Die Funktionsbausteinsprache ist eine Programmiersprache. Die Programmieranweisungen werden mit Programmiersymbolen realisiert, die mit Hilfe von horizontalen und vertikalen Verbindungslinien miteinander verknüpft werden.

Globale Variablen

Globale Variablen haben physikalische Adressen. Sie gelten für das gesamte Projekt und können in den Köpfen der Programm-Organisationseinheiten als VAR_EXTERNAL übernommen werden.

Instanz

Eine Instanz ist eine Kopie von einem Funktionsbaustein. Funktionsbausteine können nur als Instanzen aufgerufen werden.

Kompilieren

Beim Kompilieren wird das in Control FFWIN Pro erstellte Projekt in Maschinensprache übersetzt, so dass es von der SPS gelesen werden kann.

Kontaktplan (KOP)

Der Kontaktplan ist eine Programmiersprache. Seine Elemente sind Kontakte, Spulen, Funktionen und Funktionsbausteine. Diese Elemente werden mit Hilfe von horizontalen und vertikalen Verbindungslinien mit der Stromschiene verbunden.

Logik

Komplettes SPS-Programm, das vom Anwender zur Lösung von Automatisierungsproblemen verwendet wird. Die Logik des Anwenders ist in POE strukturiert.

Lokale Variablen

Lokale Variablen gelten nur in der Programm-Organisationseinheit, in deren Kopf sie deklariert wurden.

Maschinen-Programm

Maschinen-Programme können nur mit FP2, FP3 und FP5 verwendet werden. Über "Maschinen-Prog." (im Projektnavigator) können Maschinenprogramme eingebunden werden.

Panasonic Electric Works Europe AG bietet Maschinen-Programme an, die bestimmte Teilaufgaben der Programmierung erleichtern, es gibt z. B. einen PID-Regler für die Steuerung FP3.

Netzwerk

Ein Netzwerk stellt einen zusammenhängenden logischen Programmabschnitt innerhalb einer Programm-Organisationseinheit dar.

Netzwerkliste

In der Netzwerkliste können Sie sich einen Überblick über die Netzwerke verschaffen, die Sie in Ihrem Programm verwenden (**Werkzeuge** → **Netzwerkliste**).

Objekt

Objekte in Control FPWIN Pro sind alle Komponenten, die im Projektnavigator aufgelistet werden.

Online

Online bedeutet, dass der PC und die SPS miteinander verbunden sind. Im Online-Modus können Sie die SPS direkt programmieren. **VORSICHT! Wenn Ausgänge angeschlossen sind, kann ein Fehler in der Programmierung zu Schäden an Mensch und Maschine führen.**

P-Befehle

P-Befehle sind Befehle aus dem FP-Befehlssatz und werden im Gegensatz zu F-Befehlen nur ausgeführt, wenn eine steigende Flanke anliegt. Wählen Sie im Dialog OP/FUN/FB-Auswahl die Schaltfläche [P-Anweisung einfügen] wenn Sie einen P-Befehl benötigen.

Programm

Ein Programm ist mit einem Funktionsbaustein vergleichbar. Die Unterschiede zwischen Programmen und Funktionsbausteinen sind:

- Programme sind nur an der Spitze einer POE-Aufrufhierarchie zulässig (ein Programm kann also, anders als ein Funktionsbaustein, nicht von einer anderen POE aufgerufen werden).
- Zur Entwicklung eines Programms lassen sich Variablen verwenden.

Programm-Organisationseinheiten (POE)

Die Programm-Organisationseinheit (POE) entspricht einem Teilprogramm. Sie kann einer der folgenden Programmklassen zugeordnet sein:

Programm (PRG), Funktionsbaustein (FB) und Funktion (FUN).

Bei POEs in FFWIN Pro wird zwischen **Kopf**, der die deklarierten Variablen enthält, und **Rumpf**, der den POE-Algorithmus enthält unterschieden.

Projekt

Das Projekt stellt die höchste Hierarchiestufe von Control FFWIN Pro dar. Es beinhaltet die gesamte Steuerungsaufgabe.

Projektnavigator

Der Projektnavigator ist der Kompass eines Projekts in Control FFWIN Pro. Er enthält alle Objekte, die zu einem Projekt gehören: Standardbibliothek, Herstellerbibliothek, SPS-Konfiguration, Task-Pool, SDT-Pool, globale Variablen und POE-Pool.

SDT

Mit einem strukturierten Datentyp (SDT) können zusammengesetzte Datentypen definiert werden. Ein SDT wird zunächst im SDT-Pool definiert und dann wie die Standardtypen (BOOL, INT usw.) in der globalen Variablenliste oder im POE-Kopf verarbeitet.

SDT-Pool

Der SDT-Pool befindet sich im Projektnavigator und enthält alle strukturierten Datentypen. Strukturierte Datentypen sind benutzerdefiniert.

Strukturierter Text

Der ST-Editor ist ein textbasierender, syntaxfreier Editor, mit dem komplexe Ausdrücke und Kontrollstrukturen in Hochsprache programmiert werden können. Er ist für alle Steuerungen verfügbar und belegt dabei nicht mehr Ressourcen (d.h. Schritte, Labels oder Calls) als andere Editoren bei vergleichbarer Programmierung.

Task

Eine Task ruft eine oder mehrere Programm-Organisationseinheiten vom Typ PRG auf und steuert deren Ausführung. Eine Task kann ein Programm ereignisgesteuert, zyklisch oder zeitgesteuert (in Intervallen) aufrufen.

Variable

Variablen sind vergleichbar mit Operanden. Sie enthalten die Werte von Eingängen, Ausgängen oder internen Speicherbereichen der SPS. Es wird unterschieden zwischen globalen Variablen und lokalen Variablen. Globale Variablen werden in der Liste der globalen Variablen deklariert, lokale Variablen im jeweiligen POE-Kopf.

Index

A

Abarbeitungsreihenfolge in KOP- und FBS-Netzwerken	123
Ablaufsprache (AS-Editor)	98
Ablaufsprache-Editor	98
Adr_Of_Var	94
Adressbereiche Adressbereiche	106
Adressen	45
Akkumulator	158
Aktion	158
Aktions-Pool	158
Änderungen im RUN-Modus übertragen	118
Änderungsverzeichnis	167
Anweisungen	87
Anweisungsliste (AWL)	158
Anweisungsliste-Editor	96
Arbeitsweise des Compilers	123
Array	158
ARRAY	60
Array und strukturierter Datentyp	60
Assistent	6
Aufrufbaum	30
Aufteilen von Netzwerken	124
Ausdrücke	85
Ausgangsvariable	158
AWL-Editor	96

B

Behandlung von FUN/FB im Compiler ..	126
Behandlung von Sprüngen im Compiler	126
Behandlung von Variablen im Compiler	125

Benutzerdefinierte Funktionen	127
Benutzerdefinierte Funktionsbausteine	128
Benutzeroberfläche	5
Benutzt-von-Liste	31
Besonderheiten des ST-Editors	94
Bezeichner	158
Bibliotheken	22, 158

C

Comma Separated Variables	58
CSV	58
Import	58

D

Datensicherung (Backup) erstellen	26
Datentyp	158
Deklaration:	159

E

E/A Adressen	18
Editoren	85, 96, 98
Strukturierter Text	85
Ein bestehendes Projekt öffnen	25
Einführung in Reusability Level ST	147
Eingabehilfen	90, 97
Eingabeschablonen für ST-Editor	90
Eingangsvariable	159
Einzelschrittmodus	135
EN/ENO im KOP	82
Enable Input und Enable Output	82
Erste Schritte	3
Export/Import	145
Export/Import Einführung	146

F

F-Befehle	159
-----------------	-----

Fehlersuche	133
Fehlersuche Einführung	134
Festimplementierte FUN/FB	126
Funktion	159
Funktion (FUN).....	38
Funktionsbaustein (FB).....	39, 159
Funktionsbausteininstanzen im Selbsthaltebereich	129
Funktionsbausteinsprache (FBS).....	159

G

Gesichertes Projekt wiederherstellen	25
Globale Variablen	22, 44, 159
Globale Variablen deklarieren.....	45
Globale Variablen in den POE-Kopf übernehmen	47
Glossar	157

H

Haltepunkt einfügen	136
Haltepunkt löschen	136
Haltepunkte	138
Haltepunkte im AS-Editor.....	140
Haltepunkte im AWL/ST-Editor	138
Haltepunkte im FBS-Editor	139
Haltepunkte im KOP-Editor	139
Haltepunktmodus	136
Hauptfenster.....	7

I

IC-Karte, EEPROM, EPROM.....	144
Inkrementales Kompilieren	109
Installation und erste Schritte.....	1
Instanz.....	159
Interrupt-Programme im Compiler	131

K

Kommentare	90
Kompilieren	159
Kontaktplan (KOP)	160
Kontextmenü	81
Kontrolle	32, 83
KOP oder FBS-Programm überprüfen.....	83
KOP- und FBS-Editor.....	78

L

Logik.....	160
Lokale Variablen	46, 160
Lokale Variablen (VAR)	46
Lokale Variablen deklarieren	49

M

Marken und Titel definieren	74
Maschinen-Programm.....	160
Menüs / Online / SPS-Konfiguration	122
MEWNET-F-Adressliste.....	20
Monitorbetrieb Einführung.....	142

N

Navigator.....	30, 31
Netzwerk	160
Netzwerk bearbeiten	74
Netzwerk deaktivieren/aktivieren	75
Netzwerke	71
Netzwerke auswählen.....	72
Netzwerke Einführung.....	72
Netzwerkliste.....	72, 73, 160
Netzwerk-Parameter	113
Neues Projekt erstellen.....	24
Nützliche Tipps	80

O

Objekt.....	160
Objekte bearbeiten.....	27
Objekte exportieren/importieren	150
Objekte verbinden.....	79
Online.....	160
Online-Editieren-Modus	119
Online-Modus.....	114
OP/FUN/FB-Auswahl	93
Operanden	86
Operanden eingeben	93
Operatoren	86

P

Passwörter und Sicherheitsstufen	28
P-Befehle	160
POE neu anlegen.....	41
POEs.....	23
Programm	160
Programm einer Task nachträglich zuweisen	65
Programm im RUN-Modus online ändern	119
Programm installieren und starten.....	2
Programm kompilieren.....	109
Programm löschen und Systemregister zurücksetzen	121
Programm-Code	22
Programm-Code exportieren/importieren	152
Programm-Code und SPS-Konfiguration rückübertragen	120
Programm-Code und Systemregister vergleichen	122
Programme	37
Programmierbeispiel	81, 95

Programmiereditoren	77
Programmierfenster	10
Programmiersymbole bearbeiten.....	83
Programm-Organisationseinheiten ...	35, 36
Programm-Organisationseinheiten (POE)	161
Projekt.....	161
Projekt auf die SPS übertragen	117
Projekt in die SPS übertragen	103
Projekt sichern (Backup).....	26
Projekt von der SPS öffnen	24
Projekt wiederherstellen (Restore)	26
Projekt zur SPS übertragen Einführung	104
Projekte bearbeiten.....	26
Projekte erstellen	24
Projekte exportieren/importieren.....	148
Projekte in Control FPWIN Pro	15
Projektnavigator	8, 161

Q

Querverweis.....	61
Querverweislisten	61

R

Reihenfolge beim Abarbeiten von Netzwerken in einer POE	123
Rezeptureditor	62
Rezeptur-Editor	62

S

Schaltflächen in der Symbolleiste.....	80
Schnittstellen-Parameter	113
Schritt hinein	135
Schritt über.....	135
SDT	161

SDT-Pool.....	161
Sicherheitseinstellungen	115
Sichern der Indexregister in Interrupt-Programmen	131
Speicherbelegung prüfen.....	111
SPS	17
SPS-Modus ändern.....	122
SPS-Typ	105
Start-Dialog	6
Statusleiste.....	12
ST-Befehle	87
STD (StandardDatenTyp)	22
ST-Editor	85
Structured Text Editor (ST)	85
Strukturierter Datentyp (SDT)	22
Strukturierter Text	85, 161
Systemregister einstellen.....	17
Systemvariablen.....	54

T

Task	161
Tasks.....	22, 63, 64
Tastaturbefehle	154
Tastaturbefehle für allgemeine Funktionen	154
Tastaturbelegung	153

U

Überflüssige globale Variablen im POE-Kopf löschen.....	48
Überwachung	141

V

VAR_CONSTANT	46
VAR_EXTERNAL.....	46
VAR_EXTERNAL_CONSTANT	46

VAR_EXTERNAL_RETAIN	46
VAR_GLOBAL	44
VAR_GLOBAL_EXTERNAL	44
VAR_GLOBAL_RETAIN	44
VAR_IN_OUT	46
VAR_INPUT	46
VAR_OUTPUT	46
VAR_OUTPUT_RETAIN.....	46
VAR_RETAIN	46
Variable	161
Variablen	43, 44
Variablen ändern.....	55
Variablen exportieren.....	56
Variablen im Programm zuweisen	52
Variablen importieren.....	58

W

Wichtige Symbole	ii
Woraus besteht ein Projekt in Control FPWIN Pro?	16

Z

Zugriff auf geschützte Objekte	29
Zusatzspeicher.....	143
Zweite Task für FP10SH/120K und FP2SH 120k.....	68

Änderungsverzeichnis

Handbuchnummer	Datum	Änderungen
ACGM0142DED V1.0	JUNE 2000	
ACGM0142DED V2.0	AUG. 2001	Aktualisierung für Release Control FPCWIN Pro 4
ACGM0142V3.0DED	Dez. 2003	Aktualisierung für Release Control FPCWIN Pro 5 Eine Übersicht aller Änderungen finden Sie in der Hilfe auf der Seite "Neu in dieser Version".
ACGM0142V4.0DED	Juli 2005	Update für Control FPCWIN Pro V5.1 und den neuen SPS-Typ FP-X. Neue Sicherheitseinstellungen. Eine Übersicht aller Änderungen finden Sie in der Hilfe auf der Seite "Neu in dieser Version".
ACGM0142V4.1DED	Januar 2006	Keine inhaltlichen Änderungen.
ACGM0142V4.2DE	Oktober 2008	Der erweiterte Dialog des Assistenten "Neues Projekt" wurde entfernt Zusätzliche Informationen zu Systemvariablen und Tastaturbefehle (siehe Seite 154)

Nordamerika Europa Asien-Pazifik China Japan

Panasonic Electric Works Niederlassungen

Europa		
► Headquarters	Panasonic Electric Works Europe AG	Rudolf-Diesel-Ring 2, 83607 Holzkirchen, Tel. +49(0)8024648-0, Fax +49(0)8024648-111, www.panasonic-electric-works.com
► Benelux	Panasonic Electric Works Sales Western Europe B.V.	De Rijn 4, (Postbus 211), 5684 PJ Best, (5680 AE Best), Netherlands, Tel. +31(0)499372727, Fax +31(0)499372185, www.panasonic-electric-works.nl
► Deutschland	Panasonic Electric Works Deutschland GmbH	Rudolf-Diesel-Ring 2, 83607 Holzkirchen, Tel. +49(0)8024648-0, Fax +49(0)8024648-555, www.panasonic-electric-works.de
► England	Panasonic Electric Works UK Ltd.	Sunrise Parkway, Linford Wood, Milton Keynes, MK14 6LF, Tel. +44(0) 1908 231555, +44(0) 1908 231599, www.panasonic-electric-works.co.uk
► Frankreich	Panasonic Electric Works Sales Western Europe B.V.	French Branch Office, B.P. 44, 91371 Verrières le Buisson CEDEX, Tél. +33(0)16013 5757, Fax +33(0)1 6013 5758, www.panasonic-electric-works.fr
► Irland	Panasonic Electric Works UK Ltd.	Dublin, Tel. +353(0)14600969, Fax +353(0)14601131, www.panasonic-electric-works.co.uk
► Italien	Panasonic Electric Works Italia s.r.l.	Via del Commercio 3-5 (Z.I. Ferlina), 37012 Bussolengo (VR), Tel. +390456752711, Fax +390456700444, www.panasonic-electric-works.it
	PEW Building Materials Europe s.r.l.	Piazza della Repubblica 24, 20154 Milano (MI), Tel. +39022900-5391, Fax +39022900-3466, www.panasonic-building-materials.com
► Nordische Länder	Panasonic Electric Works Nordic AB	Sjöängsvägen 10, 19272 Sollentuna, Sweden, Tel. +46859476680, Fax +46859476690, www.panasonic-electric-works.se
► Österreich	PEW Fire&Security Technology Europe AB Panasonic Electric Works Austria GmbH	Jungmansgatan 12, SE-211 19 Malmö, Tel. +46406977000, Fax +46406977099, www.panasonic-fire-security.com vertreten durch PEWDE, Josef Madersperger Str. 2, 2362 Biedermannsdorf, Tel. +43(0)223626846, Fax +43(0)223646133, www.panasonic-electric-works.at
► Polen	PEW Electronic Materials Europe GmbH Panasonic Electric Works Polska sp. z o.o.	Ennshafenstraße 9, 4470 Enns, Tel. +43(0)7223883, Fax +43(0)722388333, www.panasonic-electronic-materials.com Przedstawicielstwo w Polsce, Al. Krakowska 4/6, 02-284 Warszawa, Tel. +48 223381100, Fax +48 223381200, www.panasonic-electric-works.pl
► Portugal	Panasonic Electric Works España S.A.	Portuguese Branch Office, Avda Adelino Amaro da Costa 728 R/C J, 2750-277 Cascais, Tel. +351 214812520, Fax +351 214812529
► Schweiz	Panasonic Electric Works Schweiz AG	Grundstrasse 8, 6343 Rotkreuz, Tel. +41(0)417997050, Fax +41(0)417997055, www.panasonic-electric-works.ch
► Spanien	Panasonic Electric Works España S.A.	Barajas Park, San Severo 20, 28042 Madrid, Tel. +34 91 3293875, Fax +34 91 3292976, www.panasonic-electric-works.es
► Tschechien	Panasonic Electric Works Czech s.r.o.	Prumtyšlová 1, 34815 Planá, Tel. (+420-)374799990, Fax (+420-)374799999, www.panasonic-electric-works.cz
► Ungarn	Panasonic Electric Works Europe AG	Magyarországi Közvetlen Kereskedelmi Képviselet, 1117 Budapest, Neumann János u. 1., Tel. +36(0)14829258, Fax +36(0)14829259, www.panasonic-electric-works.hu
Nord- und Südamerika		
► USA	PEW Corporation of America	629 Central Avenue, New Providence, N.J. 07974, Tel. +1-908-464-3550, Fax +1-908-464-8513, www.pewa.panasonic.com
Asien/China/Japan		
► China	Panasonic Electric Works (China) Co., Ltd.	2013, Beijing Fortune, Building No. 5, Dong San Huan Bei Lu, Chaoyang District, Beijing Tel. +86-10-6590-8646, Fax +86-10-6590-8647, www.pewc.panasonic.cn
► Hong Kong	Panasonic Electric Works (Hong Kong) Co., Ltd.	RM1205-9, 12/F, Tower 2, The Gateway, 25 Canton Road, Tsimshatsui, Kowloon, Hong Kong, Tel. +852 2956-3118, Fax +852 2956-0398
► Japan	Panasonic Electric Works Co., Ltd.	1048 Kadoma, Kadoma-shi, Osaka 571-8686, Japan, Tel. +81-6-6908-1050, Fax +81-6-6908-5781 http://panasonic-electric-works.net
► Singapore	Panasonic Electric Works Asia Pacific Pte. Ltd.	101 Thomson Road, #25-03/05, United Square, Singapore 307591, Tel. +65-6255-5473, Fax +65-6253-5689