

SPaS

Prof. A. Pretschner
Dr. J. Alder

SPaS: die Softwareprojektierungsumgebung für automatische Steuerungen

von Prof. A. Pretschner und Dr. J. Alder

Inhaltsverzeichnis

Hinweis zur Nutzung des Handbuch	xiii
I. Handbuch	1
1. Einleitung	3
1.1. Übersicht zu SPAS	3
1.2. Wichtige Empfehlungen	5
2. Anwendbare Sprachen	7
2.1. Sprachen in SPAS	7
2.2. STEP 5	7
2.3. STEP 7 MICRO (S7-200)	7
2.4. IL nach DIN IEC 61131-3	7
2.5. Sprachen C und C++	7
2.6. STEP 7	8
2.6.1. STEP 7 (S7-300/400)	8
2.6.2. STEP7-Projekt	8
2.6.3. Die Organisationsbausteine (OB)	9
2.6.4. Anlaufbausteine OB 100, 101 und 102	9
2.6.5. Verschiedene Funktionen	10
2.6.6. Import des Compilates in STEP 7	11
2.6.7. Der lebendige PRAP - OnlineViewer	11
3. Entwurfs-Prozeß oder: Wie ist SPAS zu nutzen?	13
3.1. Entwurfsprozeß	13
3.2. Binäre Prozeßanalyse	14
3.3. Informationsfluß-Modell	14
4. Prozeß-Zerlegung und Funktionsblock-Darstellung	17
4.1. Prozeßzerlegung	17
4.2. Prozeß-Zerlegungs-Graph (PZG)	17
4.3. Kriterien der Prozeßzerlegung	18
4.4. Schritte der Prozeßzerlegung	18
4.5. Ziele der Prozeß-Zerlegung	19
4.6. Koppelungen zwischen Teilprozessen - Interactions -	20
4.7. Probleme und Fehler bei TP-Koppelungen	23
4.8. Funktions-Block-Darstellung (FBD)	23
5. Im PRAP eingebettete, nichtbinäre Funktionen	25
5.1. Nichtbinäre Funktionen	25
5.2. Zeitglieder - Timer	26
5.3. Zählglieder - Counter - Grundsätzliches und Projektierung	29
5.4. Vergleichsfunktion - Comparison	34
5.5. Flanken-Funktionen - Edge Detection	36
5.6. Kopierfunktion - Copy	37
5.7. Arithmetik-Funktionen Caculate(multiple) und Calculate(single)	37
6. Datenverwaltung	39
6.1. Die Interbase-Datenbank	39
6.2. Entwurf strukturierter Datentypen - User Defined Types (UDT)	39
6.3. Instanzieren strukturierter Datentypen	40
6.4. Variable	41
6.5. Benutzen der Variablen-Anzeige und -Liste	42
6.6. Alle Variablen anzeigen und bearbeiten	43
6.7. Datenbank-Report	43
6.8. Import einer Variablentabelle (Exceltabelle)	44
6.9. Back-Up-Datenbank	44

7. Prozeß-Ablauf-Netz (PAN)	45
7.1. Die Grundlagen	45
8. Prozeßzerlegung - Editor	47
8.1. Editor zur Entwicklung des Prozeßzerlegungsgraphen (PZG-Editor)	47
8.2. PZG zum Lehrbeispiel	47
8.3. Variablenlisten nach der Prozeßzerlegung	48
9. Projekt-Editor	51
9.1. Projekt-Editor	51
9.2. Neues Projekt anlegen	51
9.3. Öffnen, Schließen und Löschen eines Projektes	53
9.4. Projekt weiter bearbeiten	53
9.5. Konfiguration einfügen	55
9.6. Ressource einfügen	55
9.7. TASK einfügen/bearbeiten	55
9.8. PRAP-Typen einfügen/bearbeiten	56
9.9. Funktion in eine Ressource einfügen	58
9.10. PROGRAM einfügen/bearbeiten	58
9.11. PRAP INSTANCE einfügen/bearbeiten	59
9.12. PROGRAM INSTANCE einfügen/bearbeiten	60
9.13. Funktions-Aufruf	60
9.14. Variablenliste importieren	60
10. PRAP-Editor	61
10.1. PRAP-Editor	61
10.2. Allgemeine Theorie des PRAP	62
10.3. Der PRAP - Ein Graph	63
10.4. Grundsätze zum Gestalten eines PRAP	63
10.5. Prüfungen des PRAP	64
10.6. Export und Import von PRAP	66
10.7. Alle PRAP zum Beispielprojekt	66
10.7.1. Entwicklung eines PRAP (zum Projekt Lehrbeispiel)	66
11. Software-Produkt-Beschreibung	69
11.1. Precompiler	69
11.2. STEP-7-AWL	70
11.3. C-Programm	70
11.4. Programm in C++	70
12. Beispielprojekte	73
12.1. Lampen- und Stern-Dreieck-Schaltung	73
12.1.1. Vorbemerkungen zum Beispielprojekt	73
12.1.2. Anlegen des Projektes, Configuration und Ressource	73
12.1.3. Typ-Programm mit Instanzen der Typ-PRAP	79
12.1.4. Variable in das Programm eintragen	81
12.1.5. Excel-Variablenliste zum Beispielprojekt	82
12.1.6. Variablenliste der Instanzen mit direkten Adressen	83
12.1.7. Die Instanzen - Realisierung der Teilprozesse	84
12.1.8. Einfügen der Task in das Projekt	87
12.1.9. Die Programm-Instanz im Projekt	87
12.2. Stern-Dreieck-Aufschaltung	88
12.2.1. Stern-Dreieck-Anlaufschaltung (Einführung)	88
12.2.2. PRAP Stern-Dreieck-Steuerung (Teil 1)	90
12.2.3. PRAP Stern-Dreieck-Steuerung (Teil 2)	92
12.2.4. PRAP Stern-Dreieck-Steuerung (Teil 3)	94
12.2.5. Projektieren des Zeitgliedes "Wartezeit"	98
12.2.6. Motorstrom-Verarbeitung (Projektierung der Vergleichsfunktion)	102
12.3. Lampenschaltung	105
12.3.1. Lampenschaltung	105

12.3.2. PRAP zur Lampensteuerung (Teil 1)	106
12.3.3. PRAP zur Lampensteuerung (Teil 2)	110
12.3.4. PRAP zur Lampensteuerung (Teil 3)	113
12.3.5. PRAP zur Lampensteuerung (Teil 4)	114
12.3.6. PRAP zur Lampensteuerung (Teil 5)	115
12.4. Anwendungsbeispiele	116
12.4.1. Beispielprojekt	116
12.4.2. Beispiel Timer	116
12.4.3. Zähler-Beispiel	116
12.4.4. Beispiel Vergleicher	119
II. Lexikon	121
13. Begriffe in SPaS	123
13.1. Begriffs-Erklärungen	123
13.2. Adresse - allgemein	123
13.3. Adresse - direkt	123
13.4. Aktual-Parameter	124
13.5. Konstante Größen	124
13.6. Formal-Parameter	125
13.7. Aufruf	125
13.8. Aufruf-Bezeichner	125
13.9. Aufrufgruppe	125
13.10. Baustein	125
13.11. Bezeichner und Namen	126
13.12. Typ-Baustein / Typ-PRAP	126
13.13. Instanz-Baustein	126
13.14. PRAP- und Fremdprogramme (ACSII-Bausteine)	126
13.15. Project (Projekt)	127
13.16. Standard-Funktionen und -Funktionsbausteine	127
14. Wörterbuch	129
14.1. Alphabet-Buttons	129
14.2. -A-	129
14.3. -B-	129
14.4. -C-	129
14.5. -D-	130
14.6. -E-	130
14.7. -F-	130
14.8. -G-	130
14.9. -H-	130
14.10. -I-	130
14.11. -J-	131
14.12. -K-	131
14.13. -L-	131
14.14. -M-	131
14.15. -N-	131
14.16. -O-	131
14.17. -P-	131
14.18. -Q-	132
14.19. -R-	132
14.20. -S-	132
14.21. -T-	132
14.22. -U-	132
14.23. -V-	132
14.24. -W-	133
14.25. -X-	133
14.26. -Y-	133

14.27. -Z-	133
III. Anhang	135
A. Erläuterungen	137
A.1.	137
A.2.	137
A.3.	137
A.4.	137
A.5.	137
A.6.	137
A.7.	138
A.8. All Variables - Anzeige aller im Projekt deklarierten Variablen	138
A.9.	138
A.10.	138
A.11.	138
A.12.	139
A.13.	139
A.14.	139
A.15.	140
A.16.	141
A.17.	142
A.18.	143
A.19.	144
A.20.	144
A.21.	144
A.22.	144
A.23.	144
A.24.	144
A.25.	145
A.26.	145
A.27.	145
A.28.	145
A.29. Zählglied-Ausgang / Counter Output Q	146
A.30.	146
A.31.	146
A.32. Zähler auf-/abwärts zählen - Count Up/Down	147
A.33.	147
A.34.	147
A.35.	147
A.36. Datenaufzeichnung und Rückverfolgung	147
A.37.	148
A.38.	148
A.39.	149
A.40.	150
A.41.	150
A.42.	150
A.43.	150
A.44.	150
A.45.	150
A.46.	150
A.47.	151
A.48.	152
A.49.	152
A.50.	152
A.51.	152
A.52.	152

A.53.	152
A.54.	152
A.55.	152
A.56.	152
A.57.	153
A.58.	153
A.59.	153
A.60.	153
A.61.	153
A.62.	153
A.63.	154
A.64.	154
A.65.	154
A.66.	154
A.67.	155
A.68. Deklaration der TASK, VAR_GLOBAL und VAR_EXTERNAL in RESOURCE	155
A.69.	156
A.70.	156
A.71.	157
A.72.	158
A.73.	158
A.74.	159
A.75.	159
A.76.	159
A.77.	160
A.78.	160
A.79.	160
A.80. Bemerkungen zum Handbuch zu SPAS	160
A.81.	160
A.82.	161
A.83.	163
A.84.	163
A.85. SPAS Manager	164
A.86. Deklaration VAR_GLOBAL und VAR_ACCESS in CONFIGURATION	165
A.87.	165
A.88.	166
A.89.	166
A.90. Inbetriebnahme des "Lebendigen PRAP"	166
A.91.	166
A.92.	166
A.93.	166
A.94.	166
A.95. Einführung - Introduction	167
A.96.	168
A.97.	168
A.98.	170
A.99.	171
A.100.	172
A.101.	172
A.102.	172
A.103.	172
A.104.	172
A.105.	172
A.106.	172
A.107.	173

A.108.	173
A.109.	173
A.110.	174
A.111.	174
A.112.	174
A.113. Operation Variables	175
A.114. Operation	175
A.115.	175
A.116.	175
A.117.	175
A.118.	176
A.119.	176
A.120.	176
A.121.	176
A.122. Information und Beschreibung zur Funktion des PRAP	176
A.123.	176
A.124.	177
A.125.	177
A.126.	177
A.127.	177
A.128.	177
A.129.	177
A.130.	177
A.131.	177
A.132.	178
A.133.	178
A.134.	178
A.135. Process Variables	178
A.136.	178
A.137.	178
A.138.	178
A.139.	179
A.140.	179
A.141.	179
A.142. Project Member Actual Function, Call Function	179
A.143. Project Member - Configuration	179
A.144. Project Member PRAP-Instance	180
A.145. Project Member PRAP-Type	180
A.146. Project Member - Ressource	181
A.147. Project Member - Task	182
A.148.	182
A.149.	182
A.150.	182
A.151.	183
A.152.	183
A.153.	183
A.154.	183
A.155.	183
A.156.	184
A.157.	184
A.158.	184
A.159.	184
A.160.	184
A.161.	184
A.162.	185

A.163.	185
A.164.	185
A.165.	186
A.166.	186
A.167.	186
A.168.	186
A.169.	186
A.170.	187
A.171.	187
A.172.	187
A.173.	187
A.174.	187
A.175.	187
A.176.	187
A.177.	187
A.178.	188
A.179. Hinweis zu Umgehung der Peripherieadressen bei STEP 7	188
A.180.	188
A.181.	188
A.182.	189
A.183.	189
A.184.	189
A.185.	190
A.186.	190
A.187.	190
A.188. Funktions-Test eines PRAP - Offline Test	190
A.189. Zeitglied-Ausgang Q / Timer Output Q	190
A.190.	191
A.191.	191
A.192.	191
A.193.	192
A.194. Typen-PRAP: Eingabe und Bearbeitung der Variablen	192
A.195.	193
A.196. Strukturierte Datentypen	193
A.197.	193
A.198.	193
A.199.	193
A.200.	194
A.201.	194
A.202. Variable der Vergleich-, Kopier- und Arithmetik-Funktionen	194
A.203. Variables - Anzeige geordnet nach P-, O-, sonstige Variable	195
A.204.	195
A.205.	196
A.206.	196
A.207.	197
A.208.	197
A.209.	197
A.210.	197
A.211.	197
A.212. Project Member Function	198
A.213. Import einer Excel-Tabelle mit Teilprozeß-Variablen	198
A.214. Excel-Liste, nach Teilprozessen geordnet	201
A.215. Excel-Liste, nach Adressen geordnet	202
A.216. Task-gesteuerter Timer	202
A.217. Zeitglied-Projektierung - Timer Configuration	203

A.218. Vergleichsfunktionen - Comparison Configuration	204
A.219. Zählglied-Projektierung - Counter Configuration	204
A.220. Flankenauswertung - Raising and Falling Edges Detection	205
A.221. Kopier- und arithmetische Funktionen - Calculate(multiple) und Calculate(single)	205
A.222. Database Options	206
A.223. Instanz strukturierter Datentypen	209
A.224. Import von Typen-PRAP und anderen	210
A.225. Export von Typen-PRAP	210
A.226. Wahl der Sprache, Erläuterungen zur Projektgestaltung	210
A.227. Project Member - PROGRAM	211
A.228. Instanzbildung: Zuordnung Formal- zu Aktualparameter	212
A.229. Project Member - PROGRAM-Instance	212
A.230. Deklaration der Aktual-Parameter und Adressen in PROGRAM-TYPE	213
A.231. PRAP Drucken	214
A.232. Probleme beim Kanten zeichnen	214
A.233. Projektierung der Vergleichsfunktion	215
A.234. Kanten zeichnen und Probleme	217
A.235. Mehrfach-Verwendung von Prozeß-Variablen	218

Hinweis zur Nutzung des Handbuch

Die Texte des Handbuchs werden aus xml-Dateien generiert. Diese Dateien wurden als Direkthilfe für die Anwendung von SPaS erstellt und für dieses Dokument nach Docbook transformiert. Aus diesem Grund werden Sie gelegentlich auf Informationen stoßen welche Ihnen unverständlich erscheinen werden. Hierbei handelt es sich um Hinweise welche die Anwendung der SPaS-Hilfe betreffen.

Die im Anhang befindlichen Abschnitte sind Anmerkungen zum Handbuch. Sie erreichen diese durch Verweise in den jeweiligen Kapiteln.

I Handbuch

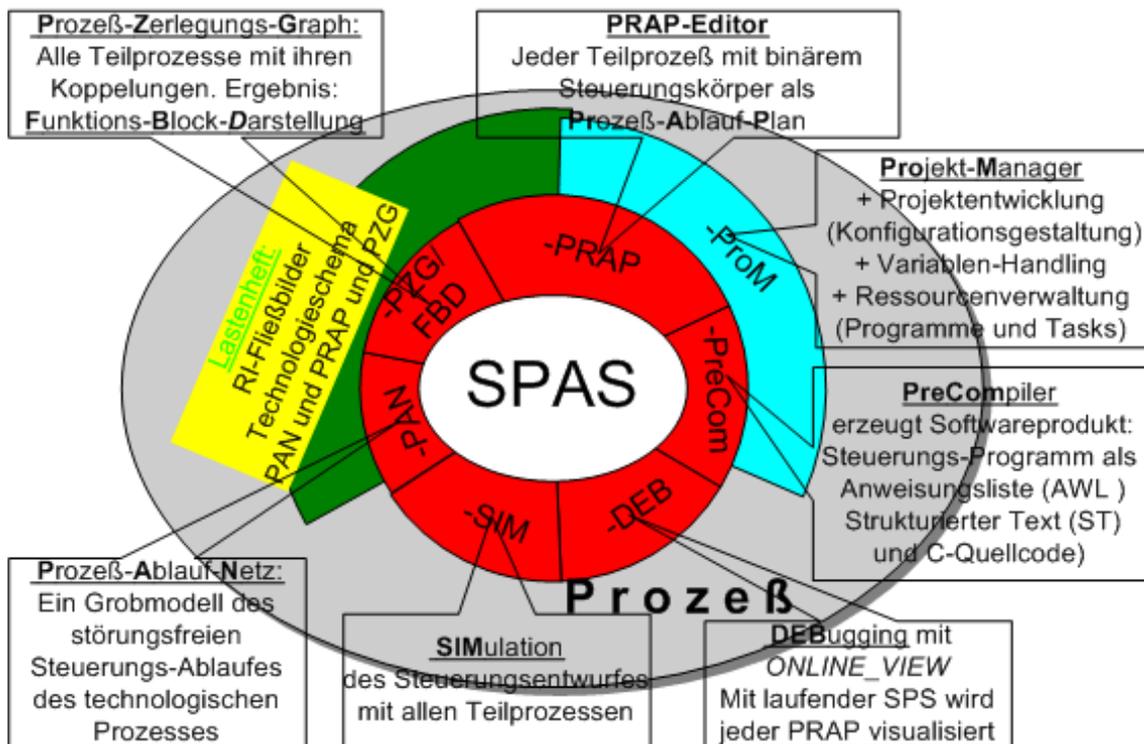
Kapitel 1

Einleitung

1.1 Übersicht zu SPAS

Für den eiligen Nutzer:

- Wichtige Hinweise zu verschiedenen Hilfe-Möglichkeiten wie ? und F1-Taste [siehe Seite 155]
- Wichtige Empfehlungen zur Anwendung von SPAS [siehe Seite 5]
- Anlegen eines neuen Projektes [siehe Seite 51]
- Öffnen, Schließen und Löschen eines Projektes [siehe Seite 53]
- Begriffe in SPAS [siehe Seite 123]
- Wählbare Sprachen [siehe Seite 7]



Nutzbare Komponenten: PRAP-Editor [siehe Seite 61] Projekt-Manager [siehe Seite 51] Datenverwaltung [siehe Seite 39] PreCompiler [siehe Seite 69]

S P A S - Software-Projektierungswerkzeug für Automatische Steuerungen zur Erzeugung von Steuerungs-Software für SPS, Microcontroller und Soft-SPS

SPAS richtet sich terminologisch und systematisch nach der Norm DIN IEC 61131-3, wodurch aus konventioneller, bislang üblicher Sicht sich zuweilen andere Vorgehensweisen (siehe Entwurfsprozeß [siehe Seite 13]) ergeben. Die Norm wird hin und wieder zitiert, aber nicht erläutert, höchstens interpretiert. Ferner erfolgt die Prozeß-Zerlegung mit dem Ziel, den Vorstellungen der IEC 61499-1 möglichst nahe zu kommen. Deshalb gibt es in Erweiterung der Prozeß-Zerlegung die Funktions-Block-Darstellung als hilfreiche Unterstützung der Projektierung. SPAS verwaltet in seinem Manager derzeit folgende nutzbare Komponenten:

- PRAP-Editor [siehe Seite 61]
 - Hiermit ist der technologisch orientierte Ablauf jeder Teilsteuerung zu entwerfen.
 - Eine schrittweise Anleitung an Beispielen ist im Editor enthalten.
 - Der Precompiler erzeugt für jeden im PRAP entworfen Steuerungsablauf einen Funktionsbaustein, kurz PRAP-FB, als FB-Type.
- Funktionen-Editor [siehe Seite 198]
 - Eine Funktion [siehe Seite 154] enthält Berechnungen, deren Ergebnisse in jedem Bearbeitungs-Zyklus eines Programmes für darin enthaltene Teilsteuerungen neu zur Verfügung stehen.
 - Im betreffenden Programm erfolgt der Funktionsaufruf mit Zuweisung der aktuellen Parameter.
- Projekt-Editor [siehe Seite 51]
 - Die Struktur des gesamten Steuerungs-Projektes ist damit zu entwerfen. Es besteht u.a. aus allen PRAP-FB und solchen, die nicht mit dem PRAP-Editor entworfen wurden (z.B. Regelungs-FB) und hier generell Fremd-FB genannt werden, die als ASCII-Bausteine vorliegen müssen.
 - Das Projekt kann aus mehreren Konfigurationen bestehen
 - Jede Konfiguration enthält mindestens eine Ressource
 - Jede Ressource enthält wenigstens eine Task, ein Typ-Programm und eine Programm-Instanz.
 - Ein Beispielprojekt ist im Projekt-Editor enthalten.
- Pre-Compiler [siehe Seite 69]
 - Die Kurzform heißt PreCOM.
 - Jede Ressource ist in eine der Wählbaren Sprachen [siehe Seite 7] in Gestalt einer Anweisungsliste im Textformat übersetzbar. Die konventionelle Sprache ist derzeit STEP 7-AWL. Auch in C oder C++ kann PreCom übersetzen. Die vorgesehenen Sprachen gemäß DIN IEC 61131-3 sind derzeit IL oder ST. Im Base Level der Hersteller sind diese nicht importierbar.
- Datenverwaltung in einer Interbase-Datenbank [siehe Seite 39]
 - Die mit dem PRAP-Editor entworfene funktionelle Variablen-Verarbeitung für jeden Teilprozeß wird in der Datenbank gehalten
 - Die Variablen einer Ressource(z.B. SPS) oder eines Programms stehen zentral in der Datenbank zur Verfügung
 - Die mit dem Projekt-Editor entworfene Automatisierungs-Programmstruktur ist in der Datenbank gespeichert

- Mit Hilfe dieses Tools ist es möglich, den laufenden Steuerungsprozeß zu verfolgen. Die Darstellung ist dabei der PRAP, der im Entwurfsablauf vom Teilprozeß (Instanz) genutzt wurde, also im Projekt enthalten ist.
Es ist auch möglich, einen PRAP statisch schrittweise ohne SPS zu untersuchen.
- Precompiler - PreCOM [siehe Seite 69]
Ein Steuerprogramm, von PreCOM erzeugt, ist ein Quellcode-Programm im Text-Format als Anweisungsliste für STEP 7, oder C- bzw. C++-Quellcode.
Die AWL muß folglich noch in den Maschinencode des betreffenden CPU übersetzt werden.
Eine obere Grenze der Anzahl der zu implementierenden PRAP- bzw. Fremd-Programme besteht im Prinzip nicht.

1.2 Wichtige Empfehlungen

Was Sie beherzigen sollten:

Sie haben sich vorgenommen, mit SPAS die Prozeß-Steuerungsprojekte zu bewältigen. Anstatt mit einer der üblichen SPS-Sprachen (Kontaktplan, Funktionsplan, Anweisungsliste oder Ablaufsprache) zu arbeiten, müssen Sie jetzt mit einer prozeßnahen Beschreibung, dem Prozeßablaufplan (PRAP), jeden Teilprozeß graphisch erfassen. Das bedeutet für Sie eine große Umstellung, weil Sie sich in erster Linie um die technologischen Zusammenhänge kümmern müssen, nicht wie bisher allein, sondern möglichst mit dem Prozeßtechnologe zusammen. Er bzw. Sie kann Maschinenkonstrukteur sein oder Verfahrenstechnologe, oder auch Verwaltungsspezialist. In jedem Fall muß es eine Persönlichkeit sein, die den zu automatisierenden Prozeßablauf in allen Einzelheiten kennt. Legen Sie auf alle Fälle jegliche Gedanken ab, wo ein Merker angelegt oder gesetzt werden soll. Machen Sie sich keine Gedanken mehr um logische Ausdrücke wie UND und ODER oder irgendetwas speichern zu müssen. Verlassen Sie sich darauf, daß das Compilat, meist eine Anweisungsliste, KEINE syntaktischen Fehler enthält, wenn Ihr Projekt keine Probleme beim Vorübersetzen macht. **ÄNDERN SIE NIEMALS** in dem Compilat, Sie lösen damit Ihr Problem nicht!! Suchen Sie die Problemlösung immer durch Korrekturen in den PRAP oder in der Datenbank.

Falls Sie mit dem Vorgehensmodell¹ in SPAS noch nicht vertraut sind, also welche Arbeits-Schritte der Reihe nach erfolgen sollten, dann finden Sie Hinweise im Thema Entwurfsprozeß. [siehe Seite 13] Die Datenverwaltung [siehe Seite 39] wird darin ebenfalls besprochen.

Wenn Sie eine Anleitung benötigen, wie ein Projekt angelegt wird, finden Sie im Projekt-Editor [siehe Seite 51] unter "Neues Projekt anlegen" allgemeine Hinweise und schrittweise Erläuterungen zu einem Musterprojekt "Lampen- und Stern-Dreieck-Steuerung."

Im PRAP-Editor [siehe Seite 61] finden Sie Ausführungen zum PRAP selbst, zur Bedienung des Editors, und die drei PRAP des Beispiel-Projektes sind ausführlich behandelt.

Dringender Hinweis vor jedem Beginn mit einem Projekt oder im Falle von Projekt-Veränderungen (Versionen):

Lesen Sie wenigstens den ersten Satz aus dem Thema im Buch Projekt-Editor: Neues Teil-Projekt anlegen [siehe Seite 51].

Hinweise zu Hilfe-Texten [siehe Seite 155]

Kapitel 2

Anwendbare Sprachen

2.1 Sprachen in SPAS

Unter dem Begriff Sprache wird normalerweise unterschieden, ob in Kontaktplan, Funktionsbaustein, Anweisungsliste usw. programmiert wird. Hier wird generell eine Anweisungsliste erzeugt, es sei denn, Structured Text(in Vorbereitung) gemäß IEC 1131-3, "C" oder "C++" wurden gewählt. Aber die Hersteller "sprechen keine einheitliche Sprache", das will heißen, daß sich die AWL von Hersteller zu Hersteller, ja sogar innerhalb der Produkte eines Herstellers(!) deutlich unterscheiden. Das ist der Grund, weshalb in SPAS der Begriff etwas anders aufgefaßt wird.

Sie können sich nun die Besonderheiten, soweit wir sie erwähnenswert gefunden haben, zu den wählbaren Sprachen ansehen:

STEP 5 (wird nicht unterstützt) [siehe Seite 7]

STEP 7 MICRO (S7-200)(wird derzeit nicht unterstützt) [siehe Seite 7]

STEP 7 (S7-300/400) [siehe Seite 8]

IL nach IEC 1131-3 / Concept (wird nicht unterstützt) [siehe Seite 7]

C / C++ [siehe Seite 7]

2.2 STEP 5

Diese Sprache wird nicht im Precompiler implementiert!

2.3 STEP 7 MICRO (S7-200)

Diese Sprache ist noch nicht im Precompiler implementiert!

Die Ressourcen-Unterschiede zwischen den CPU der 212 bis 214 bzw. der 221 bis 226 werden bezüglich verfügbarer Bereiche im Precompiler nicht berücksichtigt. Es wurde ein "kleinster gemeinsamer Nenner" bezüglich der Operationen eingesetzt, soweit das sinnvoll ist. Andererseits erfolgt das Umsetzen der Operationen ohne Rücksicht darauf, ob z.B. dies zwar in der 224, aber nicht in der 212 realisierbar ist. Sie als Programmierer müssen also entscheiden, welche CPU wofür geeignet ist.

TYPEN-TEILPROZESSE dürfen nur höchstens 16 Formalparameter enthalten. Dies gilt für die gesamte 200er Serie. Zu jeder Instanz eines Typen können Sie einen beliebigen Bezeichner eingeben, insgesamt 64 Instanzen sind erlaubt. Sie heißen hier Sub Routines "SBR[nr]", wobei [nr] von 0 bis 63 zulässig ist.

TIMER werden nur mit Einschaltverzögerung programmiert, also mit dem Standard-Funktionsblock-Typen (Data Type) TON, daher dürfen auch nur die so funktionierenden Timer-Nummern verwendet werden. Falls Sie eine Nummer außerhalb dieses Bereiches angeben, z.B. T0, dann erfolgt eine automatische Zuweisung, von der Sie durch Compiler-Mitteilung informiert werden.

Es gibt KEINE FORMALTIMER wie bei S7-300/400! Deshalb immer "Tnr" in das Feld ADDRESS eintragen und bei USAGE = ADDRESS MEMORY wählen.

VERGLEICHE sind immer in der Folge Istwert - Vergleichsfunktion - Sollwert zu schreiben.

2.4 IL nach DIN IEC 61131-3

IL wird nicht im Precompiler implementiert! Es ist nur ST vorgesehen.

2.5 Sprachen C und C++

C und C++ sind voll nutzbar.

Beachten Sie bitte, bei der Vergabe direkter Adressen, das Modulo-8-Verfahren wie bei Siemens für Bitadressen beizubehalten. ABER KEINE E;A ODER M vorstellen! Also z.B.: USAGE = Address Input und ADDRESS = 2.5

Sie benötigen den Quellcode für

- den azyklischen Programmaufruf mit Task-Priorität 0 oder 1 und
- Standard-Funktionen und -Funktionsblöcke wie TON, CTUD, R_EDGE, F_EDGE.

Fordern Sie diesen an bei
Herrn Prof. Pretschner,
Fachbereich Elektro- und Informationstechnik, HTWK Leipzig:
pretsch@fbeit.htwk-leipzig.de



Anmerkung

Es ist vorgesehen, sowohl zyklische Tasks als auch Interrupt-Tasks bereitzustellen. Ein Termin ist dafür nicht nennbar.

2.6 STEP 7

2.6.1 STEP 7 (S7-300/400)

Es werden nur Ressourcen-Unterschiede zwischen der 300er und der 400er Serie grob überprüft. Der Programmierer muß möglichst selbst an die Beschränkung der Ressourcen bei Anwendung der 300er Serie denken. Er wird aber auf Überschreitungen der Reihe 313 aufmerksam gemacht, jedoch nicht der CPU 312. Zum Beispiel gelten für die Merkerbereiche:

CPU 312: M 0.0 bis M 127.7

CPU 313 usw.: M 0.0 bis M 255.7

CPU 400: M 0.0 bis M 65535.7

- Nicht erfaßt werden in SPAS alle Bereichsgrenzen, die mit Herstellen des Maschinencodes zusammenhängen. Die OB1, DB, FB und FC können nur 16K bei den CPU 31x, aber 64K der 400-er Reihe aufnehmen.
- Falls der Multiinstanz-Baustein eines PROGRAMS "überläuft", müssen Sie ein weiteres Programm in das Projekt einfügen und dort hinein Teile des ersten verschieben.
- Die CPU 318 besitzt 4 Akkumulatoren und verarbeitet gegenüber anderen CPU, die nur 2 Akkumulatoren haben, Arithmetikbefehle anders.
- Bitte achten Sie darauf, daß alle Symbole für FB-Bezeichner oder deren Instanzen, sowie Variablennamen deutlich unter 25 Zeichen Länge bleiben.
- Peripherie-Adressen (PEW, PAW usw.), wie sie bei der Nutzung von Analog-Digital-Wandlern oder umgekehrt erforderlich sind, können im "lebendigen PRAP" (Online.exe) nicht angezeigt werden. Deshalb kann es sinnvoll sein, eine PEW z.B. auf eine MW zu kopieren.

Folgende Step-7-Themen sind erreichbar:

Step 7: Projekt [siehe Seite 8]

Organisationsbausteine [siehe Seite 9]

Wie der OB 100 eingebaut werden kann [siehe Seite 9]

Zeit- und Zählglieder, Schieben, Datentypwandlung [siehe Seite 10]

Das Ergebnis und wie es gespeichert wird [siehe Seite 11]

Inbetriebnahme mit OnlineViewer [siehe Seite 11]

2.6.2 STEP7-Projekt

Ein STEP7-Projekt beinhaltet immer ...

- wenigstens eine Ressource mit einer konkreten CPU, z.B. S7-315-2DP und wenigstens einer Hauptprogramm-Task, dem OB1, und einer weiteren Startprogramm-Task, dem OB100 oder auch zusätzlich OB101.
- wenigstens einen (Haupt-)Programm-Typ, der in einem FB eingetragen wird, dessen Nummer Sie bitte angeben müssen.
Dem Programm-Typ sind alle Instanzen nachzuordnen, denen Sie Datenbausteine zuordnen können oder keine.
- genau eine Programm-Instanz, die zunächst den (Haupt-)Programm-Typ zugeordnet erhält und darunter diejenige Task, die den OB1 enthält. Und es muß ein Datenbaustein angegeben werden, der Instanz-DB.
Er ist eine Multiinstanz, wenn die FB-Instanzen keine DB haben.



Anmerkung

Einwandfreies Nachladen geänderter Instanzen bei online laufender CPU (Schalterstellung auf "RUN-P") ist nur bei Einzelinstanzen möglich!

2.6.3 Die Organisationsbausteine (OB)

Step 7 stellt mit seinem Betriebssystem in der CPU verschiedene OB zur Verfügung, mit denen zeitzyklisch (OB 30 bis 37) oder prozess-spezifisch als Interrupt (OB40 bis 47) Programme aufgerufen werden können.

Für den azyklischen Betrieb steht der OB1 zur Verfügung, der die ihm zugewiesenen Programme aufruft, sobald sowohl die interne Datenverarbeitung als auch die ihm zugewiesenen Programme abgearbeitet sind.

Start-OB dienen der Initialisierung nach einer Unterbrechung des Betriebes. Sie werden vor allen anderen OB vom Betriebssystem gestartet und dauern nur einen CPU-Zyklus.



Anmerkung

Sehen Sie sich im STEP-7-Manager die Werte für die Priorität, die Zeitwerte [siehe Seite 137] der Intervalle und Adressvergabe für prozessabhängige Aufrufe an und wie mit den OB verfahren werden muß. Für OB 100, 101 und 102 [siehe Seite 9] gibt es noch ein Hilfsprogramm zu schreiben.

Die OB40 bis 47 sind NOCH NICHT IN SPAS BETRIEBSFÄHIG!!!

Eine Schnellinformation erhalten Sie hier: S7TaskRelations [siehe Seite 187]

2.6.4 Anlaufbausteine OB 100, 101 und 102

Für OB100 gilt in vorliegender Version 1 folgendes:

- Fügen Sie im Step7-Manager in die Bausteine Ihres Projektes den OB100 ein und programmieren Sie CALL FC0.
Natürlich können Sie eine beliebige FC-Nr. wählen.
- Editieren Sie mit einem einfachen Textprogramm, z.B. WordPad, den FC0 für eine
 - Hauptprogramm-Multiinstanz:

```

FUNCTION FC0 : VOID
BEGIN
SET;
= 'Multi-Instanz-Name'.Instanz-Name'.firstRUN;
= 'Multi-Instanz-Name'.nächster Instanz-Name'.firstRUN;
= ... alle Instanz-Namen'.firstRUN;
...
END_FUNCTION

```

- Hauptprogramm-Einzelinstanz:

```

FUNCTION FC0 : VOID
BEGIN
SET;
= 'Instanz-Name'.firstRUN;
= 'nächster Instanz-Name'.firstRUN;
= ... alle Instanz-Namen' .firstRUN;
...
END_FUNCTION

```

- Speichern Sie diese Datei unter 'Bezeichner'.AWL ab, damit Ihnen ein Import (Einfügen | Externe Quelle) möglich ist.
- Entweder:
Kopieren Sie den jeweils zur Hauptprogramm-Instanz passenden Programmtext in die vom Precompiler hergestellte 'Bezeichner'.AWL am Ende als letzten Eintrag. Danach dann übersetzen.
Oder:
Importieren Sie diese Datei in Step 7 zusätzlich zur Programmdatei. Dann aber immer als letzte übersetzen, damit der Step7-Compiler die gültigen Adressen einsetzen kann!

Die Bezeichner in '..' lesen Sie aus Ihrem Projekt ab. Alles andere einfach abschreiben.

2.6.5 Verschiedene Funktionen

Zeitglieder sind wie bei STEP 5 konzipiert. Im PRAP-TYP sind sie bei "Variable Usage" als INPUT zu deklarieren, der Timer ist dann als formale Größe bekannt mit seinem Namen. Die aktuellen Timer sind in den Instanzen zu übergeben.

Falls erforderlich, dürfen auch Timer direkt adressiert verwendet werden, sie sind dann in den Instanzen nicht mehr änderbar! Die aktuelle Größe, z.B. T12, ist als Adresse zu betrachten, "Variable Usage" muß dann als ADDRESS MEMORY deklariert werden.

Als Datentyp, eigentlich Standard-Funktionsbaustein-Typ, gilt grundsätzlich "TON" und darf nicht verändert werden!

Eingegebene Soll-Zeitwerte werden automatisch mit dem Einschalten des Timers geladen, sofern Sie einen konstanten Wert angeben. Der Datentyp ist immer "TIME" [siehe Seite 147].

Ist der Zeitwert irgendwie veränderlich und daher über Instanzierung oder Speicherplatz für den Timer als Typfunktion verfügbar, müssen Sie eine Funktion FC40 aus der Step7-Standard-Bibliothek in das Step7-Projekt² einsetzen.

Der Timer-Ausgang Q wechselt seinen Wert bei Erreichen des Sollwertes. Falls der Ausgang nicht benötigt wird, braucht er nicht projiziert zu werden.

Zählglieder sind vorwärts oder rückwärts zählend oder für beides im PRAP-TYP projektierbar. Es können nur positive Werte von 0 bis 999 gezählt werden. Geben Sie für Zählwerte als Datentyp "INT" an. Der Zähler-Ausgang Q = 0/1 hängt vom inneren Zählerwert ab. Es ist daher nur möglich, rückwärts gezählte Ereignisse über Q zu erfassen: Sobald der Zähler-Inhalt Null ist, wechselt der Q-Wert. Reine Vorwärtszähler geben über Q folglich nicht den erreichten Sollwert an!

Natürlich ist die Sollzahl mit dem Einschalten des Zählers zu laden (wird automatisch erledigt).

Zählglieder (formale Zähler) bei "Variable Usage" mit "INPUT" deklarieren. In den Instanzen sind die aktuellen Zähler zuzuweisen. Festgelegte Counter, z.B. Z12, sind als Adresse zu betrachten und in das Feld ADDRESS einzutragen, und "VARIABLE USAGE" ist ADDRESS MEMORY.

Der "DATA TYPE" = CTUD gilt einheitlich. Der Precompiler ermittelt, ob nur vorwärts oder nur rückwärts gezählt wird und schreibt automatisch den richtigen Typ in das Compilat.

Datentyp-Wandel ist wie folgt [siehe Seite 148] möglich:

BCD<-->INTBCD<-->DINT

BCD<-->WORDBCD<-->DWORD

SINT-->INT-->DINT-->REAL

REAL-->DINT mit RND (round) oder TRUNC

RND wird automatisch erzeugt oder durch REAL_TO DINT(variable name),

TRUNC(variable name) erfolgt nicht automatisch!

Bit-Schiebe-Operationen

benötigen zwei Daten, die Variable - den Operanden - und die Anzahl der zu schiebenden Bit.

Der Operand darf WORD oder INT bzw. DWORD oder DINT als Datentyp haben, die Anzahl BYTE oder SINT. Nur eine positive Anzahl ist zulässig. Denken Sie auch daran, daß die Anzahl höchstens 16 bzw. 32 sein darf.

2.6.6 Import des Compilates in STEP 7

Das Ergebnis des Compilierens - das Compilat [siehe Seite 70] - sehen Sie nach Klick auf "Show Compilat" im Projekt-Menue. [siehe Seite 161] Ein Klick auf "Save to file" erzeugt drei Dateien, die als Name die Identifikation der markierten Ressource erhalten:

1. Die Anweisungsliste mit Extension "[Ressource].AWL" und
2. Die Symboltabelle mit Extension "[Ressource].SDF"
3. Das Online-Programm mit Extension "[Ressource].ONL"

Import von AWL und Symboltabelle in den Simatic-Manager für STEP 7

Der SIMATIC-Manger gewährleistet den Import der AWL- und SDF-Dateien, so daß Sie keine Probleme beim Implementieren des Kompilates in eine S7 haben. Die AWL können Sie in den Ordner "Quellen" eines S7-Programms(..) einfügen. Markieren Sie S7-Programm: Klick auf "Einfügen" und "Externe Quelle"

Die sdf - Symboldatei - gehört zum S7-Programm(..) wie die AWL und muß vor dem Compilieren der AWL importiert werden. Markieren Sie wieder S7-Programm(..): Klick auf "Einfügen" und "Symboltabelle"



Anmerkung

Falls Sie Zeitwerte in Instanzen einem Timer übergeben, beachten Sie bitte, den Eintrag "TIM_S5TI" zusätzlich in die Symboltabelle selbst durchzuführen, wobei die Adresse "FC40" lautet.

2.6.7 Der lebendige PRAP - OnlineViewer

Das "**Online-Programm**" dient zur Darstellung des "lebendigen PRAP", d.h. mit dem Arbeiten der SPS kann über deren MPI-Verbindung am Bildschirm der aktuelle Stand des Teilprozesses angezeigt werden. Leider ist die Datenübertragung zur Zeit nur für langsame Steuerungsprozesse geeignet. An einer Profibus-Übertragung der Daten wird gearbeitet.

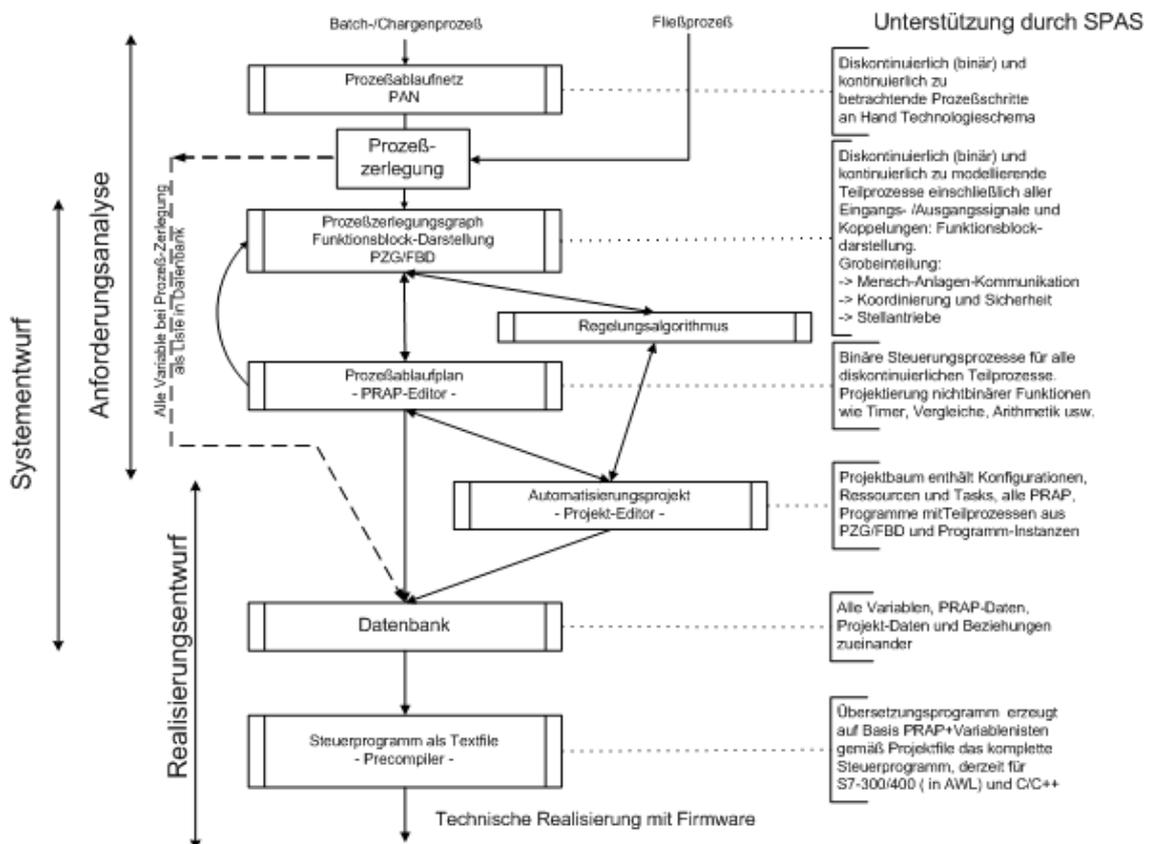
Kapitel 3

Entwurfs-Prozeß oder: Wie ist SPAS zu nutzen?

3.1 Entwurfsprozeß

Sie werden mit dem Vorgehensmodell in SPAS bekannt gemacht und hoffentlich auch bald vertraut sein. Bitte lesen Sie die aufgeführten Themen, damit das nachfolgende Schema verständlich wird:

- Grundlagen und Zusammenhänge: Binäre Prozeßanalyse [siehe Seite 14]
- Ein nützliches Hilfswerkzeug für den Anfang: Prozeß-Ablauf-Netz [siehe Seite 45]
- Eine notwendige Vorarbeit zum Automatisierungsprojekt: Prozeßzerlegung [siehe Seite 17]
- Schlichtweg DAS BESCHREIBUNGSMITTEL für jeden (Teil-)Steuerungsprozeß: Prozeß-Ablauf-Plan [siehe Seite 61]
- Das Variablen-Handling, die Datenverwaltung: Daten-Bank, Datenzugriff, Variablen-tabelle [siehe Seite 39]



Danach oder dabei ist wichtig zu wissen, wie der Prozeß und die zugehörige Steuerung betrachtet werden, dies erfahren Sie durch das Informationsflußmodell [siehe Seite 14].

Das Prozeß-Steuerungsprojekt schließt alle Teilprozesse [siehe Seite 191] ein und beschreibt die "Organisation" des Steuerungsablaufes. Dazu gehören Entscheidungen, welche Teilprozesse in welche

Steuereinheit (z.B. SPS) zu implementieren sind und in welcher Reihenfolge sie aufgerufen werden sollen. Dies alles enthält ein Projektbaum, der mit dem Projekt-Editor [siehe Seite 51] gestaltet wird. Dies ist das "Automatisierungs-Projekt"-Modell. Im Projekt-Editor finden Sie auch (ein Beispiel-Projekt [siehe Seite ?]).

3.2 Binäre Prozeßanalyse

Die Erarbeitung einer Automatisierungskonzeption ist ohne eine fehlerfrei bzw. fehlerarm und systematisch formulierte Aufgabenstellung nicht sinnvoll. Die Binäre Prozeßanalyse stellt drei graphische Beschreibungsmittel zur Verfügung, das Prozeßablaufnetz [siehe Seite 45], den Prozeßzerlegungsgraphen [siehe Seite 17] und den Prozeßablaufplan [siehe Seite 61].

Unter der Methode Binäre Prozeßanalyse ist zu verstehen, daß die Steuerung eines technologischen Prozesses, einschließlich regelungstechnischer Komponenten zum Stabilisieren von Prozeßgrößen, in sogenannten Prozeß-Schritten erfolgt. Jeder solcher Prozeßschritt ist ein technologisch begründeter Teil des zu steuernden Prozesses - besser: Prozeß-Abschnittes, der je nach Prozeßablauf durchzuführen ist oder nicht.

Diese Zweiwertigkeit der Entscheidung deutet auf ein binäres Steuerungsverhalten hin, das im Ergebnis der Analyse der steuerungstechnischen Anforderungen zu einem dementsprechenden Modell führt, dem Prozeßablaufnetz (PAN).

Enthalten Prozeßschritte Stabilisierungs-Anforderungen für Prozeßgrößen, muß dafür die Prozeßanalyse der Regelungsstrecke(n) erfolgen, die nur im Falle der Anwendung der strukturellen Analyse (siehe Fußnote) ebenfalls binären Charakter hat, ansonsten sind selbstverständlich analoge, numerisch untermauerte Analysemethoden einzusetzen.

Enthalten Prozeßschritte binäre Steuerungs-Anforderungen, dann muß zwingend die binäre Prozeßanalyse fortgesetzt werden. Ein graphisches Beschreibungsmittel, das in verständlicher Form und fehlerfrei im Sinne der Automatentheorie die binäre Steuerungsaufgabe eines oder mehrerer Prozeßschritte formuliert, ist der Prozeßablaufplan.

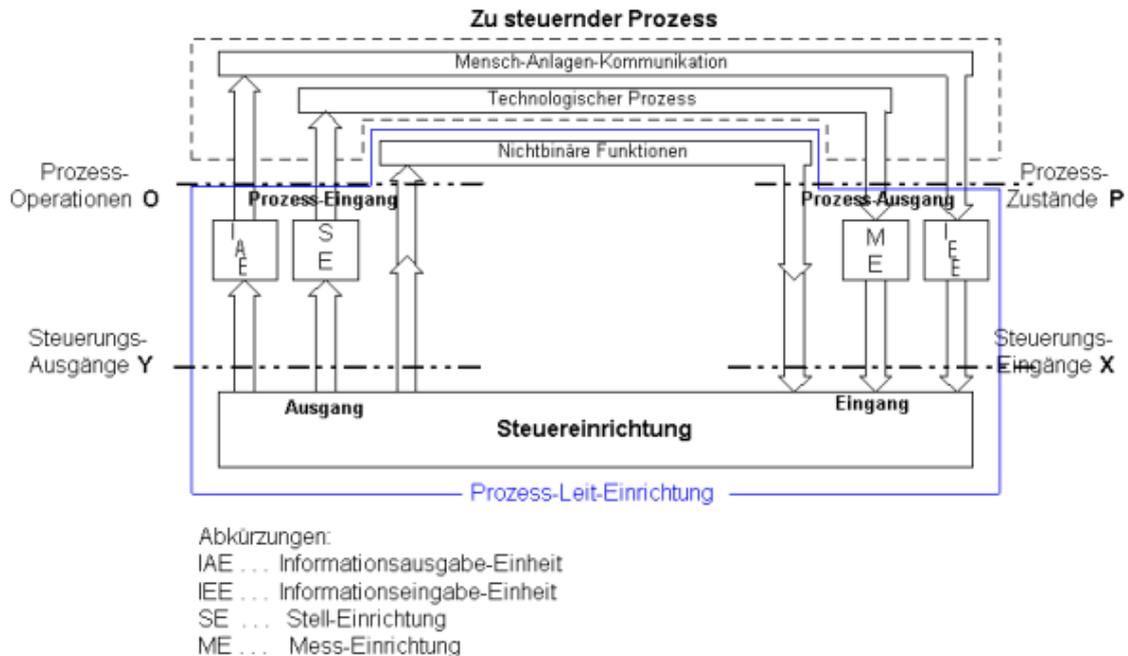
Um sich des Beschreibungsmittels PRAP richtig bedienen zu können, muß u.a. der Informationsfluß zwischen gesteuertem Prozeß / Prozeß-Abschnitt und der Automatisierungseinrichtung geklärt sein.

Dazu dient das Informationsflußmodell [siehe Seite 14].

siehe auch: Prozeßzerlegungsgraph [siehe Seite 17]

Empfohlene Basisliteratur: Reinschke, K.: Multivariable Control Springer Verlag und Verlag der Akademie der Wissenschaften , Berlin 1989

3.3 Informationsfluß-Modell



Das in der Abbildung dargestellte Modell zeigt den Fluß der Information zwischen gesteuertem Prozeß bzw. Prozeß-Abschnitt und der Steuereinrichtung, z.B. einer SPS. Als Prozeßabschnitt sollen technologisch "abtrennbare" Prozeßteile wie eine Teilanlage, eine Fertigungseinheit, ein Transportband usw. verstanden werden, die als technologischer Verbund den Gesamtprozeß ergeben.

Die Schnittstelle zwischen Prozeß und Steuereinheit wird durch die Binärvektoren [siehe Seite 197], \mathbf{X} und \mathbf{O} , \mathbf{Y} gebildet:

Wie die Abbildung zeigt, wird die am Prozeßausgang \mathbf{P} auftretende Anzahl sog. Prozeßzustandsvariable p in Eingangsvariable x des Eingangsvektors \mathbf{X} der Steuereinrichtung überführt.

Jede am Prozeßeingang \mathbf{O} wirkende Operation ist Abbild des Ausgangsvektors \mathbf{Y} der Steuereinrichtung. Die Ausgangsvariablen y werden in die Operationsvariablen o überführt.

Die Überführungs-Anforderungen müssen in den Variablenlisten der Datenbank [siehe Seite 39] enthalten sein, wobei EMSR-Stellen nach DIN 19227 oder Betriebsmittel-Kennzeichen nach DIN 40719 die Schnittstellen zum Prozeß darstellen. Die technische Realisierung erfolgt u.a. durch die Prozeß-Meß- und Stelltechnik.

Hierzu werden oft sogenannte nichtbinären Funktionen [siehe Seite 25] wie Zeitglieder und Zählglieder, arithmetische Funktionen, Kopierfunktionen (z.B. für Grenzwerte, Zeit- und Zählwerte usw.) und andere benötigt. Vergleichs- und Flankenfunktionen sind in den IEE bzw. ME enthalten.

Das Verhalten des Prozesses unter dem Einfluß der ihn steuernden, beeinflussenden Operationen spiegelt sich in Änderungen der Prozeßzustandsvariablen wider und ist im Steuerungsmodell Prozeßablaufplan (PRAP) dokumentiert. Im PRAP sind auch alle nichtbinären Funktionen eingebettet.

Kapitel 4

Prozeß-Zerlegung und Funktionsblock-Darstellung

4.1 Prozeßzerlegung

Wegen der Übersicht zu automatisierender Prozesse, ist eine Prozeßzerlegung unumgänglich, die zum Steuerungs-System [siehe Seite 189] führt. So ist es zweckmäßig, wenn Sie sich die aufgeführten Kapitel ansehen:

Ziele der Prozeßzerlegung [siehe Seite 19]

Kriterien der Prozeßzerlegung [siehe Seite 18]

Schritte der Prozeßzerlegung [siehe Seite 18]

Prozeßzerlegungsgraph [siehe Seite 17]

Funktions-Block-Darstellung (FBD) [siehe Seite 23]

Koppelungen zwischen Teilprozessen [siehe Seite 20]

Probleme und Fehler bei TP-Kopplungen [siehe Seite 23]

4.2 Prozeß-Zerlegungs-Graph (PZG)

Der Prozeßzerlegungsgraph beschreibt die steuerungstechnischen Beziehungen zwischen allen Teilprozessen eines Prozesses durch deren paarweise Steuerrelationen. Jeder Teilprozeß [siehe Seite 191] wird im PZG als Rechteck und jede Steuerrelation als gerichtete Verbindung (Pfeil-Linie) zwischen den Teilprozessen dargestellt. Bei der Anordnung der Teilprozesse sollten Sie nach der vorgegebenen Empfehlung verfahren:

Teilprozesse der MAK werden oben gezeichnet

.....MAK-Ebene

Teilprozesse für automatische Abläufe bzw. zur Koordinierung werden in der Mitte gezeichnet

.....Koordinierungseben

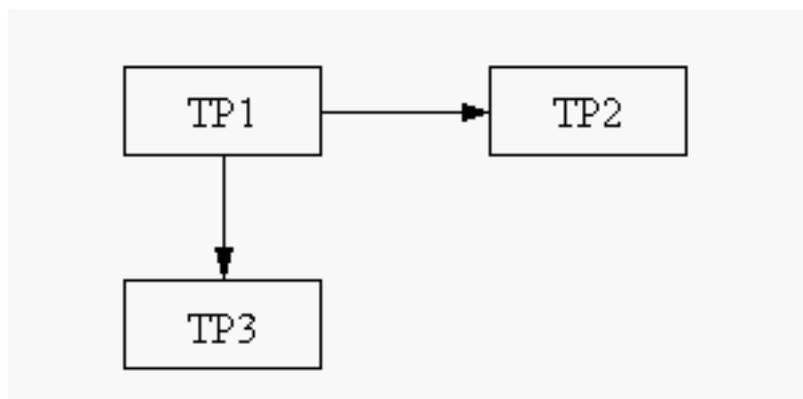
Teilprozesse der Stellglieder und Antriebe werden unten gezeichnet

.....Antriebsebene

siehe auch: Schritte der Prozeßzerlegung [siehe Seite 18]

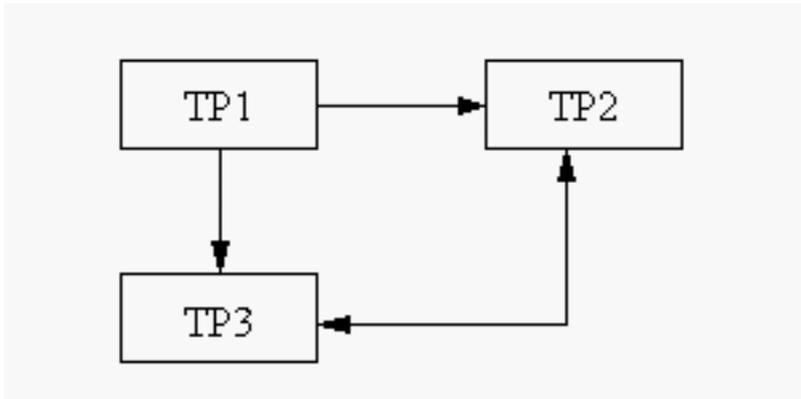
Die Teilprozesse und ihre Steuerrelationen (= steuerungstechnische Beziehungen) lassen sich verschieden darstellen. So zeigen die Beispiele:

•



TP1 beeinflusst einseitig die TP2 und TP3; TP2 und TP3 haben keine gegenseitigen Steuerrelationen.

•



TP1 wie oben, aber es beeinflussen sich auch die TP2 und TP3 und zwar in beiden Richtungen. Ein Pfeil an beiden Enden der Verbindungslinie erspart Doppelverbindungen.

Zu viele Verbindungslinien und Pfeile tragen meist nicht zur Klarheit des Modells bei.

PZG in Form des Funktionsbaustein-Modells, die Funktions-Block-Darstellung (FBD)

Bezeichnet man das bisher beschriebene Modell als Grobmodell, dann ergibt sich aus der Praxis heraus ein Feinmodell für die Prozeß-Zerlegung. Der Gedanke liegt sehr nahe, jeden Teilprozeß (TP) als Funktionsbaustein mit dem "Innenleben des zugehörigen PRAP" zu verstehen und links alle Eingänge, rechts alle Ausgänge einzutragen. Die Anordnung der TP erfolgt dann in "Signalfluß-Richtung" von links nach rechts.

Da es noch keinen PZG-Editor in der SPAS-Version 1 gibt, bleibt der Gestaltungswille frei, aber auch ohne Daten-Verbindung zum Projekt. Sie finden hier den PZG [siehe Seite 47] zum Beispiel-Projekt.

4.3 Kriterien der Prozeßzerlegung

Die Zerlegung des Gesamtprozesses in Teilprozesse [siehe Seite 191] erfolgt nach drei Kriterien:

- Der Prozeßablaufplan kann als Steuerungsmodell des Teilprozesses - TP - bei parallel laufenden Steuerungsanforderungen für den Entwerfer bald unübersichtlich werden. Dann lohnt es sich, mehr als einen TP zu entwickeln, das Problem weiter zu zerlegen.
(Das Prozeßablaufnetz kann dabei hilfreich sein)
- Jeder Teilprozeß kann sinnvoll einem Anlagenteil oder Anlagenabschnitt aus dem technologischen Schema (auch Prozeß-Abschnitt) zugeordnet werden. Dabei spielt der Aufbau aus Behältern, Transportbändern, Reaktoren, Förderschnecken usw. eine wesentliche Rolle, da dann die Struktur der Prozeßzerlegung dem Anlagenschema etwa entspricht.
- Alle Teilprozesse mit gleicher funktioneller Verarbeitung der Variablen (Antriebssteuerungen, Signalisierungen u.a.) sollten aus dem Anlagenschema bzw. der Aufgabenformulierung besonders entnommen und für die Prozeßzerlegung vorgemerkt werden. Häufig kann man für solche Steuerungsanforderungen Typen-Teilprozesse [siehe Seite 126] entwerfen und verwenden bzw. Typenprozeßablaufpläne nutzen, die schon existieren.

4.4 Schritte der Prozeßzerlegung

- Der erste Schritt sollte die Einteilung des Gesamtprozesses an Hand eines technologischen Anlagenschemas betreffen. Die so gewonnenen Prozeß-Abschnitte, die aus einem oder mehreren Anlagenteilen (Behälter, Maschinen usw.) bestehen, beinhalten sinnvollerweise je einen Prozeßschritt. [siehe Seite 183]
Sie sollten insbesondere im Falle eines Batch-Prozesses für das Zusammenwirken aller Prozeßschritte

ein Prozeßablaufnetz (PAN) entwickeln. Damit erhalten Sie ein Grob-Modell der Steuerung des Gesamtprozesses. Aus dem Grobmodell können Sie nun Teilprozesse [siehe Seite 191] des ersten Schrittes herauslösen, indem Sie einen oder mehrere Prozeßschritte zu einem Teilprozeß erklären. Solche, aus dem PAN gewonnenen Teilprozesse, müssen noch weiterbearbeitet werden (siehe PRAP-Editor [siehe Seite 61] und PZG-Editor [siehe Seite 47]).

Falls Sie für einen Fließprozeß das Steuerungskonzept entwickeln müssen, bedenken Sie, daß nach dem Anlauf alle Prozeßabschnitte simultan, bzw. mehr oder weniger parallel arbeiten. Es kann sinnvoll sein, das Steuerungskonzept für die Prozeß-Inbetriebnahme, für die -Außerbetriebnahme und z.B. -Störungsverhalten zu entwickeln. Betrachten Sie jeden Prozeßabschnitt und den(die) darin technologisch enthaltenen Prozeßschritt(e) als Teilprozeß des ersten Schrittes.

- Der zweite Schritt soll zwei Überlegungen enthalten:
 - Erstens:
Gibt es steuerungstechnische Beziehungen zwischen den Teilprozessen, dann sollten Pfeile - gerichtete Verbindungen - dies kennzeichnen. Geben Sie diesen Kopplungen Namen.
 - Zweitens:
Stellt sich bei näherem Betrachten eines TP des ersten Schrittes heraus, daß eine weitere Zerlegung sinnvoll ist, dann sollte das geschehen.
- Als dritten Schritt einer sinnvollen Prozeßzerlegung sollten Sie eine Einteilung nach den einzelnen Ebenen des PZG [siehe Seite 17] vornehmen. Jedem dabei entstehenden Teilprozeß ist wenigstens eine technologisch sinnvolle und einmalige Bezeichnung zuzuordnen. Es kann auch hilfreich sein, die Teilprozesse zu numerieren oder mit anderen Kurzzeichen zu versehen.

Die drei empfohlenen Ebenen:
Mensch-Anlagen-Kommunikation
Automatikabläufe - Koordinierungssteuerungen
Stellglied-/Antriebs-Steuerungen
- Im vierten Schritt sollten Sie zur Dokumentation Ihrer bisherigen Arbeit den Prozeßzerlegungsgraphen (PZG) zeichnen. [siehe Seite 47]
- Der fünfte Schritt erfordert die Zuweisung zu allen Prozeß- und Operationsvariablen des PRAP-Types.

Zu jedem Teilprozeß gehört eine bestimmte Anzahl Eingangs- und Ausgangsvariable, die mit technologisch prozeßnahen Bezeichnern (Name) versehen werden sollten und eine direkte Adresse - Anschlußbezeichnung am Steuergerät - erhalten müssen. Variable im Speicherbereich des Steuergerätes können entweder vom Maschinencompiler automatisch vergeben werden oder eine Adresse des Speichers erhalten.

Tragen Sie jede Variable am TP-Kästchen des PZG außen an, die Eingänge links, die Ausgänge rechts.

Achten Sie darauf, daß Ausgangsvariable auf den TP bezogen sind, d.h. sie dürfen nur in einem d.h. in keinem anderen TP vorkommen!

Damit ist der Übergang vom PZG zur FBD (Funktionsblock-Darstellung) gekommen. Lesen Sie dort weiter! [siehe Seite 23]



Tip

Es lohnt sich, eine Variablenliste - Excel-Tabelle - anzulegen, weil sie in das Steuerungsprojekt importierbar ist. Siehe ImportVarTab [siehe Seite 198]

4.5 Ziele der Prozeß-Zerlegung

Ziele der Prozeßzerlegung sind,

- einen Überblick über das gesamte Steuerungsproblem, den Systementwurf [siehe Seite 189], zu erhalten und
- Steuerungs- und Regelungsaufgaben getrennt als Teilprozesse (TP) [siehe Seite 191] analysieren zu können, ohne dabei die gegenseitigen, steuerungstechnischen Beziehungen außer acht zu lassen. Zu jedem TP gibt es Variablen, die in der Datenbank [siehe Seite 39] zu erfassen sind, wobei zu jeder Variablen der Bezug zum technologischen Schema (Fließbild, Anlagen- oder Maschinenschema) durch EMSR-Stellen bzw. andere Betriebsmittelkennzeichen zu dokumentieren ist.
- die Teilprozesse (TP), die in einer gemeinsamen Ressource [siehe Seite 187] abzuarbeiten sind, entweder in einen eigenen Prozeßzerlegungsgraphen (PZG) einzutragen oder in einem einzigen für alle Ressourcen. Im letzteren Fall sind die TP zu kennzeichnen, welcher Ressource sie zugehören.
- die durch Zerlegung gewonnenen Teilprozesse auch in der Darstellung durch ihre aktuellen Größen (Konstante und Eingangs-/Ausgangsvariable) zu ergänzen sowie den Bezug zum Typen (z.B. Typen-PRAP) herzustellen. Damit ergibt sich die Funktions-Block-Darstellung, kurz FBD.

Aus diesen Gründen ist es notwendig, daß Sie sich ein sinnvolles System von Teilprozessen schaffen. Dabei können Teilprozesse auch Stabilisierungsprobleme für Prozeßgrößen beinhalten, also Regelungsaufgaben sein.

Es hat sich gezeigt, daß eine intuitiv gewählte Prozeßzerlegung bei der Erstellung der Prozeßablaufpläne ungünstig sein kann, weil z.B. der Bezug zum leitenden Prozeß an Hand des technologischen Schemas (R- und I- Fließbild) verloren geht.

Eine der Prozeßzerlegung vorgelagerte Komponente, in deren Ergebnis ein ungestörter Prozeßablauf modelliert wird, und eine sinnvolle Prozeßzerlegung entsteht, ist das Prozeßablaufnetz(PAN). [siehe Seite 45] Es ist erwiesen, daß man durch den Einsatz des PAN insbesondere Chargen- oder Batch-Prozesse im Überblick behält und bei der Zerlegung in Teil-Prozesse sich durch das Softwaretool PAN-Editor effektiv unterstützen lassen kann.

Jeder Teilprozeß enthält prinzipiell die drei Komponenten des Prozesses, wie sie im Informationsflußmodell [siehe Seite 14] dargestellt sind. Reine Bedien- und Signalisierungs-Teilprozesse beinhalten häufig keinen Anteil des technologischen Prozesses. Ebenso gibt es Teilprozesse ohne Zeit-/Zählglieder, Vergleicher usw.

Das entstehende Teilprozeß-System benötigt Koppelgrößen, um die Teilprozesse miteinander steuerungstechnisch zu verbinden, und das Verhalten des Prozesses durch die Wirkung der Prozeßsteuerung zu gewährleisten. Das Ergebnis dieser Dekomposition wird im Prozeßzerlegungsgraphen dargestellt.

Die Prozeßzerlegung wird in mehreren Schritten nach drei Kriterien [siehe Seite 18] ausgeführt.

Zurück zum Entwurfsprozeß. [siehe Seite 13]

4.6 Koppelungen zwischen Teilprozessen - Interactions -

Der Begriff "Koppelung" soll bedeuten, daß ein Informationsaustausch zwischen zwei Teilprozessen (TP) durch Koppelvariable [siehe Seite 170] existiert. Das Steuerungssystem [siehe Seite 189] beinhaltet alle Koppelungen.

Im zweiten Schritt der Schritte der Prozeßzerlegung [siehe Seite 18] ist auf das Einführen der Koppelungen verwiesen, und im Thema Prozeßzerlegungsgraph [siehe Seite 17] sind zwei Beispiele der formalen Darstellung enthalten. Es handelt sich um eine Problemlösung, paarweise Steuerungsabläufe in Form von Prozeßablaufplänen (PRAP) miteinander zu verbinden, zu synchronisieren, damit das Gesamtergebnis des Steuerungsprozesses, also das Zusammenwirken der Teilprozesse, den technologischen Prozeß-Anforderungen entspricht. Es werden

- die natürliche Koppelung über Prozeßgrößen oder

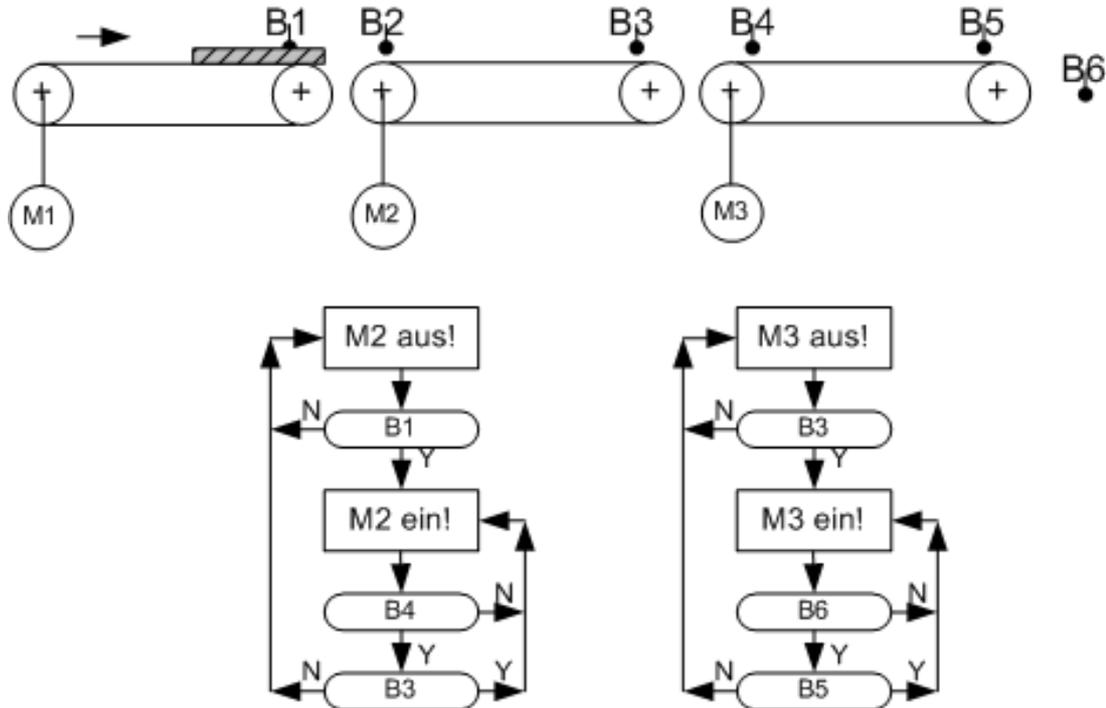
- eingeführte Koppelvariable

verwendet.

Die praktische Ausführung von Koppelungen soll nun behandelt werden:

- Natürliche Koppelung

Zwei Teilprozesse sind allein durch den gesteuerten Prozeß untereinander verbunden, wenn durch die Anordnung von Sensoren der korrekte Steuerungsprozeß gewährleistet ist. Als Beispiel sei ein Transportabschnitt betrachtet, bei dem ein Stückgut auf Transportbändern bewegt wird.



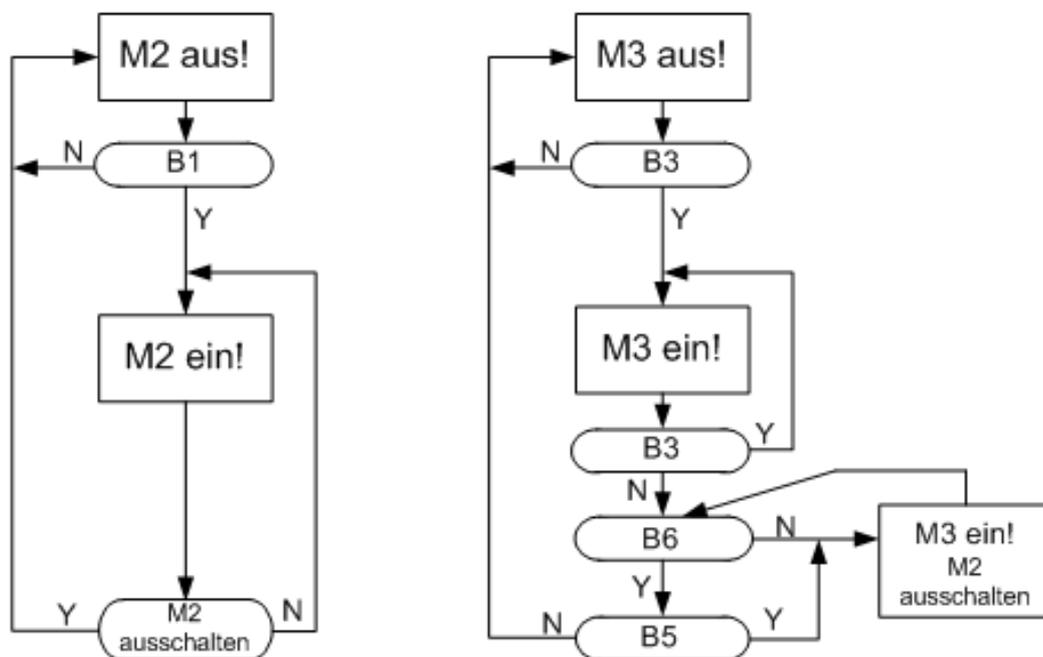
Das erste Band werde nicht beachtet, das zweite beginnt zu laufen, sobald B1 anspricht. Es schaltet wieder ab, sobald das Stückgut bei B4 angekommen ist und an B3 verlassen hat. In diesem Sinne arbeitet auch die Steuerung für den Motor 3. B3 ist in beiden PRAP vorhanden und garantiert die Koppelung. Wenn M3 einschaltet, läuft M2 noch, bis B3=N. Somit wird ein teilweiser Parallel-Lauf beider TP erreicht.

Mit dem Prozeßablaufnetz [siehe Seite 45] modellierte Prozesse ergeben Teilprozesse, die nur natürlich gekoppelt sind.

Die Wirkungsrichtung dieser Koppelung - Pfeil im PZG - geht von TP Motor 2 nach TP Motor 3, da B3 durch die Funktion des Motors 2 primär beeinflusst wird.

- Eingeführte Koppelung

Das gleiche technologische Beispiel mit eingeführter Koppelung, also mit Koppelvariable, sehen Sie nun. Die Koppelvariable "M2 ausschalten" ist in dem PRAP des Motors 3 in den Operationen "M3 aus" und "M3 ein" mit der Aussage "M2 NICHT ausschalten" vertreten. So kann M2 einschalten, wenn B1 das Stückgut meldet und bleibt in Betrieb, bis "M2 aussch." wahr wird. Um den gleichen Steuerungsablauf zu erhalten wie oben, ist noch eine Operation hinzugekommen.



Der Sensor B4 wird nicht mehr benötigt. So können Sie mit dem Entwurf bereits Geld sparen, und aus der Sicht der Zuverlässigkeit ist diese Einsparung auch ein Gewinn. Aber dann sind beide Instanzen (TP Motor 2 und TP Motor 3) nicht mehr von einem Typen-PRAP!

- Natürliche Koppelung im Steuerungssystem

Da Sie in diesem Fall auf die Dynamik [siehe Seite 150] der Prozeßzustandsvariablen achten müssen, um einen sicheren Steuerungsablauf zu erhalten, ist es auch wichtig, die Sensoreigenschaften zu kennen. Aber diese Kenntnis muß man sich immer verschaffen. Beide Instanzen erhalten natürlich die gleiche Boolesche Eingangsvariable.

- Eingeführte Koppelung im Steuerungssystem

Der TP Motor 2 wartet auf die Meldung vom TP Motor 3, ehe er abschaltet. Damit hat sich die Koppelungsrichtung geändert gegenüber der natürlichen Koppelung. Die Koppelvariable "M2 ausschalten" ist im TP Motor 3 eine Operationsvariable (Sender der Information), im TP Motor 2 ein P-Variable (Empfänger der Information). Die Dynamik ist auch hier zu beachten, damit die beabsichtigte Wirkung auch eintritt! Beide Variablen sind schon im Typen-PRAP einzubauen. Dieser Aspekt kommt bei der natürlichen Kopplung gar nicht in Betracht.

Die Adressenvergabe:

- Das Eintragen der direkten Adressen [siehe Seite 58] oder die Bereitstellung von symbolischen Adressen erfolgt zum Zwecke der Instanzierung im zugehörigen Typ-Programm (PROGRAM) oder in der Ressource (RESOURCE)
- Sie können in beiden PRAP-Typen, gemeint sind jetzt der Sender-PRAP ("USAGE"=Output) und der Empfänger-PRAP("USAGE"=Input), die gleichen Namen (Identification) einsetzen³, um für sich selbst deutlich werden zu lassen: Es sind die gleichen Variablen. Und ebenso selbstverständlich müssen beide Instanzen die gleiche Adresse im Merkerbereich durch Sie erhalten, wenn Sie "Address Memory" (unter "USAGE" in PROGRAM) angeben.
- Tragen Sie "Local Static" bei "USAGE" in PROGRAM ein, so erfolgt eine automatische Adressenzuweisung im RAM, die Ihnen nicht mitgeteilt wird. Voraussetzung: Beide haben die gleiche Identifikation, den gleichen Namen.
- Wollen Sie mit verschiedenen Namen in den Typen arbeiten, dann ist es dennoch eine einzige

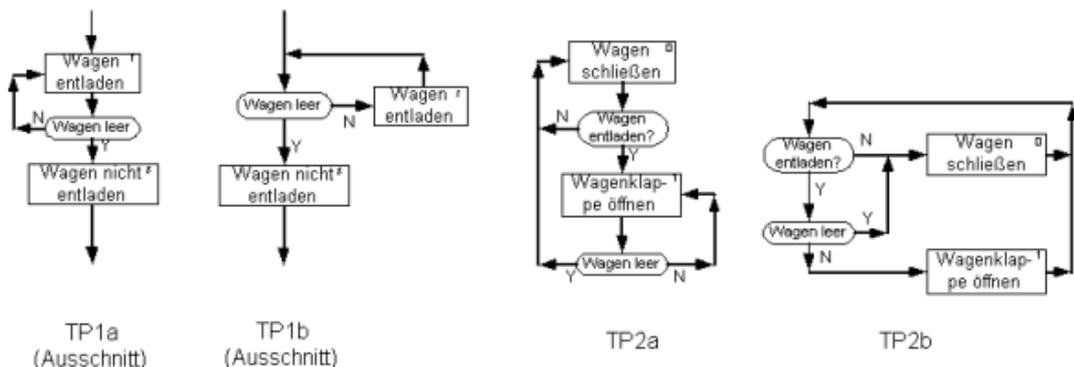
³ Den gleichen Namen einsetzen bedeutet nicht zuweisen eines schon vorhandenen! Geben Sie die gleiche Zeichenfolge nochmals ein oder kopieren Sie diese.

und bei "USAGE" entsprechend "Address Memory" wählen.

4.7 Probleme und Fehler bei TP-Koppelungen

Im folgenden sind Hinweise zur Adressierung von Koppelvariablen und Probleme sowie Fehler bei der Behandlung von Koppelungen aufgeführt:

- Adressierungs-Probleme und Fehler
Sowohl die natürliche als auch die eingeführte Koppelung haben einen "Sender" und können viele "Empfänger" haben. Problematisch ist ein "vergessener Empfänger", wenn z.B. ein Schreibfehler vorliegt, also die gemeinsame Adresse des Senders mit allen Empfängern dadurch bei einem Empfänger falsch ist. Schlimm wird es auch, wenn ein Empfänger gar die falsche Adresse, d.h. Namen hat.
- Die Dynamik der Koppelvariablen
ist zu beachten. Der Sender muß so lange diese Variable true oder false halten, wie es die angekoppelten TP, alle "Empfänger", für deren Funktion benötigen. Dieser Fehler ist im Entwurfsablauf der PRAP oft schwer zu entdecken.
- Die Aufruf-Reihenfolge der Instanzen im Programm
kann sowohl bei natürlichen als auch bei eingeführten Koppelungen zu dynamischen Fehlern führen, wenn der Sender den Wert der Koppelvariablen ändert, die Empfänger (manchmal nur einer von allen) aber den "alten Wert" noch benötigt und erst im Folgezyklus der SPS den neuen.
- Falsche Ausgabe einer Koppelung
kann geschehen, wenn eine Situation [siehe Seite 183] technologisch bedingt durchlaufen [siehe Seite ?] werden kann. Beachten Sie im folgenden Bild



beim Teilprozeß TP1a die Möglichkeit, einen leeren Wagen auch noch entladen. Das geschieht nur deshalb, weil mit dem Erreichen der Operation Or auch im Teilprozeß TP2a mit "Wagen entladen" = Y der Übergang nach O1 erfolgt. Natürlich erfolgt im nächsten Zyklus (Takt) wieder der Rücklauf zur Operation O0, weil der Wagen leer ist und die Wagenklappe wegen der Trägheit sich nicht öffnen wird. Aber solche falschen Koppelungen sollten beseitigt werden, wie es TP1b zeigt. Nur ein nicht leerer Wagen wird entladen.

Geht man davon aus, daß die Situation r in TP1a und b über die Entladedauer stabil ist, dann kann auch TP2a zu TP2b umgezeichnet werden. Das hat zum Vorteil, daß im Fall des leeren Wagens das Öffnen nie angewiesen wird. Also könnte TP2b auch mit TP1a gut zusammenarbeiten.

4.8 Funktions-Block-Darstellung (FBD)

Die Ermittlung der Eingangs- und Ausgangs-Variablen zu jedem Teilprozeß, wie dies im Kapitel Schritte

der Prozeß-Zerlegung [siehe Seite 18] im fünften Schritt beschrieben ist, muß aus drei Gründen erfolgen:

- Die natürlichen Koppelungen sind letztlich Eingangs-Variable. Jede dieser Eingangsvariablen ist genau einem TP zugeordnet. Die angekoppelten TP enthalten zum Variablen-Namen die TP-Nummer.
- Die eingeführten Koppelungen sind "an der Quelle" Ausgangsvariable. Dort sind sie auch zugeordnet. Eingangsvariable sind sie bei allen angekoppelten TP. Um sie deutlicher zu erkennen, werden sie möglichst nicht mit den Prozeß-Variablen gemeinsam dargestellt.
- Die Teilprozesse sind Instanzen der PRAP-Typen, in denen Prozeßvariable und Operationsvariable als formale Variable enthalten sind. Jeder TP muß daher deutlich den Bezug zum Typen erkennen lassen.

Die Standardisierung in der IEC 61944-1 erfüllt SPAS nicht vollständig, wählt sie aber als Vorbild.

Weiteres Vorgehen:

- Tragen Sie in eine Tabelle, am besten eine Exceltabelle (ImportVarTab [siehe Seite 198]) Namen und Datentyp zusammen mit der TP-Bezeichnung ein.
- Parallel dazu tragen Sie die Namen an jedem Teilprozeß, verbunden mit einem Eingangs- bzw. Ausgangspfeil, ein.
- Ergänzen Sie jedes TP-Kästchen mit dem Namen des Typen und dessen Variablen. Ein kleines Beispiel finden Sie hier: Beispielprojekt Lampe/Stern-Dreieck-Schaltung [siehe Seite 73]
Achten Sie in diesem Beispiel auf die Entwicklungs-Folge der PZG hin zum PZG/FBD.

Kapitel 5

Im PRAP eingebettete, nichtbinäre Funktionen

5.1 Nichtbinäre Funktionen

Binäre, also zweiwertige, Funktionen sind für die Abarbeitung der automatentheoretisch begründeten PRAP erforderlich. Der Begriff soll darüber hinausweisen, obwohl eine Flankenerkennung auch eine binäre Funktion ist. Er faßt alle nicht im "Automatenteil des PRAP" erforderlichen funktionellen Verknüpfungen zusammen.

Wichtige Vorbemerkung:

Um Datentypen-Anpassung brauchen Sie sich als Projektant nicht zu kümmern!! Nur in Ausnahmefällen, auf die Sie besonders hingewiesen werden, verwenden Sie bitte die Standardfunktionen nach IEC 1131-3.

Alle Funktionen in alphabetischer Reihenfolge:

- Arithmetik-Funktion [siehe Seite 37]
- Flankenerkennung [siehe Seite 36]
- Übertragungs-, Umspeicher- oder Kopierfunktion [siehe Seite 37]
- Vergleichfunktion [siehe Seite 34]
- Zählglieder, Counter???
- Zeitglieder, Timer [siehe Seite 26]

Neben der Projektierung der Prozeßvariablen durch binäre Größen, die unmittelbar aus Meßgebern bzw. Sensoren (Lichtschranken, Stößeltaster usw.) gewonnen werden, treten im PRAP häufig andere Probleme auf, die mit den folgenden Funktionen lösbar sind.

Aufgaben der Zeitsteuerung, ebenso Zählaufgaben, können erforderlich sein, wenn Sie eine Prozeßgröße "umgehen wollen statt sie zu messen" oder aber eine wirklich zeitabhängige Steuerung entwerfen wollen. Die Zählglieder sind hier ausschließlich Software-Zähler. Die Erläuterungen zu Zeitgliedern (Timer) [siehe Seite 26]

und

Zählgliedern (Counter)???

zeigen Ihnen, wie sie im PRAP einzusetzen sind.

Wichtige Bemerkung für Zeit- und Zählglieder:

Um diese Funktionen bedarfsgerecht ein- und auszuschalten, achten Sie auf die Operationenfolge im PRAP. In Folgeoperationen eingeschriebenes und markiertes "ein" hält ohne Unterbrechung die Funktion aufrecht. Folgt auf eine Operation mit "aus" eine Operation mit "ein", so erfolgt ein Neustart mit Löschen des Inhaltes und Einlesen des Sollwertes (Zeit- bzw. Zählwert).

Da Zählwerte oder auch Zeitwerte abhängig vom Prozeßgeschehen, also von der Prozeßsituation sein können, ist es sinnvoll, solche Vorgabewerte z.B. auf ein Merkerwort zu schreiben. Deshalb gibt es die Übertragungs- oder Kopierfunktion.

Natürlich lassen sich auch andere Anwendungen als eben erwähnt denken. Ein Reglerbaustein benötigt eventuell auch vom Prozeß her verschiedene Sollwerte. Oder Grenzwerte für Alarme, oder Signalisierungen ändern sich in Abhängigkeit von der Prozeßsituation. Auch Ausgaben an den Prozeß in digitaler Form z.B. an Ausgangsworte sind projektierbar.

Mit der Vergleichfunktion

werden Istwerte (im ganz allgemeinen Sinne) mit Sollwerten verglichen. Damit können Sie im PRAP von einer Grenzwertbeurteilung abhängige Funktionen planen. Sie können auch als Istwert eine momentane Zeit in einem Timer vergleichen mit einem festen Zeitwert, also "fragen", ob z.B. 20 Minuten im Timer

XYZ abgelaufen sind.

Berechnungen verschiedener Art werden für ganz unterschiedliche Zwecke benötigt.

Mit Hilfe der Arithmetik-Funktionen

können Sie dieses Problem im PRAP mit erledigen. Derzeit sind nur die Grundrechenarten sowie Schieben und Rotieren projektierbar. Trigonometrische Funktionen oder Wurzelziehen usw. sind noch nicht verfügbar.

In bestimmten Fällen wird eine

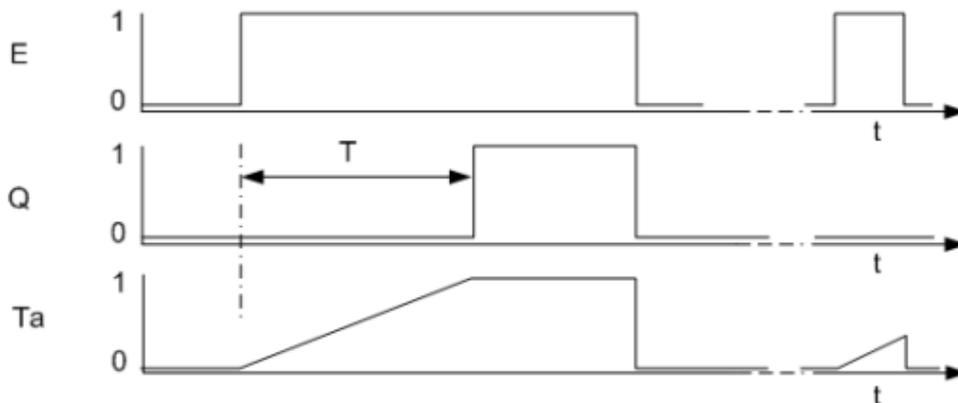
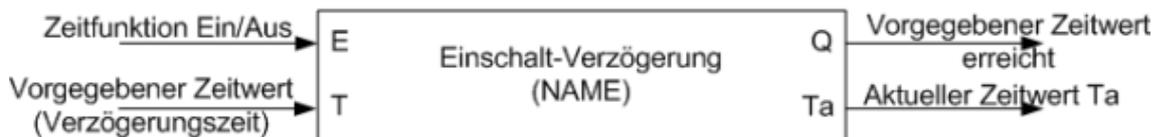
Flankenerkennung benötigt, die sich beim Übergang einer Booleschen Variablen von false auf true oder umgekehrt ergeben soll. Der Impuls ist als Prozeßvariable - Text-Inhalt eines Condition-Symbols - im PRAP projektierbar. Er beginnt mit der Flanke der sogenannten "primären Prozeßvariablen" und endet nach einem Arbeitszyklus der SPS. Z.B. können "Einlaufbeginn eines Kartons in die Waage", "Ende des Waggons erreicht" solche Texte sein, deren Primärvariable "Karton in Waage" oder "Wagon eingefahren" sein könnten.

Hinweise zu Besonderheiten der Hardware finden Sie im entsprechenden Text zu den Sprachen der SPS [siehe Seite 7].

5.2 Zeitglieder - Timer

Im Rahmen der binären Prozeßanalyse werden ausschließlich Zeitglieder mit Einschaltverzögerung - nach IEC 1131-3 die Standardfunktion TON - benötigt, da im PRAP alle Zeitfunktionen damit erfaßbar sind. Das Projektieren von Zeitgliedern erfolgt selbstverständlich unabhängig vom Hersteller der praktisch eingesetzten Ressource. Daher sind nur solche Anschlüsse vorgesehen, die international vereinbart wurden.

Jedes Zeitglied erhält einen Namen (Name) als Identifikator und eine Bezeichnung (Statement).



Anschlußbezeichnungen gemäß IEC1131-3⁴

Die im Zeitdiagramm dargestellte Funktion eines Zeitgliedes mit Einschaltverzögerung enthält alle wichtigen Eigenschaften:

- Start der Verzögerungszeit T mit dem Einschalten des Eingangs E (E schaltet ebenso das Zeitglied ab)

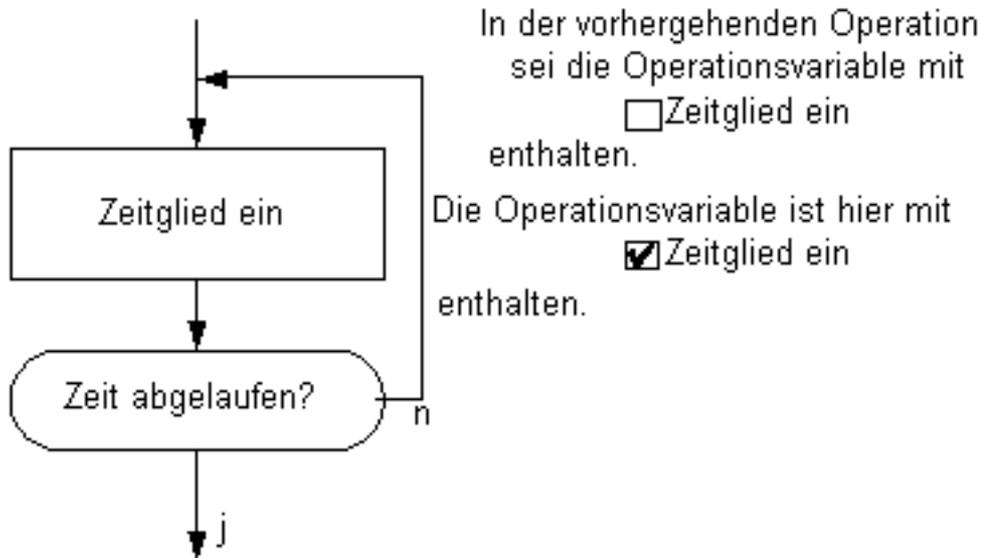
⁴ Eingänge:

- E: IN = Input
- T: PT = Preset Time

Ausgänge:

- Q: Q = Quitt
- Ta: ET = Elapsed Time

- Ausgang Q schaltet ein, sobald T beendet ist
- Bei kürzerer Einschaltdauer als T bleibt Q ausgeschaltet
- Q schaltet zusammen mit E aus



Im Teilprozeß [siehe Seite 191] verwenden Sie für E eine technologisch passende Bezeichnung einer Operationsvariablen. Für den Ausgang Q formulieren Sie die technologisch sinnvolle Prozeßzustandsvariable. Diese Form des Vorgehens entspricht dem Informationsflußmodell [siehe Seite 14].



Tipp

Sie müssen Q nicht in der gleichen Situation [siehe Seite 183] eintragen, wie das im Beispiel zu sehen ist. Eine "spätere Abfrage" bedingt jedoch, das Zeitglied so lange eingeschaltet zu lassen!

Beide Variablen können Sie selbst adressieren oder die Zuweisung einer Merkervariablen dem Precompiler [siehe Seite 178] überlassen.

Beispiele:

Operationsvariable bewertet

Füllkontrollzeit ein / aus

Anlaufwarnzeit Start/Stop

Prozeßvariable bewertet

Füllkontrolldauer zu Ende ja/nein

Anlaufwarnung beendet ja/nein

Wichtige Hinweise:

- Beginnen Sie bei der Konfigurierung des Zeitgliedes immer mit der Operationsvariablen, die das Zeitglied steuert.
- Die Operation, in der Sie das Zeitglied einschalten müssen, wird durch den Prozeßablauf bestimmt. Dort ist die Operationsvariable zu markieren. Diese Operationsvariable muß bezüglich ihrer Verwendung (USAGE) als "Local Static" oder "Output" gekennzeichnet werden. Sie können ein solches Zeitglied auch über mehrere Operationen eingeschaltet lassen (siehe Tip). Aber das Abschalten muß spätestens in einer Operation vor dem (Wieder-)Einschalten erfolgen, weil damit auch die Verzögerungszeit T wieder neu geladen und gestartet werden kann.

- Die Prozeßvariable eines Zeitgliedes, also den Ausgang Q, können Sie wie jede andere benutzen, also nicht nur in dem Teilprozeß, in dem Sie das Zeitglied starten, etwa um sie im Sinne einer natürlichen Koppelung zu verwenden. Dann muß die Variablenverwendung mit "Input/Output" angegeben werden, anderenfalls mit "Local Static".

Sie brauchen diese aber auch überhaupt nicht zu projektieren, also nicht "abzufragen", sofern der Hersteller der Hardware das zuläßt. Sie können durch Vergleich ermitteln, welche "Teil"-Zeit abgelaufen ist (Timer-Vergleich).



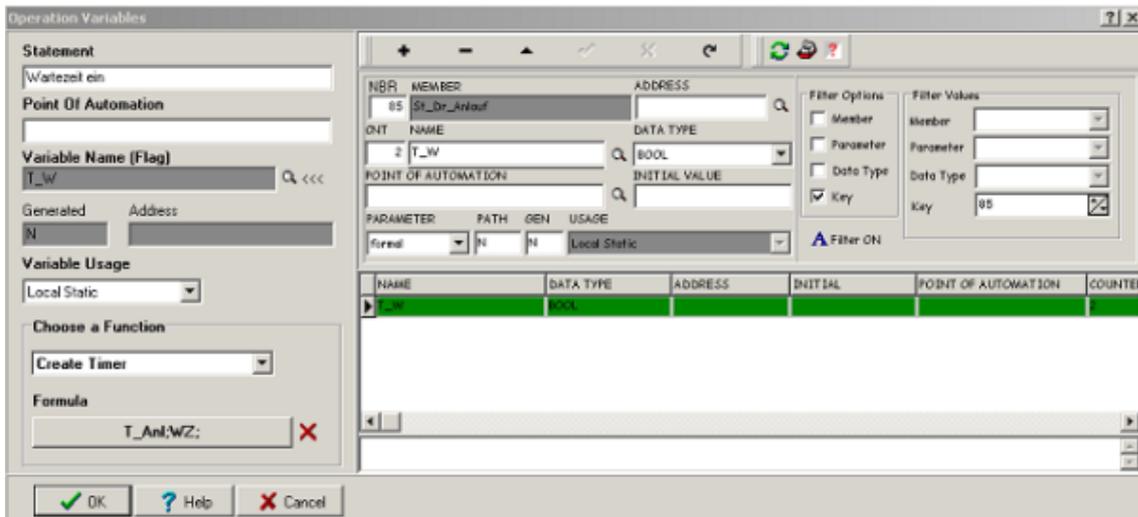
Warnung

Überschreitung des Maximalwertes für Zeitglieder

begegnet man am besten durch Programmierung einer Addier-Funktion [siehe Seite 202], die mit einer zyklischen Task aufgerufen wird. Die Zeitvariable ist vom Typ TIME, die zyklisch zu addierende Konstante ebenso und im Wert gleich der Zykluszeit. Ein Vergleich gibt Antwort auf die Frage, ob ein bestimmter Zeitwert erreicht ist.

Projektierungs-Ablauf

Der Beginn erfolgt durch Betätigen der rechten Maustaste auf die Operation, in der ein Timer eingeschaltet werden soll. Danach "Data Access"-Klick (oder Doppelklick [siehe Seite 150]) und eine neue Variable einführen und markieren, die den Timer ein- und ausschalten soll (Eingang E). Schreiben Sie das Statement ein und klicken auf , um "Variable Name (Flag)" als neue Variable benennen zu können. Im nachfolgenden Bild sehen Sie ein Beispiel aus dem PRAP "Stern-Dreieck-Anlauf".



Die "Übernahme" des Namens erreichen Sie mit Klick auf , danach geben Sie "Output" als USAGE an und wählen bei "Choose Function" die Funktion "Create TIMER". Ein Klick auf "Formula - None" öffnet die Timer-Configuration:

Benutzen Sie bitte die angebotenen Hilfetexte.

Die Q-Variablen des Timers enthält eine "Condition". Stellen Sie auf dieses Langrund den Cursor, betätigen Sie die rechte Maustaste, und nach Klick auf "Data Access" (oder DreiMalKlick [siehe Seite ?]) finden Sie ein ähnliches Formular vor wie für die Ein-Ausschaltvariable, hier wieder das Beispiel aus der Stern-Dreieck-Anlaufschaltung:

Benutzen Sie bitte die dort angebotenen Hilfetexte.

Weitere Einzelheiten zum Konfigurieren siehe: Timer-Projektierung [siehe Seite 203]

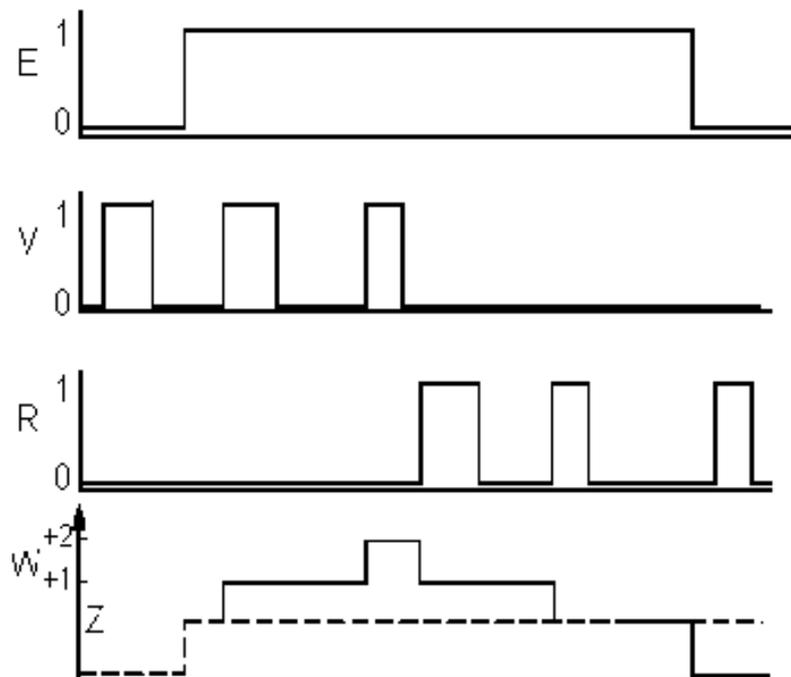
Ein ausgeführtes Beispiel finden Sie bei Stern-Dreieck-Anlaufschaltung, Projektieren des Zeitgliedes. [siehe Seite 88]

5.3 Zählglieder - Counter - Grundsätzliches und Projektierung

Für Zählglieder benötigen Sie

- die Eingangs-Bit-Größe E zum Einschalten / Ausschalten (Operationsvariable)
- eine Eingangs-Bit-Größe zum Zählen (V / R), bei Vorwärts-Rückwärts-Zählern auch beide (Operations- oder Prozeßvariable)
- die Ausgangs-Bit-Größe Q zum Anzeigen, ob der Zählwert erreicht ist (Prozeßvariable)

- einen Zählerwert Z in Form einer Konstanten oder einer Wortadresse, der auch den Wert 0 haben kann. In diesem Fall kann das Rückwärtszählen in negative Werte reichen, das nicht alle Hersteller ermöglichen



Zur Information klick auf: [Anschlüsse gemäß IEC1131-3⁵](#)

Wichtig für alle Zähler ist der E-Wechsel von 0 auf 1 (=Einschaltvariable) vor dem ersten Zählvorgang (siehe erster Impuls bei V), weil ansonsten der Zählwert Z nicht eingelesen wurde.

Richten Sie es im PRAP so ein, daß Sie Q erst "abfragen", also die entsprechende P-Variable in einen Pfad eintragen, wenn die Operation mit der Operationsvariablen zum Einschalten entweder bei deren Erreichen von einem Haltepfad ohne diese P-Variable stabilisiert wird, oder Sie können in einer Folgesituation ermitteln, ob der Zählvorgang zu Ende ist.

5

Anschlußbezeichnungen der Eingänge:

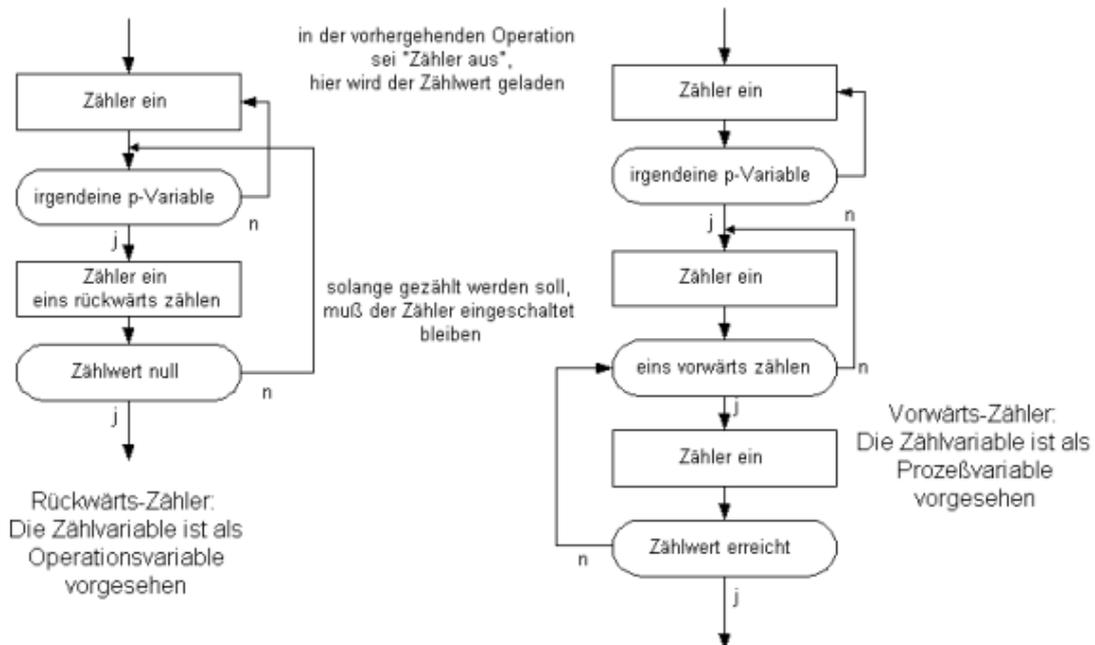
- E: LD = Load Preset Value, Lade PV
- V: CU = Count Up, Vorwärts Zählen
- R: CD = Count Down, Rückwärts Zählen
- (nicht vorgesehen) R = Reset
- Z: PV = Preset Value, Start-Zählwert

Anschlußbezeichnungen der Ausgänge:

- Q: QU = Quitt upcount (wird nicht von QD getrennt)
- Q: QD = Quitt downcount
- W: CV = Counter Value

Während die Zählvariablen mit ihrer Vorderflanke den Zähler weiterstellen, also Kurzzeitsignale darstellen, muß die Variable E solange eingeschaltet bleiben, wie der Zähler arbeiten soll. Die Zählerfunktion ist insofern vergleichbar mit der Zeitfunktion, nur anstelle ET ist der Zählerinhalt W (=CV) zu denken.

Die beiden Beispiele sind nicht für alle SPS geeignet, also in den Herstellerunterlagen nachlesen oder Hinweise beachten.



Sämtliche Booleschen Variablen können Sie adressieren oder dies dem Precompiler überlassen (USAGE: Local Static).

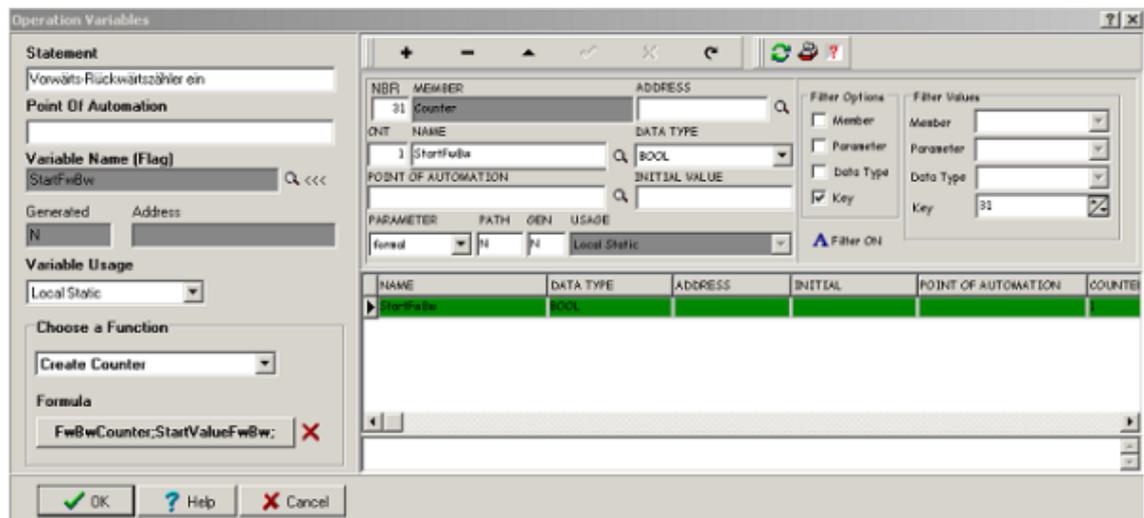
Projektierungs-Ablauf

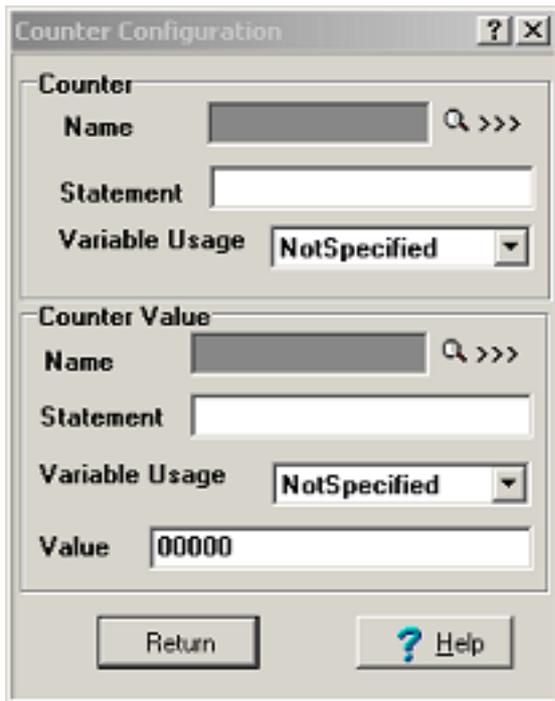
Hinweis:

Den PRAP hierzu finden Sie:BeispCounter [siehe Seite 116]

Der Beginn erfolgt durch Betätigen der rechten Maustaste auf die Operation, in der ein Counter eingeschaltet werden soll. Danach "Data Access"-Klick (oder DreiMalKlick [siehe Seite 150]) und eine neue Variable einführen und markieren, die den Counter ein- und ausschalten soll (Eingang E). Schreiben Sie das Statement ein und klicken auf , um "Variable Name (Flag)" als neue Variable benennen⁶ zu können. Geben Sie "Local Static" als USAGE an und wählen bei "Choose Function" die Funktion

6



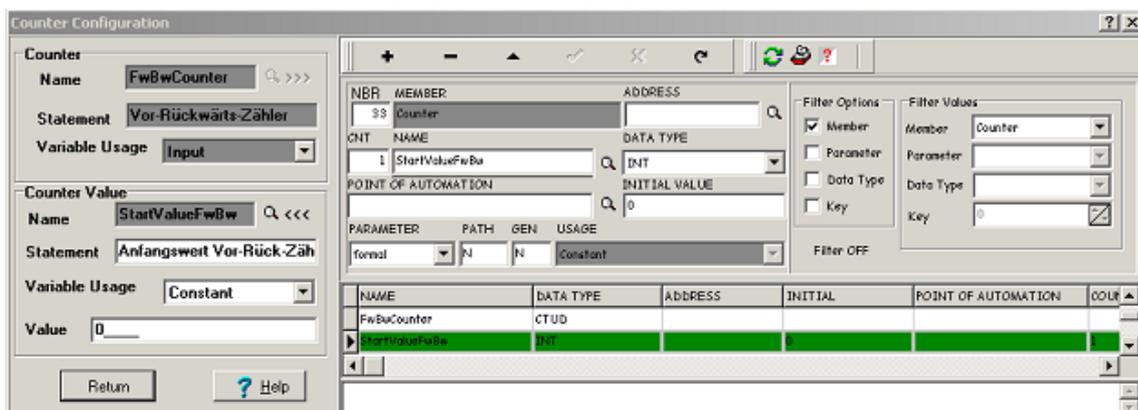


Tragen Sie schon das Statement zum Zähler ein und klicken dann auf  >>>, wodurch Sie den Namen, also das Kurzzeichen zum Statement, eingeben können. Der Datentyp ist hier eine Standard-Funktion "CTUD", wie sie von IEC 1131-3 vorgeschrieben ist. USAGE ist sprachabhängig???

Als Beispiel ist hier das folgende Bild angegeben:



Danach ist der Zählerwert, Datentyp immer INT, anzugeben. Im Beispiel ist er 0, weil ein Vorwärts-Rückwärts-Zähler alle Zugänge und alle Abgänge erfassen soll, ohne eine Vorgabe:



Damit ist zunächst der Counter konfiguriert. Die Ausgänge Q und W müssen nicht "belegt" werden (bei manchen Sprachen schon). Das Projektieren des Q-Ausgangs beginnt mit dem Einführen einer "Condition". Stellen Sie auf dieses Langrund den Cursor, betätigen Sie die rechte Maustaste, und nach Klick auf "Data Access" (oder Doppelklick) finden Sie ein ähnliches Formular vor wie für die

Ein-Ausschaltvariable. Im Beispiel zu erkennen:

Mit Einführen der Variablen FwBw_0 ist es noch nicht getan. Sie müssen noch die Funktion "Counter Output Q" wählen, und nach Klick auf Formula wählen Sie den zutreffenden Zähler aus, im Beispiel "FwBwCounter".

Zähler können sowohl von Operationsvariablen als auch von Prozeßvariablen aufwärts und abwärts gezählt werden. Im Beispiel sehen Sie im ersten Bild das Aufwärtszählen durch eine Operationsvariable:

Im zweiten Bild erkennen Sie die Nutzung einer p-Variablen zum Aufwärtszählen im Vorwärts-Rückwärts-Zähler:

Weitere Einzelheiten zum Konfigurieren siehe: Counter-Projektierung [siehe Seite 204]

5.4 Vergleichsfunktion - Comparison

Allgemeines

Die Projektierung der verschiedenen Vergleichsfunktionen erfolgt im PRAP, wo eine Prozeßzustandsvariable zum Ausdruck bringt, daß eine Prozeßgröße wie Füllstand, Position oder Druck usw. einen bestimmten Wert erreicht, über- oder unterschritten hat. Das Vergleichsergebnis entspricht der Aussage Yes/No - ja/nein einer P-Variablen, die Sie adressieren (Input/Output) können oder dies dem Precompiler (USAGE: Local Static oder Local Temporary) überlassen, der dann eine lokale Merker-Variablen einsetzt. Der Grenzwert / Sollwert kann fest oder z.B. über die Kopierfunktion [siehe Seite 37] prozeßabhängig vorgegeben werden.



Anmerkung

In den Vergleichs-Ausdrücken sind die Daten einzutragen. Bedenken Sie bei der Angabe der Datentypen, daß Byte, Word und DWord nur für Gleichheit oder Ungleichheit zulässig sein können, weil ansonsten eine erforderliche Vorzeichenangabe fehlt! Kümmern Sie sich ansonsten nicht um die Anpassung der Datentypen, das besorgt - weitgehend - der Precompiler automatisch.

Die Vergleichsfunktion VGL dient der Verarbeitung der folgenden Operatoren (Rechenzeichen):

VGL kann sein =, <, >, <=, >= und <>

Es muß immer

Istwert VGL Sollwert

notiert werden, wobei der Istwert immer von einer Adresse gelesen wird. Sie können auch Zeit- oder Zählglied-Ausgänge (Timer.ET oder Counter.CV) als Istwerte nutzen. Der Sollwert kann eine Zahlenkonstante, auch ein Zeitwert, sein, kann aber ebenso von einer Adresse entnommen werden. Beachten Sie die Hinweise zu den SPS-Sprachen.

Sie dürfen folgende Ausdrücke vorgeben:

- Beide Seiten zu VGL können arithmetische Ausdrücke enthalten.
- Mehrere Vergleicherausdrücke können Sie mit AND oder OR oder XOR verbinden. Bitte jeden Ausdruck einklammern oder wenigstens ein Leerzeichen davor und danach. Am Ende der Formel ";" nicht vergessen.
- Geben Sie nur eine Vergleicherausdrücke an und keine zusätzlichen Berechnungen

Projektierungs-Ablauf

Führen Sie im PRAP den Cursor auf die "Condition", die die Vergleicherausdrücke beinhalten soll und klicken auf die rechte Maustaste, und danach erhalten Sie mit Klick auf "Data Access" (oder DreiMalKlick [siehe Seite 150]) das Formular:

- Es kann angenommen werden, daß Sie den PRAP schon so weit entworfen haben, daß nur noch die nichtbinären Funktionen, hier die Vergleicherausdrücke, eingetragen werden müssen. Daher tragen "Statement" und "Variable Usage" konkrete Begriffe. Während "Statement" ein technologienaher Text [siehe Seite 143] ist, kann "Variable Usage" durch "Input/Output" gekennzeichnet sein, falls die Instanzen eine Bitadresse angeben, oder durch "Logic Static" oder "Local Temporary", falls die Boolesche Variable lokal im Baustein verwendet wird, dann generiert der Precompiler eine Bit-Adresse im RAM.
- Durch Klick auf   sind die Daten der Booleschen Variablen definiert, ansonsten holen Sie das jetzt nach. Dabei können Sie sich an das Beispiel aus der Lampensteuerung [siehe Seite 110] halten. Damit ist die Boolesche Variable vorbereitet, das Ergebnis einer Vergleicherausdrücke zu tragen (Flag-Eigenschaft).
- Wählen Sie nun bei "Choose a Function" die Bezeichnung "Comparison". Danach öffnen Sie mit Klick auf "Formula nothing" und "Yes" (Neue Vergleicherausdrücke generieren) das Formblatt.



Ein Klick auf den Help-Button gibt Ihnen die notwendigen Hinweise zum Ausfüllen.

Ein Klick auf  öffnet die "Access Table". Dort können Sie neue Variable eintragen, falls für die Vergleichsfunktion erforderlich.

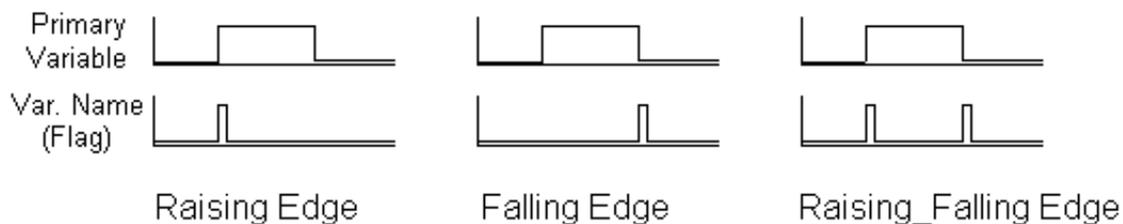
5.5 Flanken-Funktionen - Edge Detection

Allgemeines

Projektierbar sind steigende (rising), fallende (falling) oder sowohl ansteigende als auch fallende (rising_falling) Flanke (Edge) eines Binärsignales zu erkennen und auszuwerten.

Beim Wechsel des Wertes der auszuwertenden Variablen⁷ (Primary Variable) von "No nach Yes" handelt es sich um eine steigende Flanke, folglich bei "Yes auf No" um eine fallende.

Die betreffende "Ergebnis"-Variable (Variable Name (Flag)) ist stets eine P-Variable ("Condition"). Diese Ergebnis-Variable ist nur einen Arbeitszyklus der Ressource/Steuerungseinheit (z.B. SPS) lang true.



Projektierungs-Ablauf

Führen Sie im PRAP den Cursor auf die "Condition", die die Flanken-Verarbeitung beinhalten soll und klicken auf die rechte Maustaste, und danach erhalten Sie mit Klick auf "Data Access" (oder Doppelklick [siehe Seite 150]) das Formular "Process Variables". Bei "Choose a Function" können Sie drei Funktionen wählen:

- "Raising Edge"
- "Falling Edge"
- "Raising_Falling Edge"

Danach Klick auf "Formula"-Button und mit "Yes" das Erzeugen der Funktion bestätigen. Es öffnet sich das Formblatt "Edge Detection Configuration", wo Sie die auszuwertende Variable (Primary Variable) eintragen müssen. Mit Klick auf  öffnet sich die Variablen-tabelle, aus der Sie eine schon vorhandene Variable auswählen können oder eine neue eintragen. Mit Klick auf  erfolgt dann der Eintrag in das Feld "Primary Variable | Name".

Weitere Einzelheiten erfahren Sie unter Flankenauswertung. [siehe Seite 205]

⁷ Der Standard IEC 1131-3 beschreibt die Standard-Funktionen "Rising Edge Detector" und "Falling Edge Detector" mit dem Eingang "CLK Input" und Ausgang "Q"

5.6 Kopierfunktion - Copy

Die Projektierung der Übertragungs- oder Zuweisungs-Funktion geschieht ausschließlich auf Anweisung (=Befehl) einer Operationsvariablen. Die gewünschte Kopierfunktion wird daher nur in den Operationen ausgeführt, in denen der Befehl markiert ist. Sie können diese Boolesche Variable selbst adressieren oder überlassen dies dem Precompiler, indem Sie eine lokal statische oder temporäre Verwendung (USAGE) wählen. Sie können 1-, 8-, 16- oder 32-Bit-Größen auf Merker oder auf Ausgänge übertragen. Sie können auch konstante Werte auf Merker oder Ausgänge schreiben. Wählen Sie gewissenhaft die zum Datentyp passende Zahlendarstellung, die Sie als Initialwert eintragen.

Wenn Sie z.B. eine 8bit- auf eine 32bit-Variable übertragen wollen, brauchen Sie sich nicht um die Datentyp-Anpassung zu kümmern, das erledigt automatisch der Precompiler. Damit steht Ihnen die Möglichkeit offen, Zeiten, Grenzwerte, Zählgrößen, Sollwerte für Vergleiche, Regler usw. situationsabhängig [siehe Seite 183] mit Hilfe entsprechend lautender Operationsvariablen zuzuweisen.

Natürlich müssen Sie darauf achten, daß die Empfänger dieser Nachricht von der gleichen Adresse ihre Werte abholen. Beachten Sie ferner, daß z.B. die Soll-Zeitwertausgabe in einem anderen PRAP stehen darf als das Zeitglied, das eine solche Vorgabe benötigt.

Einzelheiten siehe unter Kopier- und Arithmetikfunktionen. [siehe Seite 205]

5.7 Arithmetik-Funktionen Caculate(multiple) und Calculate(single)

Mit dem Begriff "Arithmetik-Funktionen" sind folgende Verknüpfungsoperationen gemeint, die auch verwendet werden dürfen:

- Zweistellige Operationen:
Addieren (+), Subtrahieren (-), Multiplizieren (*), Dividieren (/)
Bitweise UND-Verknüpfung (AND), ODER-Verknüpfung (OR)
- Einstellige Operationen:
Absolutwert eines Operanden ABS (für Integer- und Real-Typen)
- Bitweise Schieben oder Rotieren eines Operanden rechts/links um eine Anzahl N Bit.
Beispiel:
SHL(IN:=Operand,N:=Anzahl)

Aber auch Potenzieren (**) und trigonometrische Funktionen werden in zukünftigen Versionen zugelassen sein. Sprachabhängig [siehe Seite 7] bleiben natürlich bestimmte Einschränkungen. Vergewissern Sie sich also, ob eine von Ihnen gewünschte Funktion auch auf der Ressource durchführbar ist.

Ein arithmetischer Ausdruck wie z.B. $E := a + b * c - SHR(IN:=d,N:=Anz)$ wird mit $Anz=9$ wie folgt berechnet:

- Zuerst alle Rotier- und Schiebefunktionen. Also SHR wird berechnet und auf d gespeichert. Die Anzahl, hier 9, darf auch auf einem Speicherplatz stehen. Das Ergebnis wird vorzeichenbehaftet verarbeitet, weil alle Operanden eines solchen Ausdrucks diese eigenschaft haben müssen.
- Anschließend folgt die "Multiplikation", also $b*c$.
- Danach erfolgt die Addition des Zwischenwertes mit a und schließlich die Subtraktion minus d.
- Der Ergebniswert wird auf dem Ergebnis- oder Ziel-Speicherplatz E abgelegt.

**Tip**

Die sogenannte Punkt-Vor-Strich-Abarbeitung einer arithmetischen Formel können Sie durch Klammern, also "(" und ")", verändern. Klammerausdrücke werden vorrangig berechnet.

Um **Datentypen-Anpassung** brauchen Sie sich selten zu kümmern, weil der Precompiler das meist selbständig durchführt. Es werden allerdings, wie schon erwähnt, nur vorzeichenbehaftete Datentypen angenommen, das sind SINT, INT, DINT und REAL. Die Ausnahme sind reine Zuweisungen bzw. Kopierfunktionen (z.B. A := D), Schiebe- und Rotierfunktionen und andere Bitmanipulationen wie AND, OR usw. Sollte die Anpassung nicht automatisch erfolgen können, dann erhalten Sie eine Mitteilung.

Das Ausführen einer arithmetischen Funktion bestimmt eine Operationsvariable [siehe Seite 176]. Sobald sie in einer Operation [siehe Seite 174] true ist, erfolgt der oben erklärte Ablauf so lange (bei jedem SPS-Zyklus), bis der Wert der Operationsvariablen wieder false ist, also etwa in der folgenden Operation. Daher "Caculate(multiple)", also "berechne mehrfach".

Falls Sie diese Boolesche Variable mit Local Static angeben, erzeugt der Precompiler eine lokale Variable selbst.

Sollte der Ergebnis-Speicher auch im arithmetischen Ausdruck erscheinen,

z.B. $a := a + 1$,

dann muß die Berechnung innerhalb genau eines Bearbeitungszyklus' der SPS erfolgen, weil ansonsten das Ergebnis die Anzahl der SPS-Zyklen am Ende der betreffenden Situation [siehe Seite 183] enthielte.

Zum Beispiel sei $a=5$ beim Aufruf der Additionsfunktion (ist gleich Übergang in diese Situation), und es sollen 60 Bearbeitungszyklen während der Dauer der Situation angenommen werden, dann ist am Ende $a=65$ anstelle $a=6$. Sie können solche Ausdrücke zum Zählen verwenden anstelle Zählglieder zu projektieren. Aber eine einmalige Berechnung müssen Sie mit "Calculate (single)" wählen. Es sei denn, Sie wollen den Zähleffekt nutzen.

Weitere Details unter Kopier- und Arithmetikfunktionen. [siehe Seite 205]

Kapitel 6

Datenverwaltung

6.1 Die Interbase-Datenbank

Zugang zum Inhalt der Datenbank erhält im Prinzip nur der Administrator über die Interbase Console. Dem User eines Projektes in SPAS sind aber Zugriffe in mehrerer Hinsicht ermöglicht:

- Das Formblatt Database Options [siehe Seite 206] gewährt
 - das Neu-Anlegen eines Projektes
 - das Wiedereinloggen
 - das Arbeiten auf dem lokalen Server oder einem Server über das Internet
 - das Erzeugen eines Datenbank-Backup als komprimierte Datei
 - das Rückladen einer Backup-Datei
- Anzeige und Editieren aller in der Datenbank eingetragenen Variablen [siehe Seite 43]
- Zu jedem Member [siehe Seite 172] gehörige Daten und Variable sind in Arbeits- bzw. Formblättern editierbar (Hilfetexte siehe dort)

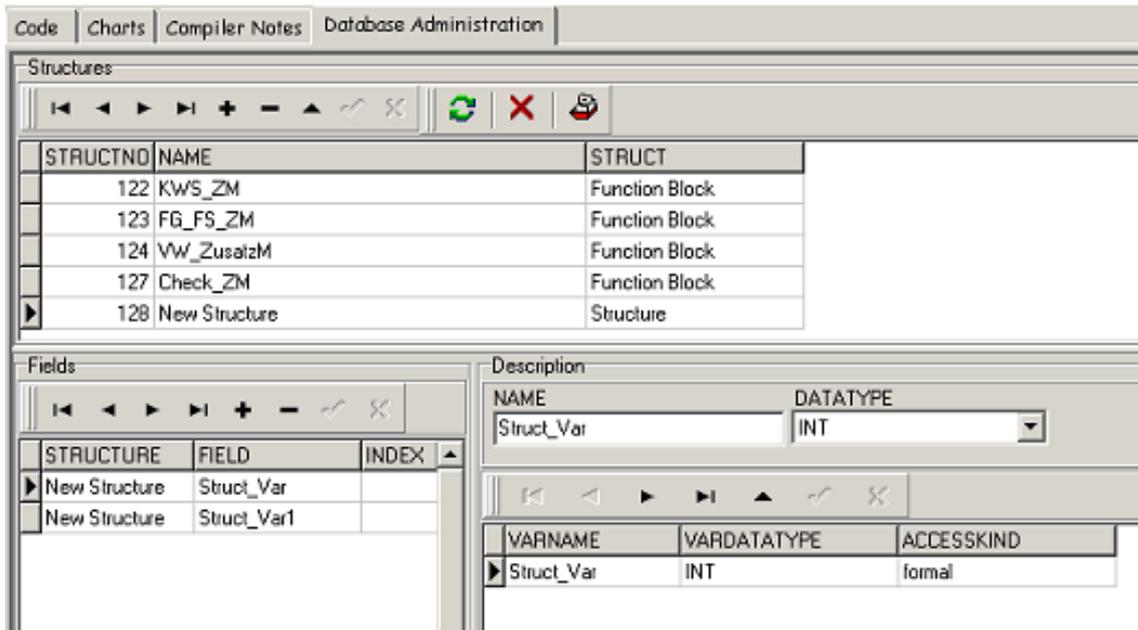
6.2 Entwurf strukturierter Datentypen - User Defined Types (UDT)

Die SPAS-Version 1 gestattet es, neben den Datentypen wie BOOL bis REAL oder TIME, TON (Timer mit Einschaltverzögerung) usw. selbst definierte, strukturierte Datentypen [siehe Seite 189] nach dem Muster der Datenbausteine zu definieren und in einer Ressource einzuführen.

Diese Möglichkeit wurde unter dem Aspekt der besseren Nutzung der S7-CPU vorläufig geschaffen, wobei es um die Nutzung in den Members "PROGRAM TYPE" als Block von Aktual-Parametern geht. In den Sprachen C und C++ können diese ebenfalls verwendet werden.

Nach dem Klick auf  werden Sie aufgefordert, einen Bezeichner des neuen Datentypes (Identification) einzutragen. "UDT Settings" bedeutet bei SIEMENS STEP 7 nur, eine Zahl einzugeben, die Sie selbst in der Ordnung der UDT und anderer Datenbausteine zu wählen haben. Bei C oder C++ fehlt dieses Sheet. Danach "OK" klicken, wodurch dieser Datenbaustein in die Ressource eingefügt ist.

Er besteht aus dem Bezeichner der Struktur (NAME) und dessen Feldern (Field) in der Form NAME.Field



Gehen Sie nun wie folgt weiter vor:

- Markieren der Ressource, in die UDT eingefügt wurde. Danach Klick auf "View | Structures"
- Das geöffnete Panel enthält die Bereiche "Structures", "Fields" und "Description".
- Suchen⁸ Sie im Bereich Structures nach dem neuen Datenbaustein. Dabei werden Sie erkennen, wie alle bisher im Projekt enthaltenen Bausteine hier "geführt" sind.
- Kopieren Sie die von Ihnen gewählte Bezeichnung unter "NAME" in den Bereich "Fields" unter "Structure". Automatisch steht unter "Field" eingetragen: Struct_Var...
- Im Bereich "DESCRIPTION" können Sie "NAME" und "DATATYPE" editieren. Im Beispiel wurde einfach "1" angehängt. Nach der Änderung Klick auf .
- Das nächste Feld des Datenbausteins "New Structure" fügen Sie mit Klick auf  ein.



Anmerkung

Die Struktur-Typen sind intern wegen der Rückdarstellbarkeit (Online Viewer) wortweise organisiert. Achten Sie deswegen bitte darauf, immer Ganzzahlige von 16 Bit anzulegen.

- Zum Schluß müssen Sie die eingegebenen Variablen in die Typ-Struktur mit Klick auf  übernehmen.

Das Instanzieren der StrukturVariablen [siehe Seite 40] erfolgt in dem PROGRAM TYPE, in dem Sie diese Datenstruktur benötigen.

6.3 Instanzieren strukturierter Datentypen

Im Gegensatz zur Instanzierung von PRAP-Typen bleiben die Variablen des Strukturtyps erhalten wie bei RECORD-Strukturen, nur der Name, der Bezeichner wird neu, so daß Sie von einem Struktur-Typ wieder viele Instanz-Strukturen bilden können.

Das weitere Vorgehen erfahren Sie (UDTInstance [siehe Seite 209]).

⁸ Die aktuelle Zeile der Tabelle ist links am schwarzen Dreieck zu erkennen, dorthin den Cursor führen und Klicken. Damit ist diese Struktur markiert. Nun mit Aufwärts oder Abwärtstaste die Tabelle durchblättern.

6.4 Variable

Jede Variable hat einen "NAME" als Identifikator, dem wegen der begrenzten Anzahl Zeichen noch ein "STATEMENT", zuweilen auch "DESCRIPTION", mit bis zu 128 Zeichen (sprachabhängig!) beigegeben wird und weitere Eigenschaften:

- **MEMBER:**
Zuordnung zu einer Programm-Organisations-Einheit (POE [siehe Seite 177]). Eine Variable erhält damit ihren Bezug, z.B. MEMBER.NAME
- **DATA TYPE:**
Alle einfachen Datentypen wie z.B. BOOL, INT, REAL, TIME usw., aber auch FC, FB, P(rogram), UDT [siehe Seite 39] (User defined Types)
- **ADDRESS:**
KEIN Eintrag bei USAGE von Constant und Input bis Input/Output, weil Parameter "formal" dazugehört.
KEIN Eintrag bei USAGE Local, weil Maschinen-Code-Compiler automatisch RAM zuweist.
Eintrag ERFORDERLICH bei USAGE Address ..., aber nur Adress-Zahlen. Bei Booleschen Adressen Byte.Bit mit Bit = 0 bis 7.
- **USAGE:**
Constant, Input, Output, Input/Output, Local Static, Local Temporary
Address Input, Address Output, Address Memory
- **INITIAL VALUE:**
Anfangswert bei erstmaligem Start der CPU, einer Ressource.
Feste Wertzuordnung bei USAGE = CONSTANT.
- **POINT OF AUTOMATION:**
Bezeichner der technologischen Aufgabenstellung aus dem Technologie-Schema, z.B. EMSR-Stellen nach DIN 19227-1.
Teilprozeß-Koppelungen mit Schlüsselwort KOP und nachfolgender Angabe des absendenden TP und Angabe aller empfangenden TP. Beispiel: KOP 12 nach 2,34,17, falls die TP im Prozeß-Zerlegungs-Graph oder in der Funktions-Block-Darstellung Nummern erhalten haben.
- **PARAMETER:**
Formal für USAGE = Input bis Input/Output, formal oder actual bei Local Static oder Temporary, actual bei Address Input bis Address Memory.

Die Gesamtansicht aller Variablen erhalten Sie im SPAS_Manager mit Klick auf

- "View|Variables" oder
- "View|All Variables" und anschließend auf .

Klicken Sie "View|Variables", erhalten Sie eine auf die Variablenart bezogene Ordnung (Use_Var_list [siehe Seite 42]). Hier können Sie auch Variablen-Bezeichner (Statement = Varlist Comment) nachtragen, z.B. an Stelle von "Global Variable" bei Aktual-Variablen, oder auch ändern.
Mehr zu Listen: DatabaseReport [siehe Seite 43]



Tipp

Wollen Sie eine Variable löschen, dann beginnen Sie es hier (rechte Maustaste auf markierter Zeile). Die endgültige Beseitigung aus der Datenbank ist anschließend entweder bei View|All Variables [siehe Seite 43] möglich, oder Sie öffnen das zugehörige Member, markieren die betreffende Variable und löschen diese in der Variablen-tabelle (Access table) mit Klick auf



6.5 Benutzen der Variablen-Anzeige und -Liste

Die Variablenanzeige

- gestattet keine Ergänzungen
- gestattet die Übersicht unter verschiedenen Filtermöglichkeiten
- ermöglicht einige Änderungen
- Erlaubt insbesondere Statements zu in der Ressource oder Programm-Typen eingeführten Aktualparametern zu schreiben

Die Liste enthält zwei Felder, die linke Liste enthält alle Variablen, eventuell nach Klick auf .

Daneben ist  zum Löschen aller nicht im Projekt zugewiesenen Variablen. Eine Variable gilt als zugewiesen, wenn diese in einem Member deklariert ist. Bitte überzeugen Sie sich VOR dem Klick auf diesen Button, ob Sie nicht noch eine Variable benötigen.

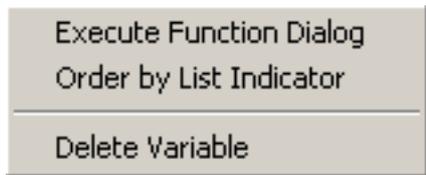
Die rechte Seite zeigt weitere Daten der Variablen.

Bedienen der Filterfunktion:

- Allgemein: Nur eine Option auswählen
Kombinierbar: Kind/Function & Member
- Suchen Sie unter Filter Values das Zutreffende aus. Variablen zu Instanzen werden nicht angezeigt, da dort keine deklarierbar sind!
"Kind" bedeutet die Art der Variablen, also Prozeß-Zustands-Variable [siehe Seite 183] oder Operations-Variable [siehe Seite 176]. Lokal deklarierte sind von der Art "Variable".

Die Datenfelder

- Kümmern Sie sich nicht weiter um die Felder, in denen Indizes angezeigt werden, die im Zusammenhang mit der Interbase- Datenbank stehen.
- Varlist Comment
ist ein Feld, in dem Statements bzw. Comments oder Descriptions erscheinen, die im Laufe der Projektentwicklung entstanden oder auch nicht. Hier können Sie den Text ändern [siehe Seite 143], falls erforderlich überhaupt erst eintragen.
- Jede Änderung ist entweder rückgängig durch Klick auf  zu machen oder in die Datenbank einzutragen durch Klick auf .
- Empfehlung: Keine weiteren Änderungen hier durchführen! Ändern Sie dort, d.h. in dem MEMBER, wo die Variable (=NAME)zugeordnet ist.
-



weitere Auskunft über ihre funktionelle Verknüpfung geben

oder gelöscht werden.

6.6 Alle Variablen anzeigen und bearbeiten

Die Gesamtansicht aller Variablen erhalten Sie im SPAS_Manager mit Klick auf

- "View|Variables" oder
- "View|All Variables" und anschließend auf , falls keine Anzeige.

Klicken Sie "View| All Variables" erhalten Sie eine Sicht auf das Member bezogen:

Im Feld "MEMBER" sehen Sie, wo die Variablen zugeordnet sind bzw. deklariert wurden, und im Feld "USAGE" wird angezeigt, wie diese Variable verwendet wird. Falls "Multiple Choice" bei "MEMBER" eingetragen ist, dann finden Sie unterhalb der Tabelle angegeben, wo überall zugeordnet. Im Feld "COUNTER" erkennen Sie, wie oft die Variable zugeordnet wurde.



Anmerkung

Nutzen Sie die Möglichkeit, zu jeder in der Tabelle markierten Variablen mit rechter Maustaste und Klick auf "Search member of Variable declaration" oder "Search instances using the variable" mehr zu erfahren.

- Sie können alle erforderlichen Korrekturen nur in den Feldern vornehmen, NICHT IN DER TABELLE.
- Fehlende Variable können Sie nach Klick auf  hier eintragen. Die Zuordnung jeder Variablen kann nur im Member erfolgen. Die Felder "MEMBER" und "USAGE" bleiben also leer. Bis zu dieser Zuordnung bleibt auch offen, ob die Variable ein formaler oder aktueller Parameter ist. Dies ist wichtig zur Zuordnung der Variablen zum Typ bzw.zur Instanz. Auf jeden Fall nach der Korrektur  anklicken!
- Überflüssige, nirgends zugewiesene Variable entfernen Sie entweder mit Klick auf  oder einzeln nach dem Markieren mit Klick auf . Vorher bitte View|Variables [siehe Seite 41] anklicken und dort löschen.

6.7 Datenbank-Report

Variablen des markierten Members werden aus der Datenbank druckreif angezeigt, sind jedoch nicht veränderbar. Bitte beachten Sie die Bedeutung der Symbole am oberen Rand des Fensters.

Dort finden Sie hinter dem Wort "Kind" drei mögliche Listen-Typen:

- Process Variable:
Die in die "Conditions" eingetragenen Namen und Statements, alle sind Boolesche Variable.
- Operation Variable

Die in die "Operations" eingetragenen Namen und Statements, alle sind Boolesche Variable.

- Variable
 - Hier sind alle anderen Variablen aus Funktionen enthalten, die mit den Process oder Operation Variables verknüpft sind.
 - Es werden auch alle Aktual-Parameter in dieser Liste geführt. Deklarationen in PROGRAM oder RESOURCE enthalten nur diese Liste. Dies sind dann "Global Variables".
 - Strukturierte Datentypen sind ausschließlich in der Liste "Variables" enthalten.

Falls Sie Fehler feststellen, dann brauchen Sie die Anzeige nicht zu schließen, Sie korrigieren entweder im zugehörigen Member oder in der Gesamtansicht [siehe Seite 43] aller Variablen.



Warnung

Der Druckvorgang funktioniert nur für "Current Page". Stellen Sie also die zu druckende Seite zentral im Bildschirm und klicken auf "OK". Dies müssen Sie leider bei allen Seiten wiederholen.

Zuweilen werden mehr Seiten als nötig angezeigt. Achten Sie bitte auf den Inhalt der Seite, die zum Drucken dem Drucker übergeben wird.

6.8 Import einer Variablen-tabelle (Excel-tabelle)

Wenn Sie für die Prozeßvariablen eine Excel-tabelle anlegen, können Sie diese importieren, mehr dazu: ImportVarTab [siehe Seite 198]

6.9 Back-Up-Datenbank

Die im Verlaufe der Projektentwicklung entstehende Datenbank kann für Zwecke der Archivierung in komprimierter Form gespeichert werden. Bei Wiederverwendung muß mit dem Rückladen wieder eine normale Datenbank (Back Up and Restore [siehe Seite 163]) erzeugt werden.

Kapitel 7

Prozeß-Ablauf-Netz (PAN)

7.1 Die Grundlagen

(Das Tool PAN-Editor ist noch nicht verfügbar!)

Das Prozeßablaufnetz (PAN) ist ein Bedingungs-Ereignis-Netz (BEN) und damit ein Beschreibungsmittel, bzw. ein Modell auf Grundlage des Petri-Netzes.

Das PAN ist bei großen oder schwer überschaubaren Prozessen als Vorläufer der Prozeßerlegung zu empfehlen. Insbesondere Chargen- bzw. Batch-Prozesse mit mehreren parallelen (simultanen) technologischen Abläufen lassen sich gut modellieren. Sowohl Binär-Steuerungs- als auch Stabilisierungs-Anforderungen, also Regelungen, lassen sich im PAN darstellen.

Ziele des Prozeßablaufnetzes:

- Modellierung des ungestörten Gesamtprozeßablaufes
- Unterstützung bei der Zerlegung des Prozesses in Teilprozesse
- Unterstützung bei der Überführung der Steuerungs-Teilprozesse in PRAP-Grundstrukturen, die die im Gesamtprozeßmodell vorhandenen Kopplungen beinhaltet

Es gilt beim PAN:

Ein Prozeßschritt ist mathematisch betrachtet eine Sequenz von Bedingung - Ereignis - Bedingung bzw. Platz - Transition - Platz .

Den Inhalt von Prozeßschritten bilden Abschnitte des Prozeßablaufes und erhalten eine dementsprechende Benennung. Ein Prozeßschritt beginnt, sobald der erste Platz (Eingangsplatz) markiert wird. Der Prozeßschritt benötigt Prozeßzeit, an deren Ende die Markierung dem zweiten Platz (Ausgangsplatz) zugewiesen wird. Dann gilt er als abgeschlossen.

Für die Transitionen zwischen den Prozeßschritten gilt die Schaltregel, daß alle Vorprozeßschritte ihren Ausgangsplatz markiert haben müssen, und bei allen Nachprozeßschritten kein Eingangsplatz markiert ist.

Die Transitionen können also erst schalten, wenn alle Vorprozeßschritte abgeschlossen sind, und Nachprozeßschritte dürfen entweder von einer anderen Transition keine Marke erhalten haben oder müssen ebenfalls abgeschlossen sein.

Muß ein Konflikt gelöst werden, oder sind andere technologische Zusatzbedingungen zu erfüllen, damit eine Transition schalten kann, gibt es die Bedingungsschritte, die ihre Markierung erhalten, sobald die beschriebene bzw. genannte Bedingung erfüllt ist.

Ferner ist die Inhibitorkante nutzbar, die zwar keine Marke transportiert, aber der Transition die Schalterlaubnis entzieht, solange der betreffende Ausgangsplatz eines Vor-Prozeßschrittes eine Marke trägt.

Die Anfangssituation des Prozeßablaufnetzes wird durch eine Anfangsmarkierung festgelegt.

Kapitel 8

Prozeßzerlegung - Editor

8.1 Editor zur Entwicklung des Prozeßzerlegungsgraphen (PZG-Editor)

Diesen Editor gibt es noch nicht, Sie müssen also ein beliebiges Graphikprogramm benutzen, ein Beispiel dazu gibt es im Lehrbeispiel. [siehe Seite 47] Die Grundlagen der Prozeßzerlegung sind im Buch Entwurfsprozeß [siehe Seite 17] enthalten.

Ermitteln Sie zu jedem Teilprozeß die Eingangs- und Ausgangsvariablen und schreiben Sie diese z.B. in eine Excel-Tabelle, ein Beispiel finden Sie hier. [siehe Seite 82]

8.2 PZG zum Lehrbeispiel

Die Entwicklung vom Anfang bis zum Ende des Lehr-Beispiels ist an den folgenden PZG zu erkennen:

Anfangs-PZG⁹

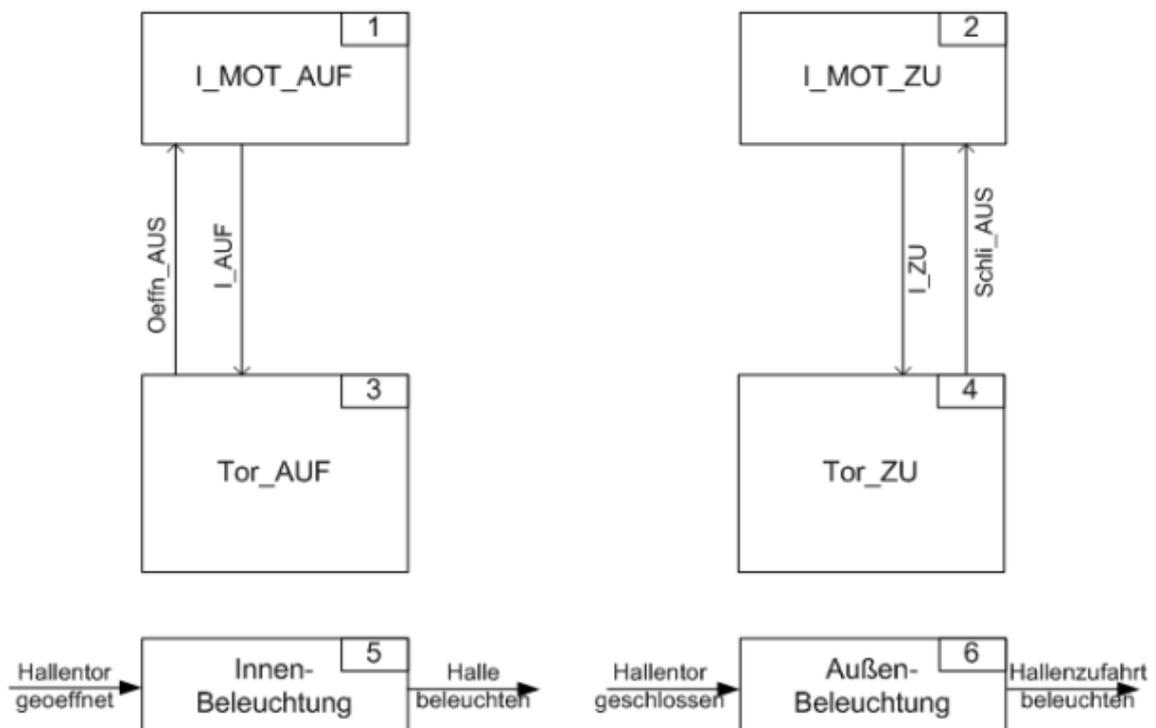
Zweiter Schritt der PZG-Entwicklung¹⁰

Unvollendeter PZG¹¹

9



10



Im Laufe der Prozeßzerlegung gewinnt man z.B. nicht nur die Teilprozesse sondern auch die Zuordnungen der TP-Bezeichnungen, die Anschlußbezeichnungen für die Prozeßvariablen an die SPS und die Koppelvariablen, die nur im RAM konkret als Merker existieren. Ein Konzept der Zusammenstellung der Daten sehen Sie in der (Variablenliste [siehe Seite 48]).

8.3 Variablenlisten nach der Prozeßzerlegung

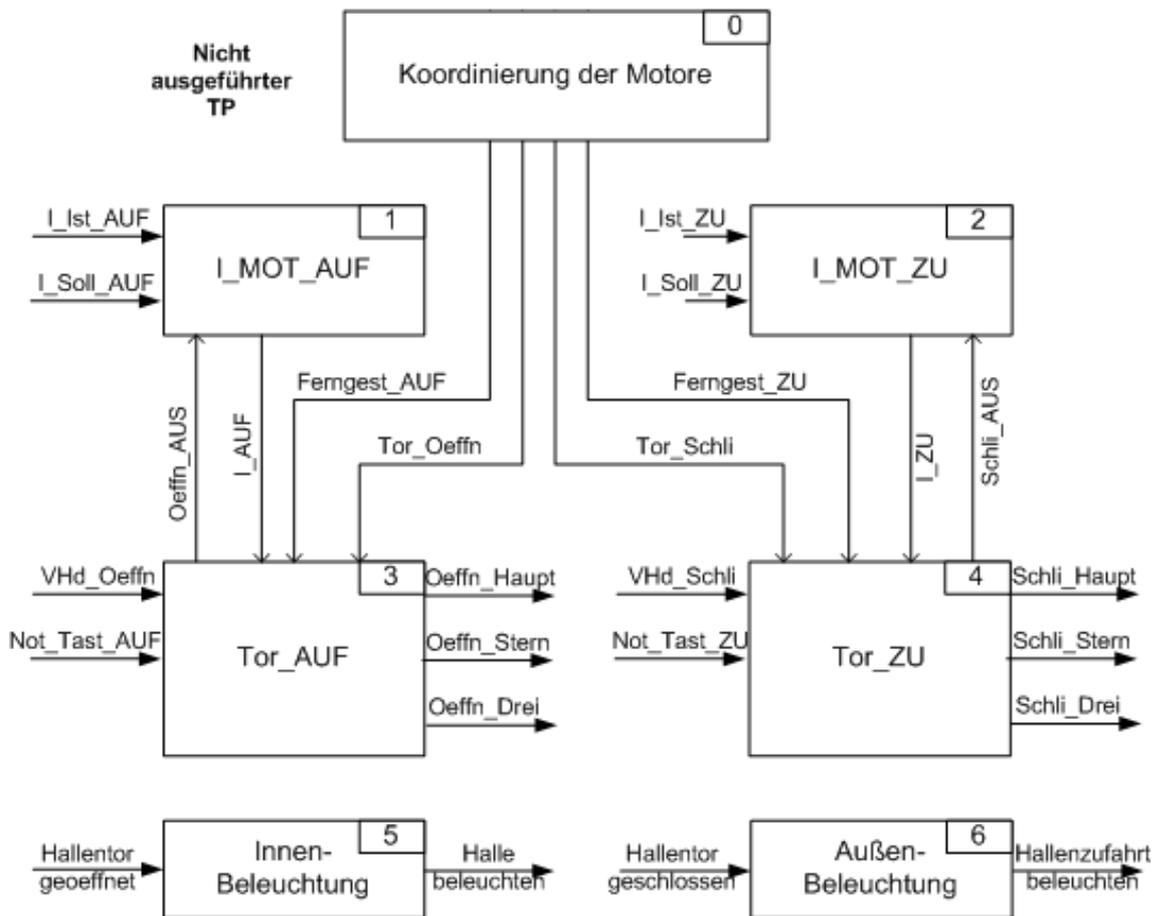
Die erste Variablenliste

enthält zum Lehrbeispiel [siehe Seite 73] in einer nach Teilprozessen geordneten Excel-Tabelle 1 [siehe Seite 201] Eingetragungen zu:

- A: Namen (Identifikation) der Aktualparameter
(Sich wiederholende Namen sind eingeklammert, wobei bei Koppelvariablen vorzugsweise die "Empfänger" eingeklammert wurden)
- B: Bezeichner der Aktualparameter - Actual Statement
- C und D: direkte Adressen, getrennt nach Präfix gemäß STEP 7 und [Byte].[Bit] gemäß STEP 7, um auch die Übersetzung nach C oder C++ zu ermöglichen (bis auf PEW!!)
- E: Datentyp der Aktualparameter
- F: Speicherort bzw. Verwendung der Aktualparameter

¹¹ Unvollendet bedeutet, die Koordinierung ist nicht als TP ausgeführt.

Die TP tragen einen als Identifikation geltenden Namen, wobei die PRAP-Typen mit ihrem Namen mitverwendet wurden. Außerdem sind Nummern in die TP eingetragen worden, um die EMSR-Stellen einfacher bezeichnen zu können. Die TP-Nummern stehen nach dem EMSR-Zeichen in der Variablenliste [siehe Seite 48].



Prozeß-Zerlegungs-Graph zum Lehrbeispiel
Lampen-Stern-Dreieck-Schaltung

- G: EMSR-Stellenbezeichnung nach DIN 19227. Die EMSR-Stellen-Nummern wurden aus TP-Nummer gemäß PZG¹² und einer laufenden gebildet.
Koppelungen sind mit "KOP" gekennzeichnet und benennen "Sender" mit nach und "Empfänger" mit von.

Das sind die sechs Daten, die auf eine Ressource oder ein Programm bezogen importiert werden können: Markieren Sie das Member, Klick auf rechte Maustaste zeigt Ihnen im Menue den Text: Import Variable List [siehe Seite 198]

Alle weiteren Informationen, wie die Zugehörigkeit der Variablen zum Teilprozeß usw. (Spalten H bis Ende), müssen bei der Instanziierung [siehe Seite 126] berücksichtigt werden. Außerdem liegen sie am Beginn der Prozeßanalyse [siehe Seite 14] noch nicht vor und wurden nachträglich in die Tabelle ergänzend eingetragen.

In dieser Tabelle sind Variable eingetragen, die erst mit dem PRAP-Entwurf, d.h. der funktionellen Verarbeitung der Variablen erforderlich werden. Dazu gehören Variable des Timers (T1 und T2) und des Vergleichers (PEW250 und PEW252). Diese sind im Laufe der PRAP-Entwicklung einzuführen und beim Bilden der Instanzen in PROGRAM oder RESOURCE zu deklarieren.

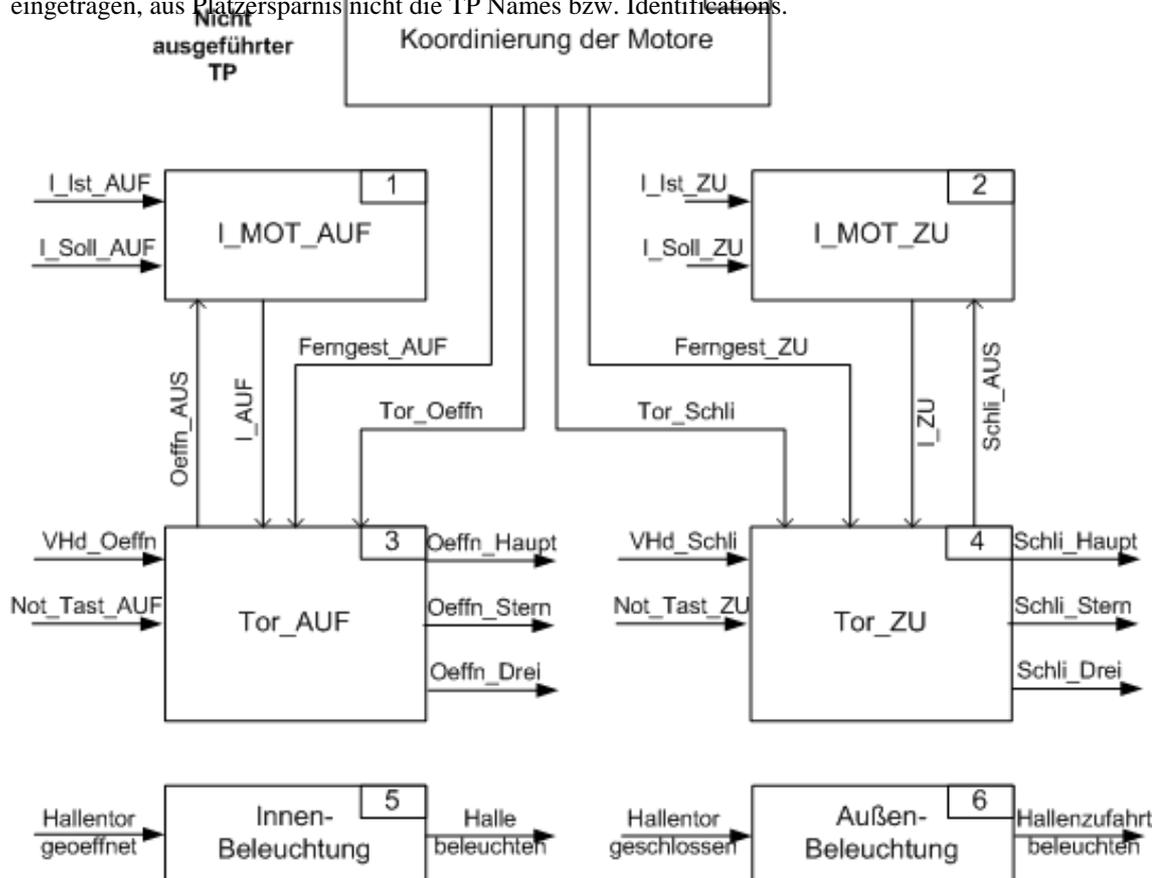


Anmerkung

Koppelvariable sind im Beispiel sowohl mit direkten Adressen (Address Memory) als auch mit internen (Local Static) eingetragen, um den Unterschied zu zeigen.

Die zweite Variablenliste

enthält zum Lehrbeispiel [siehe Seite 73] in einer nach Adressen geordneten Excel-Tabelle 2 [siehe Seite 202] Eingetragungen zu den Spalten sind mit der ersten Tabelle identisch. Aber jeder Name erscheint nur einmal und in der Spalte "H" sind alle TP aufgeführt, in denen "Name" vorkommt. In der Spalte "G" sind die Koppelungen von Teilprozeß nach Teilprozeß mit den im PZG¹³ enthaltenen Nummern eingetragen, aus Platzersparnis nicht die TP Names bzw. Identifications.



Prozeß-Zerlegungs-Graph zum Lehrbeispiel
Lampen-Stern-Dreieck-Schaltung

Kapitel 9

Projekt-Editor

9.1 Projekt-Editor

Der Projekteditor dient dem Entwickeln und Aufbau eines neuen Projektes [siehe Seite 127] ebenso wie der Pflege eines vorhandenen. Das Projekt ist in einer Interbase-Datenbank vollständig enthalten.

Der entstehende oder vorhandene Projektbaum [siehe Seite 182] gliedert sich in Knoten, auch Member [siehe Seite 172] genannt, die je nach Zuordnung in verschiedene Ebenen [siehe Seite 150] eingetragen werden. Viele Begriffe sind aus der Norm IEC 1131-3 entnommen und werden in diesem Sinne auch verwendet.

Es stehen folgende Buttons zur Verfügung:



Beachten Sie die Hinweise des SPAS-Managers zu jedem Button in den Anzeige-Fächern der unteren Zeile!

In der ersten Reihe finden Sie die Menüs

File??? | Edit??? | Project??? | View??? | Run??? | Help???

Ferner erhalten Sie auch mit der **rechten Maustaste** entsprechende Angebote zur weiteren Bearbeitung des Projektes [siehe Seite 161], auch den Übergang zum PRAP-Entwurf oder dessen Bearbeitung. [siehe Seite 61]

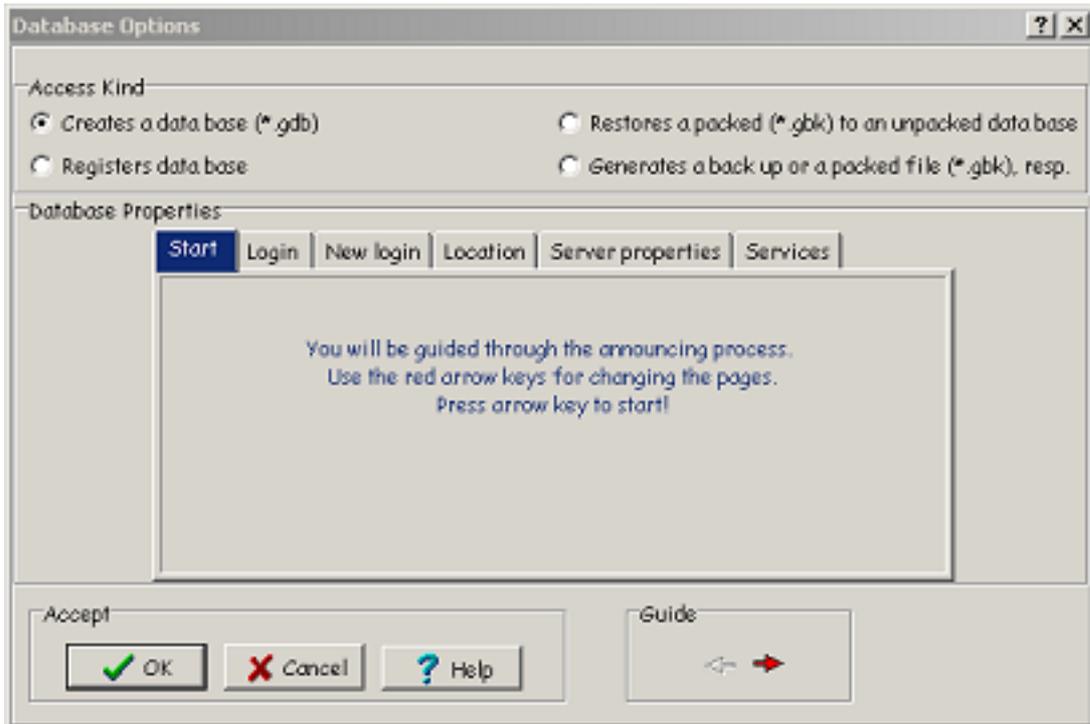
Weitere Themen des Projekteditors:

-  Neues Projekt anlegen [siehe Seite 51]
-  Angelegtes Projekt weiterbearbeiten [siehe Seite 53]
-  Konfiguration einfügen [siehe Seite 55]
-  Resource einfügen [siehe Seite 55]
-  Strukturierte Datentypen einfügen [siehe Seite 39]
-  Task einfügen/bearbeiten [siehe Seite 55]
-  PRAP-Typen einfügen/bearbeiten [siehe Seite 56]
-  Funktion einfügen [siehe Seite 58]
-  PROGRAM einfügen/bearbeiten [siehe Seite 58]
-  INSTANCE einfügen/bearbeiten [siehe Seite 59]
-  Funktions-Aufruf???
-  PRAP aus einer Bibliothek in die Ressource einfügen [siehe Seite 210]
-  PRAP aus dem Projekt in eine Bibliothek einfügen [siehe Seite 210]
-  Beispielprojekt Lampen- und Stern-Dreieck-Schaltung [siehe Seite 73]

9.2 Neues Projekt anlegen

Beachten Sie den Abschnitt Wichtige Empfehlungen. [siehe Seite 5]

Bevor Sie ein neues Projekt beginnen, legen Sie sich ein Verzeichnis (Directory, Ordner) dafür an. Sobald Sie das Icon  oder "File | New..." anklicken, erscheint das Formblatt

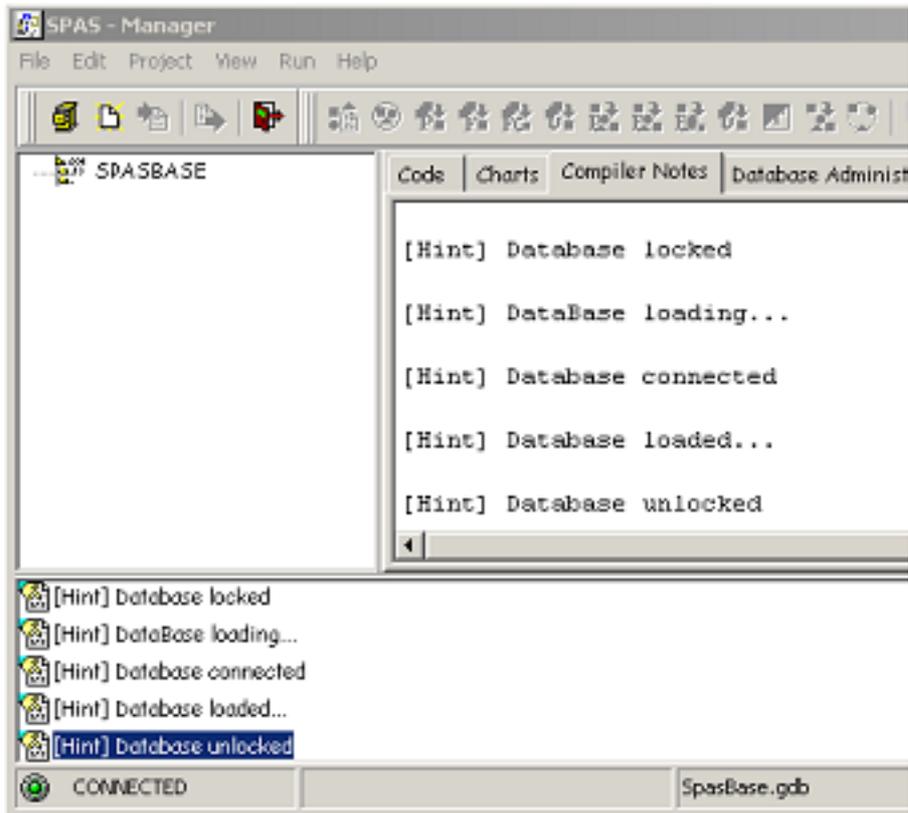


Verwenden Sie zum Ausfüllen die Hilfe über oder den Help-Button.

Siehe auch Database Options [siehe Seite 206].

Nach dem Ausfüllen den OK-Button anklicken oder Enter-Taste betätigen.

Jetzt haben Sie ein Projekt angelegt, nun müssen Sie es öffnen [siehe Seite 53], um damit arbeiten zu können. Haben Sie alles erfolgreich getan, können Sie den Text "Database unlocked" lesen und erkennen im Fenster des Projekteditors den ersten Knoten des Projektes:





Tipp

Im Meldungs Fenster (rechts vom Projektbaum oder unten) können Sie mit Klick auf die rechte Maustaste verschiedene Handlungen ausführen, also z.B. die Meldungen löschen.

Projekt weiter bearbeiten [siehe Seite 53]

Zurück zu Projekt-Editor [siehe Seite 51]

9.3 Öffnen, Schließen und Löschen eines Projektes

Das **Öffnen eines Projektes** ist auf mehreren Wegen möglich:

- Beachten Sie das Icon . Entweder Sie klicken es aus der Button-Leiste an oder nach Klick auf "File". Sie erhalten ein "ALIAS"-Fenster zur Auswahl und müssen das gewünschte, also auch ein neu angelegtes Projekt markieren (Cursor hinführen und Klick).
Angezeigt werden nur registrierte Projekte!
- Es gibt noch die Möglichkeit bei "File | Reopen", also die Wieder-Eröffnung schon einmal geöffneter und registrierter Projekte. Nach Betätigen der Enter-Taste oder Klick auf "OK" öffnet sich das Login-Fenster, in das Sie das beim Anlegen des Projektes vereinbarte Passwort eingeben müssen.
- Ist ein Projekt nicht registriert, dann müssen Sie  anklicken, dort gibt es die Möglichkeit der Neu-Registrierung oder des Neu-Ladens. Folgen Sie dem dortigen Hilfetext.

Zum **Schließen eines Projektes** markieren Sie den Projektknoten. Der Projektbaum beginnt mit dem Symbol , genannt Projektknoten. Danach Klick auf "File" in



und Klick auf / Close.

Das **Löschen eines Projektes** bedeutet zum einen, das Datenbankfile (GDB-File) konventionell, also mittels Explorer, zu löschen. In der Registratur für SPAS ist es noch eingetragen, was durch Klick auf  (Projekt Alias) oder auch "File" + "Reopen" sichtbar wird. Im Fenster "Projekt Alias" markieren Sie das Projekt, betätigen die rechte Maustaste und "Unregister Database" können Sie auf einem Button lesen, den Sie anklicken müssen.

Sie brauchen das GDB-File auch nicht zu löschen, aber Sie können es aus Ihrer Registratur auf diese Weise entfernen.



Tipp

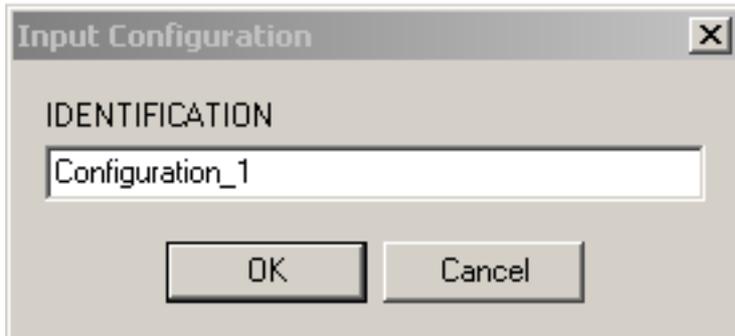
Falls Sie das Projekt in komprimierter Form speichern wollen, dann Klicken Sie "File | Back Up", es öffnet das Formblatt "Database Options". Nutzen Sie dort die -Hilfe!

9.4 Projekt weiter bearbeiten

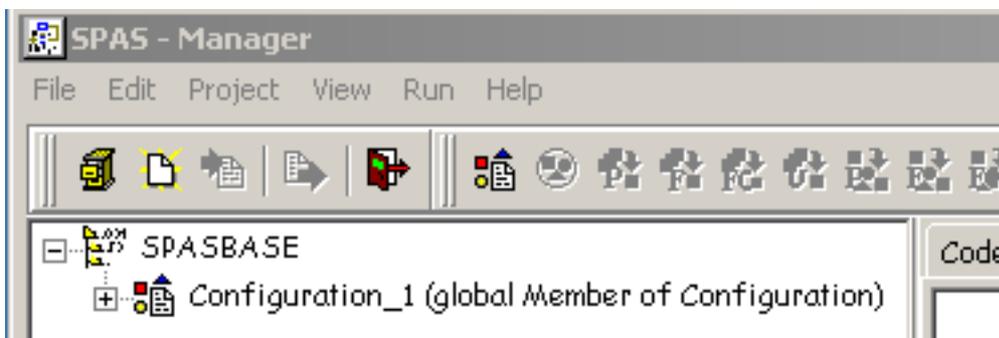
Konfiguration einfügen

Wenn sie ein neu angelegtes Projekt weiter bearbeiten wollen, gibt es noch keine Konfiguration [siehe Seite 144] C [siehe Seite 55]. Oder Sie wollen eine weitere Konfiguration in das vorhandene, geöffnete Projekt aufnehmen:

Markieren Sie den ersten Knoten, den Projektknoten, falls er noch nicht blau hinterlegt ist, und klicken auf . Oder Sie betätigen die rechte Maustaste und gehen weiter zu "INSERT" | "CONFIGURATION". Es öffnet sich das Fenster:

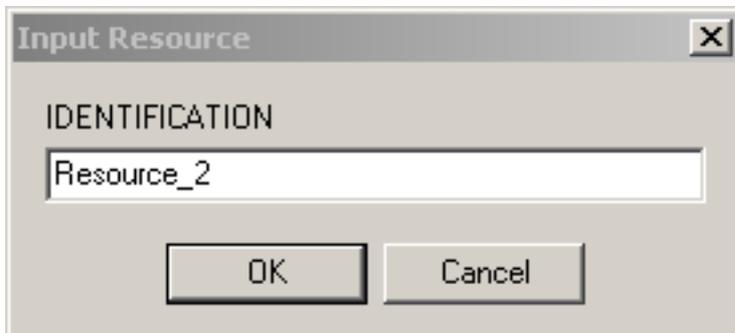


Tragen Sie hier einen technologisch relevanten Text ein. Denken Sie dabei daran, daß eine Konfiguration wenigstens eine Steuereinrichtung (SPS) als Ressource enthält und das Automatisierungssystem zu einer technologischen Anlage darstellt! Deshalb ist es dringend erforderlich, spätestens jetzt eine Prozeßzerlegung [siehe Seite 17] durchzuführen. Nach dem OK sehen Sie im Projektbaum



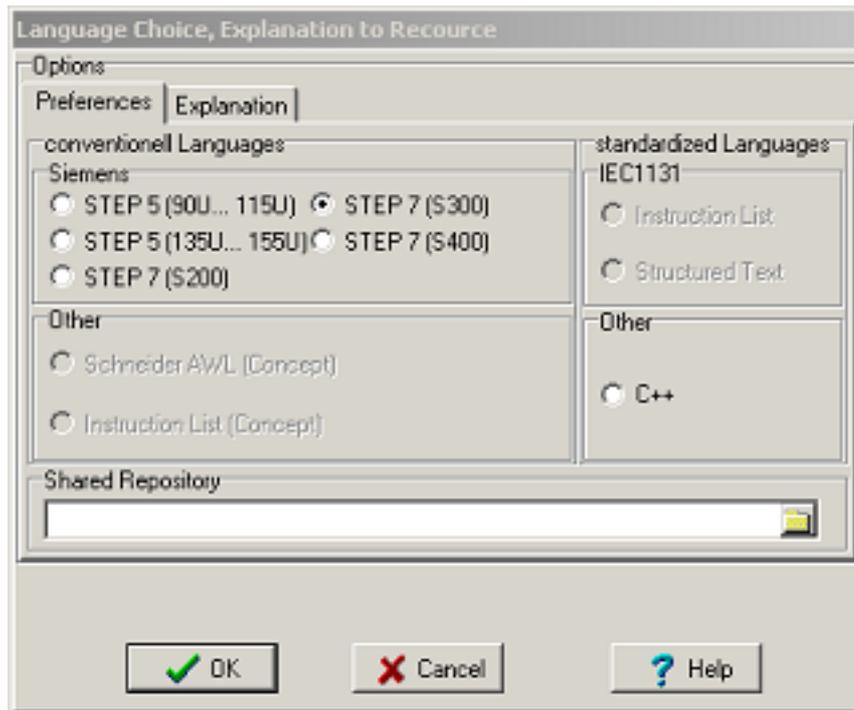
Ressource einfügen

Fügen Sie nach dem Markieren der Konfiguration eine RessourceR [siehe Seite 55] durch Klick auf  ein, indem Sie wiederum einen technologisch passenden Text eintragen.



Nach dem Eintrag dieses Knotens in den Projektbaum können Sie die erste Ressource erkennen.





Sie müssen unbedingt

die Sprache angeben. Lesen Sie das Kapitel über wählbare Sprachen [siehe Seite 7] und danach die Software-Produkt-Beschreibung von PreCOM. Dort wird Ihnen etwas zum Aufbau des Compilates und der Merker gesagt, die die Zustände und andere Eigenschaften kennzeichnen.

Ausführlichen Beschreibungstext bzw. Erläuterungstext zum Projekt schreiben Sie bitte in das vorgesehene Feld unter "Explanation".

Im Sinn von IEC 1131-3 entspricht jedes Teilprojekt dem Inhalt einer Ressource (engl. RESOURCE) [siehe Seite 187], ein Projekt entspricht im allgemeinen einer CONFIGURATION. [siehe Seite 144]

Weiter mit dem Einfügen von Prap-Typen [siehe Seite 56]

9.5 Konfiguration einfügen

In ein Projekt eine Konfiguration einfügen (Insert Configuration) bedeutet, Sie benötigen innerhalb Ihres Projektes einen vollkommen eigenen, in sich abgeschlossenen Bereich eines Automatisierungssystems. Die Konfiguration enthält wenigstens eine Ressource.

Wie eine Konfiguration einzufügen ist. [siehe Seite 53]

Zurück zu Projekt-Editor [siehe Seite 51]

9.6 Ressource einfügen

In eine Konfiguration ist wenigstens eine Ressource [siehe Seite 187], d.h. eine Steuereinrichtung (z.B. SPS, Microcontroller) aufzunehmen. Daher ist mit dem Einfügen auch deren Eigenschaft bezüglich Sprache [siehe Seite 210] zu markieren. Jede Ressource ist für einen bestimmten technologischen Abschnitt der zu automatisierenden Anlage "zuständig". Deshalb ist wenigstens ein Programm [siehe Seite 58] zuzuordnen, das die TP [siehe Seite 191] in Form von Instanzen verwaltet.

Innerhalb der Ressource sind die Tasks zu deklarieren:

Task - Eigenschaften und einfügen [siehe Seite 55]

Variablendeklarationen¹⁴

Wie eine Ressource einzufügen ist. [siehe Seite 53]

Zurück zum Projekt-Editor [siehe Seite 51]

9.7 TASK einfügen/bearbeiten

Die Norm IEC 1131-3 formuliert unter "Begriffe (Definitions)", Abschnitt 1.3, für den Begriff TASK:

"Element der Ausführungssteuerung, das für eine periodische oder getriggerte Ausführung einer Gruppe

von zugehörigen Programm-Organisationseinheiten sorgt."

Tasks sind

- "SINGLE", das sind einmalige Aufrufe, Interrupts, bei steigender Flanke einer Booleschen Variablen (keine Intervall-Angabe!)
- zyklisch: "INTERVAL", wenn eine Zeitangabe erfolgt (z.B. Aufruf alle 200ms), kein Eintrag bei "SINGLE"!
- ohne Angaben (weder bei "SINGLE" noch bei "INTERVAL"): Aufruf SPS-zyklisch, d.h. immer von vorn beginnend, wenn das Programm abgearbeitet ist. Die Priorität ist die niedrigste.
- Tasks werden unter einer Priorität (priority) vom Betriebssystem behandelt. Höchste Priorität= 0.
- Tasks, die zyklisch deklariert sind, sollten eine niedrige Priorität erhalten (abhängig von den Eigenschaften der Hardware, also der Ressource)als Interrupts.

Tasks können in jede Ressource mehrere eingefügt werden, aber wenigstens eine ist erforderlich!

So wird eine TASK eingefügt:

- Ressource, in die die TASK eingefügt werden soll, markieren
- ENTWEDER Klick auf  ODER Cursor zur Ressource führen, rechte Maustaste: Inerst | Task anklicken
- Felder für IDENTIFICATION und DESCRIPTION ausfüllen, dabei Bemerkungen beachten, die nach Klick auf "Help" erscheinen.
- Felder der "RELATIONS" ausfüllen. Dabei sollten die Eigenschaften der Zielmaschine (Ressource) beachtet werden, sofern nicht schon SPAS nur ausgewählte Möglichkeiten in der HILFE anbietet.

Zurück zum Projekt-Editor [siehe Seite 51]

9.8 PRAP-Typen einfügen/bearbeiten

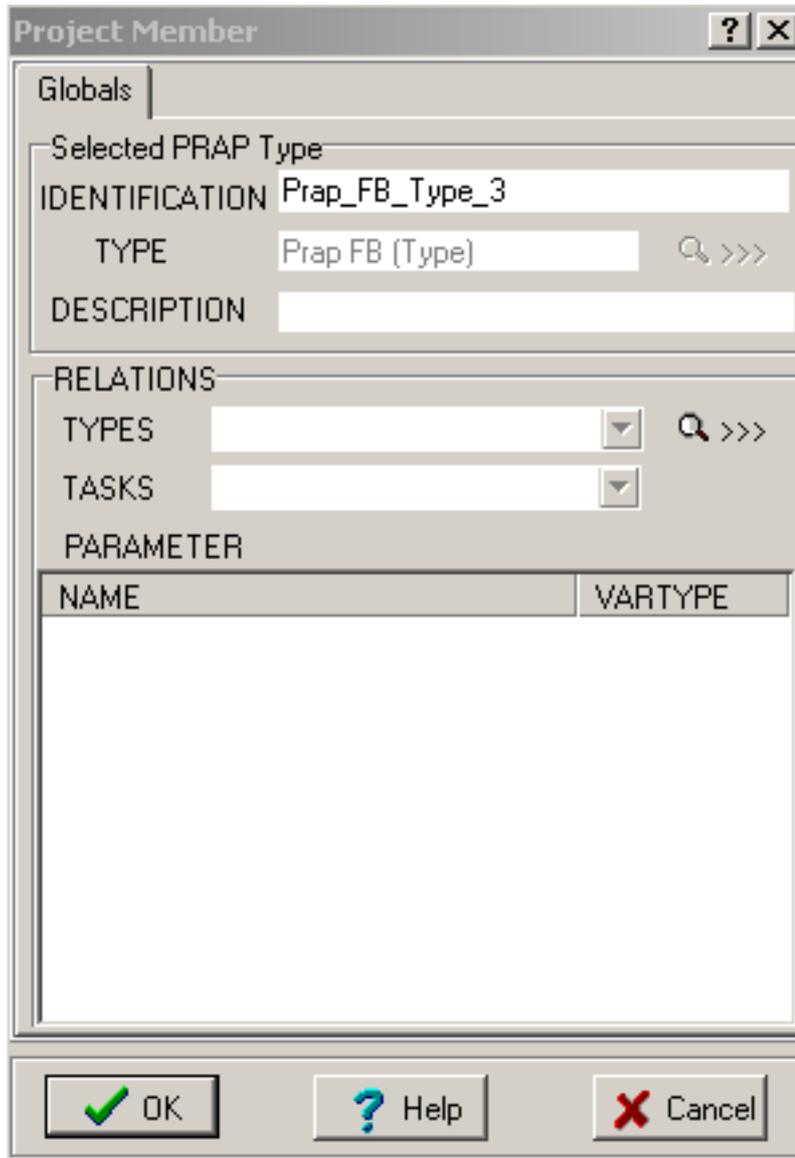
Voraussetzungen für das Einfügen von PRAP-Typen: [siehe Seite 193]

- Studieren Sie aufmerksam das Problem der Prozeßzerlegung [siehe Seite 17] und den Sinn des Prozeß-Zerlegungs-Graphen (PZG).
- Markieren Sie im PZG die Teilprozesse [siehe Seite 191], die unter einer Ressource aufzuführen sind. Das sind alle diejenigen, die in einer Steuereinrichtung (z.B. SPS) "laufen".
- Behalten Sie im Auge, daß der Prozeßzerlegungsgraph kein starres Schema ist, sondern mit den Erkenntnissen der Projektentwicklung verändert werden muß.
- Falls Sie für Teilprozesse auf vorhandene PRAP zurückgreifen, prüfen Sie sorgfältig, ob sie wirklich die notwendige Funktion erfüllen. Eventuelle Modifikationen neu benennen.
- Machen Sie sich mit dem PRAP-Editor [siehe Seite 61] vertraut. (Der folgende Text gilt nicht für Version 1.4.: Entwerfen Sie erforderliche PRAP nicht nur hier im Rahmen des Projektes, nutzen Sie den extern verfügbaren PRAP-Editor/Debugger, dessen Ergebnisse Sie in das Projekt importieren können.)
- Notieren Sie sich im PZG, welche Teilprozesse Sie als Instanzen eines gemeinsamen Typen-PRAP im Projekt führen wollen. Dazu eignet sich z.B. die Identification des Members. Beispiel: Instanz =

Pumpenantrieb, Type = Motor, Identification = Pumpenantrieb_Motor

- Es gibt die Möglichkeit, auch PRAP mit direkten Adressen [siehe Seite 123] hier einzufügen. Verschiedene konventionelle Sprachen machen dies erforderlich, weil die Anzahl der Formal-Parameter begrenzt ist. PRAP als Unikate [siehe Seite 193] sind auch zugelassen.

Die Ressource ist zu markieren und danach z.B. auf  zu klicken. Oder die rechte Maustaste gibt den Einfügebefehl vor. Es öffnet sich ein Fenster



Project Member [?] [X]

Globals

Selected PRAP Type

IDENTIFICATION Prap_FB_Type_3

TYPE Prap FB (Type) [?] >>>

DESCRIPTION

RELATIONS

TYPES [?] [?] >>>

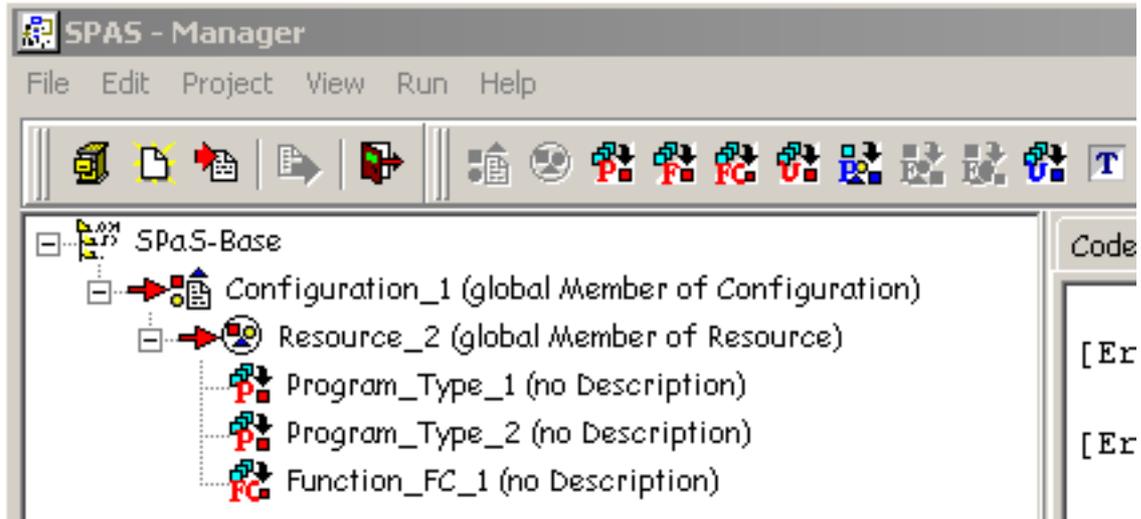
TASKS [?]

PARAMETER

NAME	VARTYPE

[OK] [Help] [Cancel]

Benutzen Sie die zur Verfügung stehende Hilfe zum Ausfüllen dieses Formulars. Im Ergebnis haben Sie einen Knoten eingefügt, aber ohne funktionellen Hintergrund. Im Beispiel sehen Sie zwei solcher Typen:



Zunächst sei festgestellt, daß jeder PRAP als typische funktionelle Darstellung in Form eines Graphen betrachtet wird. Die Folge davon ist, daß der vom PreCompiler erzeugte Quellcode einen Funktionsbaustein-Typ (= Function Block Type) ergibt.

Häufig tritt in der Anlagenautomatisierung das **Unikat** auf. Das bedeutet, es gibt im ganzen Automatisierungssystem genau einmal diesen Funktionsablauf bzw. Funktionsbaustein. SPAS kann dieses Problem des Unikates auf zwei Wegen verarbeiten. Entweder Sie stellen die einzige Instanz zu dem Typ einfach in das aufrufende Programm oder geben dem "Typen" bereits in der Ressource direkte Adressen für alle Variablen und fügen im Programm nur noch den Bezeichner (= Identification) ein als FB Instance ohne Aktualparameter-Übergabe. Falls Sie in der Bibliothek bereits einen passenden PRAP vorrätig haben, importieren Sie ihn mit Klick auf  in die Datenbank. Sollten Sie mit TeCADŠ einen PRAP vorliegen oder entworfen haben, können Sie ihn in SPAS hineinkonvertieren mit Klick auf .

Soll der PRAP neu entworfen werden, gehen Sie weiter zum PRAP-Editor. [siehe Seite 61]

Zurück zum Projekt-Editor [siehe Seite 51]

9.9 Funktion in eine Ressource einfügen

Eine Funktion ist gemäß IEC 1131-3 als eine Programm-Organisationseinheit definiert, die keine Speicherfunktionen in sich enthält.

Markieren der Ressource, danach Klick auf  oder nach dem Markieren rechte Maustaste und Insert | Function.

Es öffnet der Memberdialog. Hier Müssen Sie in "IDENTIFICATION" den Namen der Funktion [siehe Seite 172] eingeben. Die Zeile "DESCRIPTION" soll eine Beschreibung [siehe Seite 143] der Funktion ihrer Aufgabe entsprechend erhalten, in möglichst ein oder zwei Worten.

9.10 PROGRAM einfügen/bearbeiten

Der Begriff PROGRAM = Programm ist in IEC 1131-1 definiert als "eine logische Anordnung von allen Programm-Sprachelementen und -Konstrukten, die für die beabsichtigte Signalverarbeitung ... erforderlich sind."

In SPAS bedeutet PROGRAM zunächst die Einordnung aller Instanz-Funktionsbausteine [siehe Seite 126] und Funktionen [siehe Seite 154] in der Reihenfolge ihres Aufrufes. Die Einordnung erfolgt in der nächsten Ebene nach "PROGRAM".



Anmerkung

Damit wird ein Programm-Typ definiert.

Alle Programm-Instanzen erhalten zusätzlich die Zuweisung über die zeitliche Aufrufordnung durch Tasks.

In einem Typ-Programm können

- globale Variablen deklariert werden, die direkten Adressen [siehe Seite 123] zugeordnet sind. Es handelt sich um Variable, die innerhalb des Programms gelten, wozu alle zu den Instanzen und Funktionen gehörenden Aktualparameter zählen, insbesondere "USAGE" mit Input bzw. Output oder beides.
- Es können auch Koppelvariable (kopp [siehe Seite 170]) aufgeführt werden, solange sie innerhalb des Programms für den Datenaustausch zwischen den TP, also den Instanzen des Programms, zuständig sind. Direkte Boolesche RAM-Adressen sind mit "Address Memory" als USAGE deklariert. Koppelvariable, die mit "USAGE" = "Local Static" gekennzeichnet wurden, erhalten durch den Maschinencompiler die Adressen automatisch zugewiesen und deshalb KEINE DIREKTEN ADRESSEN.

Für spätere SPAS-Version vorgesehene Externals¹⁵
Zurück zum Projekt-Editor [siehe Seite 51]

9.11 PRAP INSTANCE einfügen/bearbeiten

Jedem PRAP-Typ ist wenigstens eine Instanz zuzuordnen. Unikate [siehe Seite 193] werden bezüglich Verwendung der Variablen / Variable Usage [siehe Seite 195] als Instanz behandelt. Markieren Sie das Typ-Programm, in das eine PRAP-Instanz eingefügt werden soll. Ein Klick auf die rechte Maustaste zeigt Ihnen eine Befehlsauswahl

in der Sie "Insert" anklicken und die PRAP-Instanz zum Einfügen angeboten wird. Natürlich können Sie auch gleich das Symbol  anklicken.

¹⁵ In einem Programm-Typ können externe Variable deklariert werden, wenn aus den Global-Deklarationen bei CONFIGURATION bzw. RESOURCE Variable benötigt werden. Halten Sie sich bitte an die Hilfetexte zu den Formblättern, um Variable neu zu deklarieren oder als extren zu übernehmen.

Nutzen Sie die Hilfeangebote zum Ausfüllen des Formblattes.

Zurück zum Projekt-Editor [siehe Seite 51]

9.12 PROGRAM INSTANCE einfügen/bearbeiten

Jede Programm-Instanz basiert auf einem Typ-PROGRAM und bedarf einer Ausführungs-"Bestimmung" durch eine Task, die bei der Ressource deklariert wurde.

Programm-Instanz zum Beispielprojekt [siehe Seite 87]

Zurück zu Projekt-Editor [siehe Seite 51]

9.13 Funktions-Aufruf

Der Aufruf einer Funktion, die in der Ressource deklariert wurde [siehe Seite 58], erfordert die Zuordnung der Aktualparameter zu den formalen. Die aufzurufende Funktion [siehe Seite 154] sollte im Projektbaum eingefügt werden, wo die zu verarbeitenden Daten schon vorhanden sind und das Ergebnis noch im gleichen Bearbeitungszyklus der CPU genutzt wird. Diese Regel gilt vor allem für temporär deklarierte Variable.

Der Aufruf in einem PRAP ist noch nicht möglich.

9.14 Variablenliste importieren

Markieren Sie im Projektbaum das PROGRAM(Type) bzw. die RESOURCE, in das Sie eine Variablenliste (nur Excel) importieren wollen, betätigen Sie die rechte Maustaste und dann Klick auf "Import Variable List".

Weitere Hinweise: ImportVarTab [siehe Seite 198]

Kapitel 10

PRAP-Editor

10.1 PRAP-Editor

Zugang zum PRAP-Editor aus dem Projekt-Editor:

siehe Projekt-Menue [siehe Seite 161]

Graphischer Entwurf der Steuerungsfunktion eines Teilprozesses als Typ-PRAP [siehe Seite 193] oder als TP-PRAP [siehe Seite 126].

Der Entwurf umfaßt die Automatenfunktion einschließlich der sogenannten eingebetteten oder nichtbinären Funktionen [siehe Seite 25]:

- Flankenerkennungen
- Vergleiche
- Zeit- und Zählglieder
- Umspeicherung von Daten (Kopierfunktion)
- Arithmetikfunktionen

Soweit die kurze Einführung.

Dem PRAP liegt die Automatentheorie zu Grunde, die Sie als Anwender nicht kennen müssen. Ein paar Bemerkungen sind im Thema

Allgemeine Theorie [siehe Seite 62] zusammengefaßt.

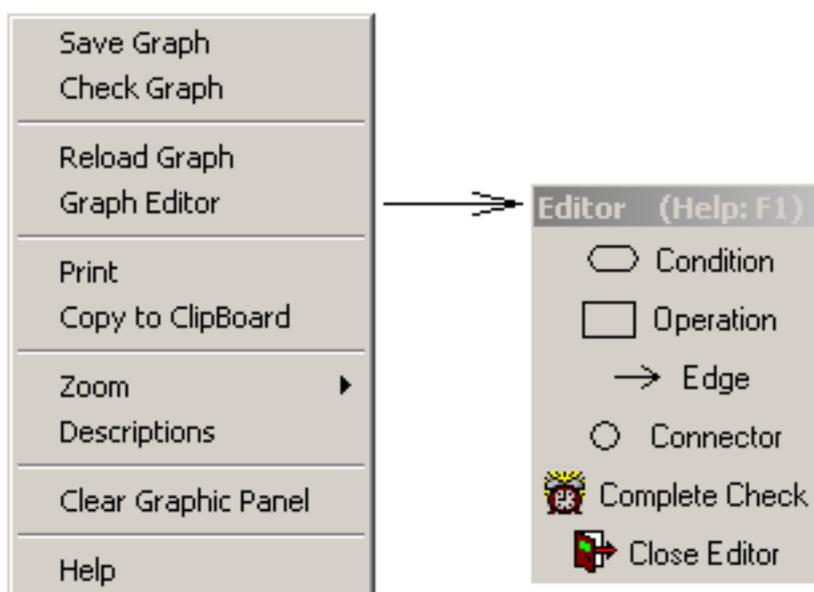
Der PRAP ist aber auch auf des Basis der Graphentheorie entwickelt, worüber Sie im Thema

Der PRAP als Graph [siehe Seite 63] etwas erfahren.

Damit Sie wissen, wie man einen PRAP aufbauen sollte, gibt es Hinweise im Thema Grundsätze zum Gestalten eines PRAP [siehe Seite 63].

Am Ende eines Entwurfes oder auch mittendrin sollte man wissen, worauf zu achten ist. So entstand das Thema Prüfungen des PRAP [siehe Seite 64].

Zum Entwickeln bzw. Zeichnen eines PRAP stehen die Menues



zur Verfügung. Das linke erhalten Sie durch Betätigen der rechten Maustaste im Feld "CHARTS", das rechte durch Klick auf "Graph Editor". Bitte nutzen Sie dort die angebotenen Hilfetexte.

Der Druck eines PRAP wird mit Klick auf 'Print' [siehe Seite 214] eingeleitet.



Tip

Sie können den ganzen PRAP oder Teile auf der Bildebene verschieben. Dazu die linke Maustaste festhalten und die Maus ziehen. Innerhalb des sich aufspannenden Rechtecks, rosa hinterlegt, werden die Objekte markiert. Dann linke Maustaste loslassen, ein markiertes Objekt anfassen¹⁶ und mit Maus verschieben.

Falls Sie alles rückgängig machen wollen, dann im linken Menue Klick auf "Reload Graph".

An Hand von Beispielen soll etwas über die Praxis berichtet werden. Dazu dient das Thema Entwicklung eines PRAP [siehe Seite 66].

10.2 Allgemeine Theorie des PRAP

Als eine prozeßorientierte Sprache beschreibt der PRAP den funktionellen Zusammenhang zwischen Prozeßeingang und -ausgang (siehe: Informationsflußmodell [siehe Seite 14]). Dabei handelt es sich zumeist um einen Teil des zu steuernden technologischen Prozesses, also eines Teilprozesses. [siehe Seite 191]

Der Prozeßablaufplan ist das Steuerungs-Modell eines zu steuernden (Teil-)Prozesses.

Der PRAP ist ein gerichteter, stark zusammenhängender [siehe Seite 190] Graph, der auf automatentheoretischen Grundlagen (siehe Fußnote) beruht:

- Das Automatenmodell ist grundsätzlich ungetaktet und mit einem Initialzustand behaftet.
- Jeder Automatenzustand wird durch eine Prozeß-Situation [siehe Seite 183] definiert, und es gibt genau eine Initialsituation [siehe Seite 166].
- Jede Situation wird gemäß dem Modell des Moore-Automaten behandelt, wenn sie sich nicht in einer kombinatorischen Struktur befindet.
- Jede kombinatorische [siehe Seite 168] Struktur enthält wenigstens zwei Situationen, und eine kombinatorische Struktur wird als ein Zustand gemäß dem Modell des Mealy-Automaten behandelt.
- Ferner erfüllt der PRAP automatentheoretische Forderungen, wie Widerspruchsfreiheit [siehe Seite 197] und Vollständigkeit [siehe Seite 197].

Die Eigenschaft der Vollständigkeit garantiert eine technologisch gesicherte Aussage über die steuerungstechnischen Ausgaben, d.h. Wirkungen jeder Operation [siehe Seite 174] auf den zu steuernden (Teil-)Prozeß.

Der starke Zusammenhang garantiert einen technologisch gesicherten Steuerungsablauf in allen Situationen des PRAP.

Voraussetzung für die Erstellung eines PRAP ist, daß der erste Schritt der Prozeßerlegung stattgefunden hat.

Wie ein PRAP entwickelt bzw. entworfen wird, zeigt Ihnen das Thema Entwicklung eines PRAP [siehe Seite 66].

Die nichtbinären Funktionen sind in die Ausgabe- bzw. Eingabefunktion des Automaten eingebettet. Die Flankenerkennungen und die Vergleichsfunktionen liefern ihr Funktionsergebnis an eine Boolesche Variable und sind damit Prozeßvariable (Bedingungen, Conditions). Das gilt auch für Zeit- und Zählgliedfunktionen bzw. deren Funktionsergebnis. Andererseits sind das Ein- und Ausschalten von Zeit- und Zählfunktionen sowie Durchführung arithmetischer Berechnungen oder Kopierfunktionen an Operationen bzw. Situationen im PRAP gebunden und daher durch Operationsvariable zu steuern.

Automatentheoretische Grundlagen unterscheiden sich grundsätzlich von Petri-Netzgrundlagen, die dem Prozeßablaufnetz zu Grunde liegen.

¹⁶ Cursor auf ein markiertes Objekt führen, linke Maustaste gedrückt halten.

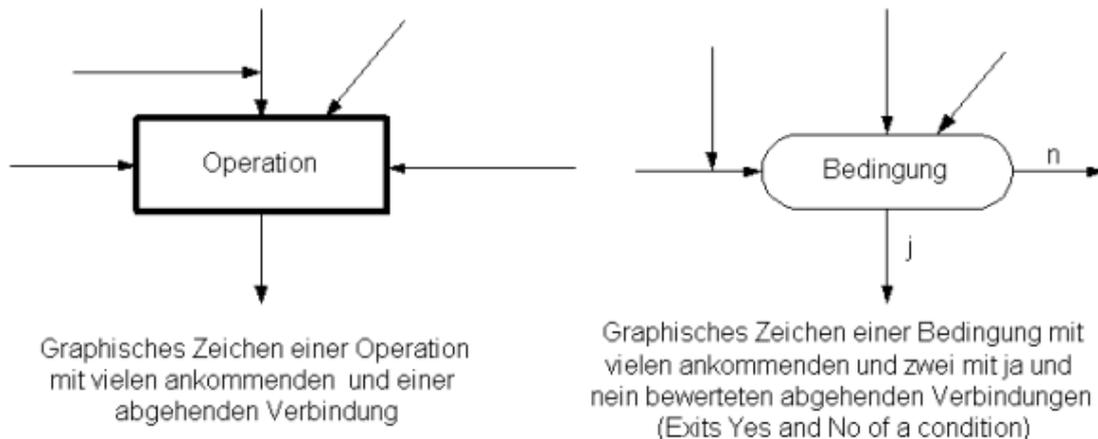
10.3 Der PRAP - Ein Graph

Jeder PRAP verfügt über zwei Grundmengen, Knoten des Graphen, die Menge der Operationen [siehe Seite 174] und

die Menge der Prozeßzustandsvariablen [siehe Seite 183].

Die Menge der Operationen und die Menge der Prozeßzustandsvariablen sind die Knotenmengen des Graphen, der stark zusammenhängend [siehe Seite 190] ist.

Zur Darstellung dieser Knoten bedient man sich graphischer Elemente:



Die Kanten sind die gerichteten Verbindungen [siehe Seite 217] zwischen diesen Knoten. Sie werden durch Pfeile dargestellt.

Zu einem O-Element können viele Pfeile hinführen, es darf aber nur durch einen verlassen werden.

Zu einem P-Element können viele Pfeile hinführen, es muß aber nur von genau zwei Pfeilen verlassen werden. Diese dürfen zu einem O- oder P-Element führen und sind mit der Bewertung "ja" bzw. "nein" zu versehen (Yes / No).

Zwei Operationen heißen benachbart, wenn sie durch einen Pfad [siehe Seite 177] verbunden sind, dabei heißt die am Anfang des Pfades Start- oder Quell- oder Anfangsoperation (Start Operation), die am Ende Ziel- oder Folgeoperation (Succeeding Operation).

Ein Weg durch einen PRAP enthält mehrere Pfade und verbindet beliebig viele Elemente beider Art.

10.4 Grundsätze zum Gestalten eines PRAP

- Da der PRAP ein Steuerungsmodell eines Teilprozesses (TP) [siehe Seite 191] ist, gehören zu ihm Prozeß- und Operations-Variable [siehe Seite 194], die Sie möglichst schon benennen sollten. Die konkreten, technologisch relevanten Variablenbezeichner sind Instanzen [siehe Seite 126] oder Unikaten [siehe Seite 193] vorbehalten. Demnach kann ein PRAP nur ein Typ [siehe Seite 193] oder ein Unikat sein. Ist er ein Typ, dann gestalten Sie die Variablenbenennungen allgemeiner, aber immer technologisch sinnvoll und ohne Abkürzungen.

- Anfangsoperation festlegen! Jede Operation [siehe Seite 174] wird durch ein liegendes Rechteck dargestellt. Die Bezeichnung der Operation (=Statement) sollte den technologischen Inhalt ausdrücken.

Folgeoperationen nach dieser Anfangsoperation überlegen.

Unter Folgeoperation ist stets der unmittelbare Nachfolger zu einer betrachteten (Anfangs-)Operation zu verstehen.

Die Anzahl der Folgeoperationen zu einer betrachteten ist theoretisch unbegrenzt.

Operationsvariable [siehe Seite 176] und deren Wertung eintragen.

Hallenzufahrt beleuchten ist ein Beispiel dafür, daß in der betreffenden Operation die

Beleuchtung eingeschaltet wird, die Wertung wird durch den "Haken" markiert. Soll in der darauf folgenden Operation die Beleuchtung ausgeschaltet werden, ist die Wertung zu ändern, also "kein

Haken". Anderenfalls bleibt auch in der Folgeoperation die Beleuchtung eingeschaltet.

- Welche Prozeßzustandsvariablen gehören zum Übergang Anfangsoperation => Folgeoperation?
- Jede Prozeßzustands-Variable [siehe Seite 183] wird durch ein liegendes "Langrund" (kein Rechteck sondern rechts und links abgerundet) dargestellt, eine sog. Bedingung (=Condition). Den Übergang bilden alle Bedingungen zwischen den beiden Operationen mit einer der zugelassenen Aussagen Ja oder Nein und bilden damit einen Pfad.
Jeder Pfad P_{jk} zwischen zwei Operationen O_j und O_k ist ein Teil p des Übergangs-Prozeßzustandes P_{jk} . [siehe Seite 193]
EMPFEHLUNG:
In jedem Pfad sollte möglichst jede Prozeßvariable nur einmal verwendet werden.
- Ist eine Prozeßgröße nicht erfaßbar, kann sie durch einen Zeit- oder Zählbaustein ersetzt (simuliert) werden. In vielen Fällen kann auf das Ersetzen der Prozeßgröße durch Zeit- oder Zählbausteine verzichtet werden, wenn man einen physikalischen Zusammenhang geschaffen hat, der die erwartete Prozeßgröße zu berechnen gestattet bzw. eindeutig abbildet.
- Beim Übergang von einer Operation O_j zur Folgeoperation O_k können sich beliebig viele der Operationsvariablen bezüglich ihrer Bewertung ändern. Zu beachten ist dabei, daß immer alle zum PRAP gehörenden Operationsvariablen in Betracht gezogen werden!
- Außer den Übergangsprozeßzuständen P_{jk} ($j \neq k$), gibt es auch solche, bei denen $j=k$ ist, die sogenannten Stabilitäts-Prozeßzustände P_{jj} . [siehe Seite 183] Sie sind die Voraussetzung dafür, daß die in jeder Operation O_j eingetragenen Operationsvariablen an den Prozeß ausgegeben werden. O_j und P_{jj} gehören als Paar zeitlich zusammen, und es muß gelesen werden:
Solange P_{jj} erfüllt ist, wird O_j ausgeführt.



Tip

Der Grundsatz "Solange P_{kk} erfüllt ist, wird O_k ausgeführt" enthält eine Ausnahme. Im ersten Verarbeitungs-Zyklus während eines Übergangs von einer Operation der Situation S_j zu einer anderen der Situation S_k erfolgt die Ausgabe der zugehörigen Signale auch dann, wenn P_{kk} zu O_k nicht erfüllt ist. Im nächsten Verarbeitungszyklus wird die dann folgende Situation erreicht.

Damit wird gewährleistet, daß Koppelsignale ausgegeben werden, auch in einem solchen instabilen Fall der S_k . Andererseits können Sie sich diesen "Effekt" nutzbar machen.

10.5 Prüfungen des PRAP

Der PRAP zeichnet sich u.a. dadurch aus, daß er automatentheoretische, wie auch graphentheoretische Forderungen erfüllt. Die Widerspruchsfreiheit und Vollständigkeit, Eigenschaften des Automaten, sind zwangsläufig durch die Struktur des PRAP gegeben, deshalb sind folgende Prüfungen, die nach dem Editieren durchgeführt werden müssen, notwendig:

- Halteprozeßzustand zu jeder Operation
- starker Zusammenhang
- Wertebelegung der Prozeßvariablen Y und N
- Start- oder Initial-Operation (Launch Operation)
- Anzahl der abgehenden Kanten jeder Operation = 1, es muß also ein "Folge"-Objekt - Condition - existieren, das aber keine Operation sein darf.

- Anzahl der abgehenden Kanten jeder Bedingung = 2, es müssen also zwei "Folge"-Objekte existieren.

Alle diese Eigenschaften prüft Ihnen der PRAP-Editor, wenn Sie das Grapik-Menü¹⁷ (rechte Maustaste) aufrufen und dann "Check Graph" anklicken.

Außerdem ist der PRAP hinsichtlich seiner Stabilität, Sicherheit, Dynamik und Vereinfachbarkeit zu überprüfen; Die im Folgenden aufgeführten Fehler finden Sie im Sheet "Compiler Notes" kommentiert:

- Ein Durchlauf [siehe Seite 150] einer Operation ist kritisch zu betrachten und zu vermeiden. Beachten Sie die Eigenschaften bedingungsstabil [siehe Seite 139] und strukturstabil [siehe Seite 189] eines Übergangs von einer Operation zur Folgeoperation.
- Es ist zu darauf zu achten, daß nicht alle Übergangszustände in einem Operationen-Zyklus [siehe Seite 176] gleichzeitig auftreten. Es bilden sich dadurch instabile Zyklen, die unbedingt vermieden werden müssen. Anderenfalls besteht Gefahr für den gesteuerten Prozeß! Die Funktionalität des PRAP ist so gestört, daß die gesamte Steuerung ausfallen kann.
- Wenn Unsicherheiten bezüglich der Zuverlässigkeit bei Übergängen von einer Operation zur Folgeoperation bestehen, so sollten Sie strukturstabile Übergänge bevorzugen.
- Bei der Simulation der Steuerfunktion des PRAP (siehe) ist es wichtig, die Dynamik der Prozeßvariablen zu beachten. Sie kann statisch [siehe Seite 188] oder dynamisch [siehe Seite 150] sein.
- Durch Einführung kombinatorischer Strukturen [siehe Seite 168] kann die Gestalt des PRAP mitunter deutlich vereinfacht werden. Der Vorteil der einfacheren Lesbarkeit hat auch den Vorteil eines kürzeren Programmcodes.
- Kontrollieren Sie die richtige Folge der Wertungen der Operationsvariablen, weil davon die Steuerungsfunktion abhängt! Lesen Sie eventuell nochmals im ersten Punkt bei Gestalten des PRAP [siehe Seite 63]
- Zu Ihrer Sicherheit führen Sie eine Übersetzung aus, indem Sie den Cursor auf die linke Seite des Managers (Projektbaum) stellen; Betätigen der rechten Maustaste öffnet das Projektmenü, dann ein Klick auf "Build".
Nur durch den Übersetzungslauf erhalten Sie Mitteilungen über z.B. Probleme mit allen eingebetteten Funktionen. Wenn Sie alles richtig gemacht haben, meldet das Fenster "Compiling" bei "Status" das erlösende Wort "Done". Ansonsten sind die Fehlermeldungen zu beachten.

17

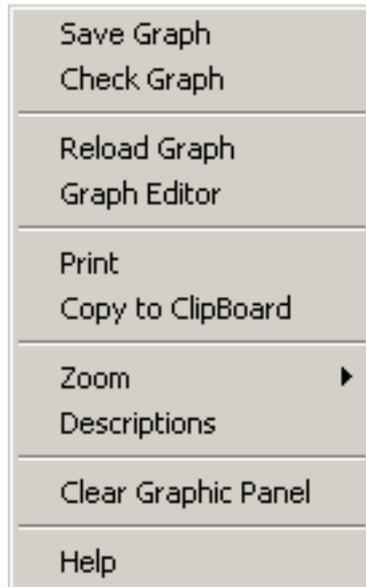


10.6 Export und Import von PRAP

Export eines PRAP

- Sie können sich ein Archiv für entworfene PRAP anlegen. Klicken Sie auf . Zur Erläuterung dient: EXportPRAP [siehe Seite 210]

- Sie finden im Menue



die Zeile "Copy to Clipboard", womit Sie

den PRAP im Format Bitmap zwischenspeichern können. Öffnen Sie mit einem geeigneten Programm¹⁸ diese Daten und speichern danach an beliebiger Stelle.

Import eines PRAP

Aus einer Bibliothek, die exportierte PRAP enthält, können Sie diese wieder verwenden. Klicken Sie auf .

10.7 Alle PRAP zum Beispielprojekt

10.7.1 Entwicklung eines PRAP (zum Projekt Lehrbeispiel)



Anmerkung

Es ist zu betonen, daß ein PRAP zwar eine Teilprozeßbeschreibung ist, aber nur im Fall eines Unikates [siehe Seite 193] trifft dies vollständig zu. Ansonsten ist ein PRAP ein Typ [siehe Seite 193], dessen Konkretisierung durch Instanziierung [siehe Seite 126] erfolgt und dann erst als Teilprozeß gilt. Alle Variablen im PRAP sind daher im allgemeinen formale.

Zunächst muß festgestellt werden, welche Steuer-Größen am Teilprozeßeingang und welche Prozeßzustandsgrößen am Teilprozeßausgang auftreten (siehe: Informationsfluß-Modell [siehe Seite 14]).

Zu jedem Teilprozeß [siehe Seite 191] gehört eine bestimmte Anzahl Ausgangsvariable (jede Ausgangsvariable ist einer Operationsvariablen [siehe Seite 176] zugeordnet die es in keinem anderen TP gibt. Eine Wertebelegung [siehe Seite 197] aller Operationsvariablen wird Operation [siehe Seite 174] genannt.

¹⁸ Es können empfohlen werden:

- MS Word und "Bearbeiten | Einfügen" mit gutem Bild, aber als DOC-Datei speicherbar.
- Corel Photo Paint und "Datei | Einfügen aus Zwischenablage" mit gutem, scharfen Bild als BMP-Datei speicherbar.

Die Eingangsvariablen jedes TP ergeben sich aus den Prozeßvariablen des Teilprozeßausgangs und dürfen zu mehreren TP gehören.

Die Entwicklung eines PRAP ist ein iterativer Vorgang, der zwischen Prozeßzerlegung [siehe Seite 17] und Teilprozeßbeschreibung (anderer Ausdruck für Prozeßablaufplan) verläuft.

Allgemeine Grundlagen:

Grundsätze zum Gestalten eines PRAP [siehe Seite 63]

Prüfungen und Optimieren eines PRAP [siehe Seite 64]

Bedienen des PRAP-Editors an Hand von Beispielen:

Beispiel Lampensteuerung [siehe Seite 105]

Beispiel Stern-Dreieck-Schaltung (mit Zeitglied-Projektierung) [siehe Seite 88]

Beispiel Motorstrom-Verarbeitung (mit Vergleichs-Projektierung) [siehe Seite 102]

Kapitel 11

Software-Produkt-Beschreibung

11.1 Precompiler

Der Precompiler erzeugt NACH KLICK AUF **BUILD** in der angegebenen Zielsprache

- bei MARKIERTER RESOURCE
 - den Quellcode aus jedem graphisch entwickelten PRAP
 - den Quellcode des Steuerungsprogramms exakt gemäß Projektbaum
 - die Symbolliste aller Variablen
- bei MARKIERTEM PROGRAMM
 - den Quellcode aus zum Programm gehörigen PRAP
 - den Quellcode des zum Programm gehörigen Teils der untergeordneten Instanzen, bzw. Teilprozesse
 - die Symbol-Teilliste der zum Programm gehörenden Variablen
- bei MARKIERTEM PRAP oder MARKIERTER FUNCTION
 - den Quellcode dieses PRAP oder dieser FUNCTION
 - die Symbolliste dieses PRAP oder dieser FUNCTION

Der Precompiler erzeugt NACH KLICK AUF **SAVE TO FILE** in der angegebenen Zielsprache

- eine oder mehrere Dateien für die Symbolverwaltung
- eine oder mehrere Dateien für die Steuerprogramm-Implementierung
- eine Datei für die Darstellung der Instanzen als PRAP für
 - statisches Debugging jeder Instanz
 - dynamische Darstellung der Instanzen als PRAP mit den aktuellen Signalen aus der Ressource (Profibus zwischen PC und Ressource)

Das Softwareprodukt ist syntaktisch fehlerfrei. Das bedeutet, z.B. STEP7 importiert ohne Fehlermeldung sowohl die AWL als auch die Symbolliste.

Sollten im Projekt Mängel enthalten sein, die nicht zu einem fehlerfreien Ergebnis führen, gibt es ausführliche Meldungen. In der Version 1 sind diese häufig nicht kommentiert, daher oft nicht zu verstehen.

Der Aufbau eines Kompilates zu einem PRAP ist generell wie folgt:

- Am Anfang stehen die Variablendeklarationen, geordnet nach Input, Output, InputOutput, Local, Temporary.
Hier finden Sie auch automatisch im Precompiler erzeugte, die zum Automaten [siehe Seite 62] gehören.
- Darauf folgend ist die Anlaufroutine zu sehen, sie garantiert den Start im Initialzustand des Automaten (markierte Anfangsoperation im PRAP gehört zum Initialzustand).
- Bevor der Automat mit seinem ersten Zustand im Kompilat steht, sind erst alle Vergleiche, Zeit- und Zählgliedabfragen und Flanken-Funktionen von Eingangs-Variablen eingetragen.
- Der Automat, in seine Zustände aufgegliedert, schließt sich nun an. Anfangs sind es die Zustandsübergangs-Bedingungen, danach erforderlichenfalls die Stabilitäts-Bedingungen.
- Am Ende des Automaten steht die Ausgabe aller Booleschen Variablen, die in den Operationen markiert waren, mit dem logischen Wert 1. Die nicht markierten erhalten den logischen Wert 0.
- Nach dem Automaten folgen alle Zeitbausteine, Zähler, Kopierfunktionen (reine Zuweisungen) und arithmetische Funktionen, auch Schiebe- und Rotier-Vorgänge.
- Hieran schließen sich für die ONLINE-Darstellung des PRAP notwendige Programmzeilen an.

Der Aufbau des Kompilates zu einem PROGRAM:

- Genau in der von Ihnen angegebenen Reihenfolge im Projekt erfolgen die Aufrufe der Instanzen. Diese sind am Schlüsselwort "CALL" (manchmal auch nur CAL) oder "RUN" zu erkennen.
- Zu jedem Aufruf gehört zwingend die Übergabe der Aktual-Parameter an die formalen im aufgerufenen Typ.

Bemerkungen zu den Sprachen:

STEP 7 - Anweisungsliste [siehe Seite 70]

Quelltext zu C [siehe Seite 70]

Quelltext zu C++ [siehe Seite 70]

11.2 STEP-7-AWL

Das Ergebnis des Precompilers ist davon abhängig, ob eine Ressource oder das Programm oder ein Typ-PRAP im Projektbaum markiert wurde.

Die Vorübersetzung einer Ressource liefert drei Dateien, eine NAME.AWL, eine NAME.SDF (Symboltabelle) und eine NAME.ONL - dies ist eine Online-Datei, die die Rückdarstellung der PRAP während des Betriebes der Steuerung bzw. der SPS gewährleistet, Sie erleben den sog. "lebendigen PRAP". NAME ist der Eintrag bei "Identification" der Ressource.

Sie können für die FB-Instanzen auch Instanz-DB bei "DB Settings" erzeugen. Tragen Sie hier nichts ein, dann entsteht ein Multi-Instanz-DB bei "PROGRAM INSTANCE", ansonsten ist er ein Instant-DB für die Programm-Instanz allein.

Mehr zu STEP 7 idh_S7300400 [siehe Seite 8]

11.3 C-Programm

Zur Zeit keine Erklärung

11.4 Programm in C++

Zur Zeit keine Erklärung

Kapitel 12

Beispielprojekte

12.1 Lampen- und Stern-Dreieck-Schaltung

12.1.1 Vorbemerkungen zum Beispielprojekt

Das Beispiel hat kein real existierendes Vorbild, es ist also ein Phantasie-Projekt mit den wichtigsten Entwurfs-Abläufen, wie sie in SPAS benötigt werden. Bitte stellen Sie sich vor, es handelt sich um ein Tor einer Halle, das horizontal bewegt wird. Es soll einen Antriebsmotor (Drehstrom) zum Öffnen und einen zum Schließen geben. Wenn es geschlossen ist, soll eine Lampe das geschlossene Tor beleuchten, wenn es geöffnet ist, soll die Halle beleuchtet werden. Soweit die Beschreibung einer gedachten Teilanlage, die natürlich Mängel hat. Darauf kommt es aber hier nicht an.

Das Beispiel enthält zwei grundsätzliche funktionelle Zusammenhänge, die Beleuchtungssteuerung und die Motorsteuerung der Antriebe. Natürlich muß es noch weitere Steuerungs-Teilprozesse geben, z.B. einen zur Koordinierung der beiden Antriebe. Hier sollen nur die Funktionen der Lampe und der Anlaufsteuerung - in Form einer Stern-Dreieck-Schaltung - betrachtet werden.

Von diesen beiden Typen ist je ein Prozeßablaufplan - Typen-PRAP - zu entwerfen. Und jeder dieser PRAP ist die Grundlage zu den insgesamt vier beschriebenen realen Funktionen der Beleuchtung und der Antriebe. Das sind die sogenannten Instanzen der Typen, wie sie in einem Prozeß-Zerlegungs-Graphen [siehe Seite 17] (PZG) im ersten Entwurf¹⁹ enthalten sein sollten.

Es sei hier schon darauf hingewiesen: Ein weiterer Typen-PRAP für zwei Instanzen wird noch hinzugenommen werden, weil die Diskussion um den Stern-Dreieck-Anlauf dies ergibt.

Zur Vorbereitung des Steuerungsprojektes legen Sie bitte einen Ordner/ein Verzeichnis "LaSteDre" an. Weiter mit dem Anlegen des Beispielprojektes [siehe Seite 73].

12.1.2 Anlegen des Projektes, Configuration und Ressource

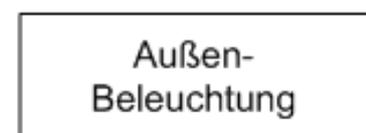
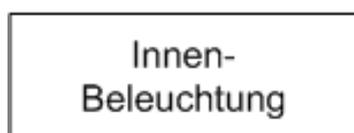
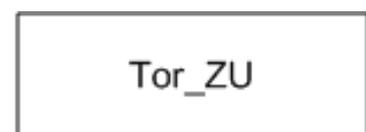
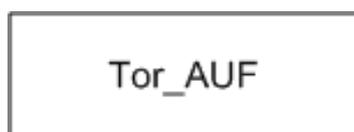
Vorbemerkung:

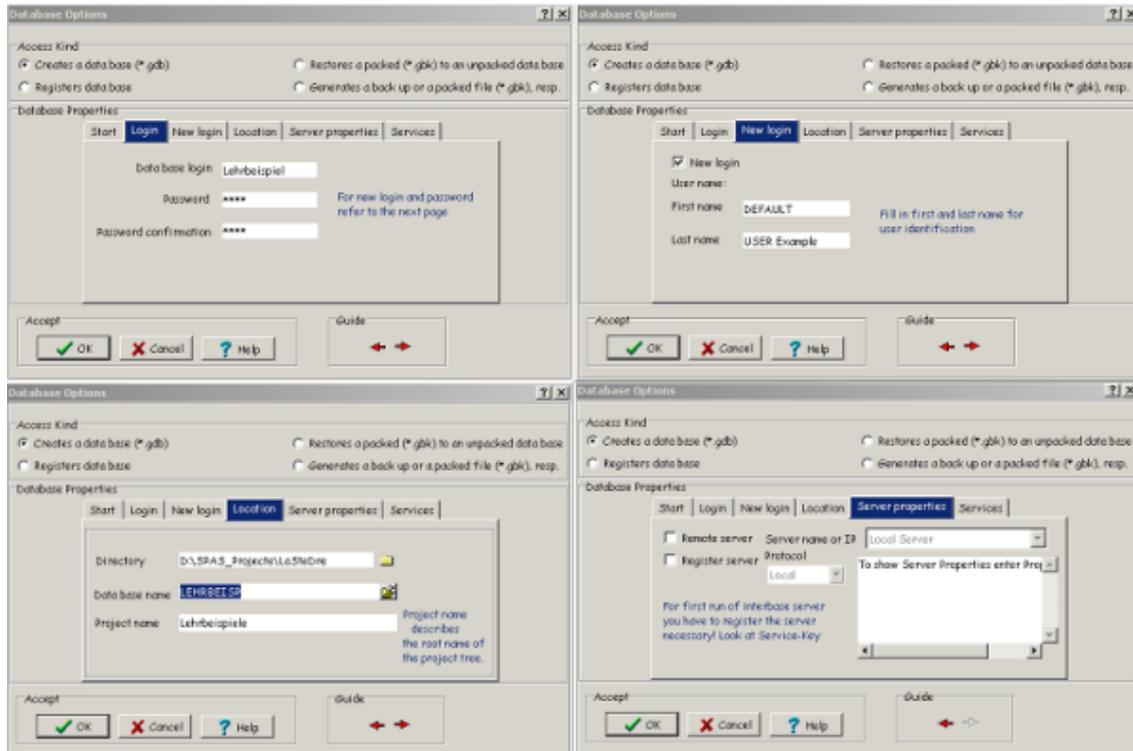
"[ERROR] Database not connected" steht nur am Anfang als Hinweis darauf, die Eröffnung einer Datenbank noch durchzuführen.

Es wird angenommen, daß ein Ordner, z.B. "LaSteDre" (=Lampe Stern-Dreieck), für das Beispielprojekt eingerichtet wurde.

Gemäß dem Vorgang beim Anlegen eines neuen Projektes [siehe Seite 51] sind die Database Options in den nachfolgenden Bildern zu erkennen:

¹⁹

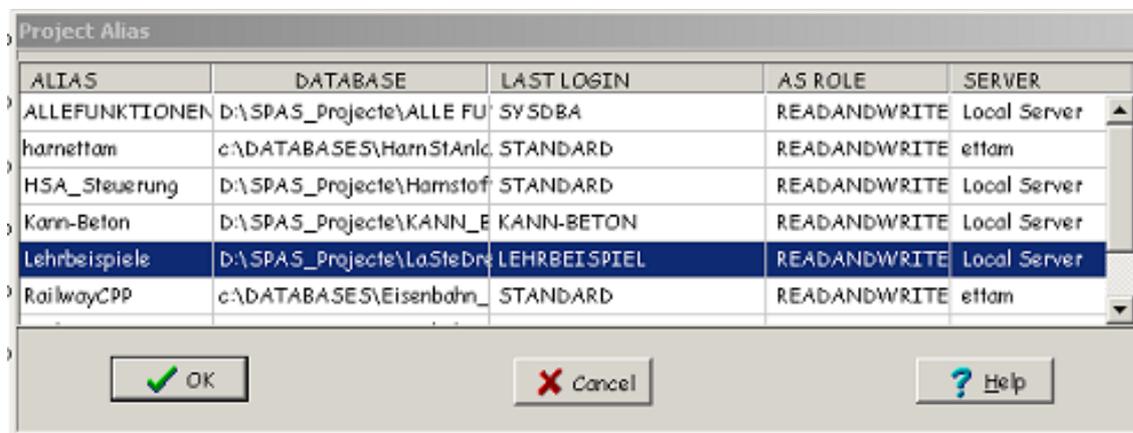




Es wurde ein neues Login eingetragen. Das Passwort ist "spas".

Nach dem Klick auf OK bzw. Betätigen der Enter-Taste erhält man das Protokoll über das Anlegen der Datenbank. Die letzte Meldung besagt, die Datenbank ist geöffnet (unlocked). Nun ist anschließend auch das Projekt zu bearbeiten.

Der Zugang zum neuen Projekt erfolgt mit Klick auf . Das Beispiel-Projekt ist auszuwählen (Markieren) und OK zu klicken:

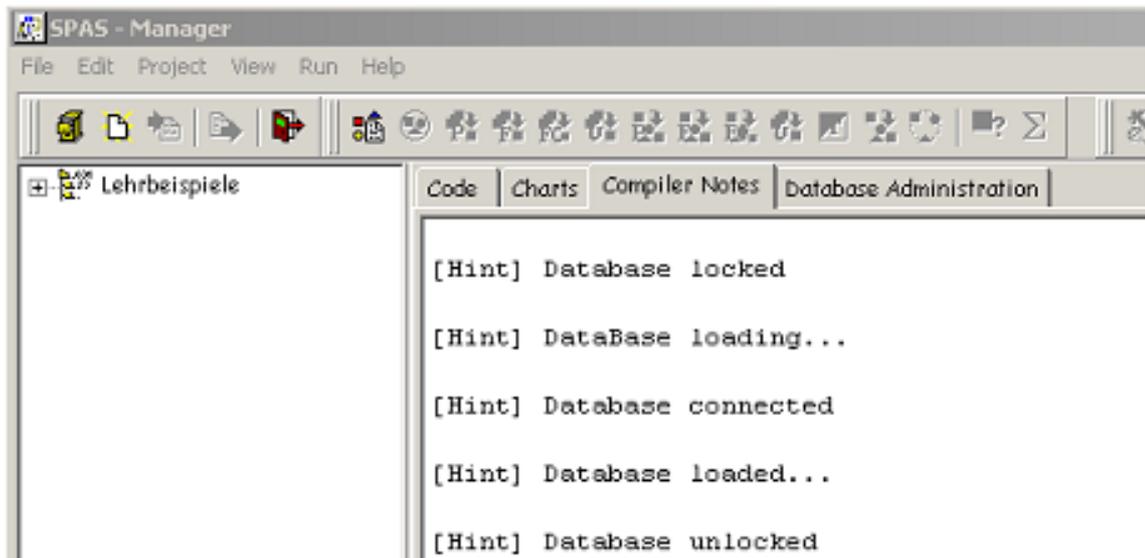


Nun benötigen Sie Login und Passwort, die beim Anlegen des Projektes verwendet wurden. Da das User-Login "Lehrbeispiel" und "spas" das Passwort ist, können Sie das Projekt öffnen.

Nun folgen Arbeitshinweise, bis am Ende mitgeteilt wird, die Datenbasis ist geöffnet, die Projektbearbeitung kann beginnen.



Das Ergebnis:

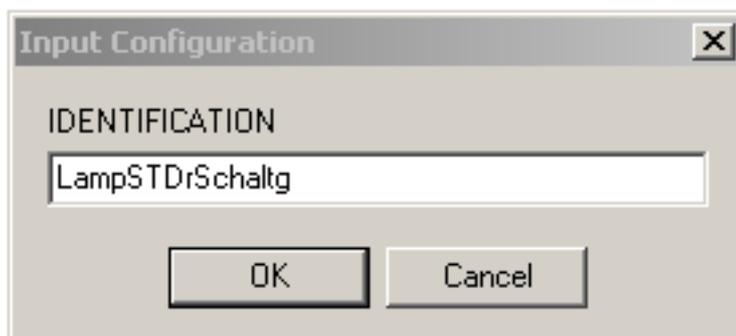


Tipp

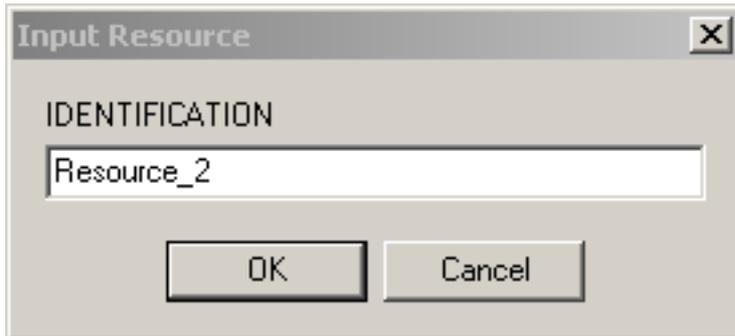
Löschen Sie ab und zu die Mitteilungen (Rechte Maustaste auf dem Mitteilungsfeld), wenn sie Ihnen im Wege sind.

Nun ist die Konfiguration einzufügen. Das Lehrbeispiel beschränkt sich auf eine Konfiguration mit einer Ressource.

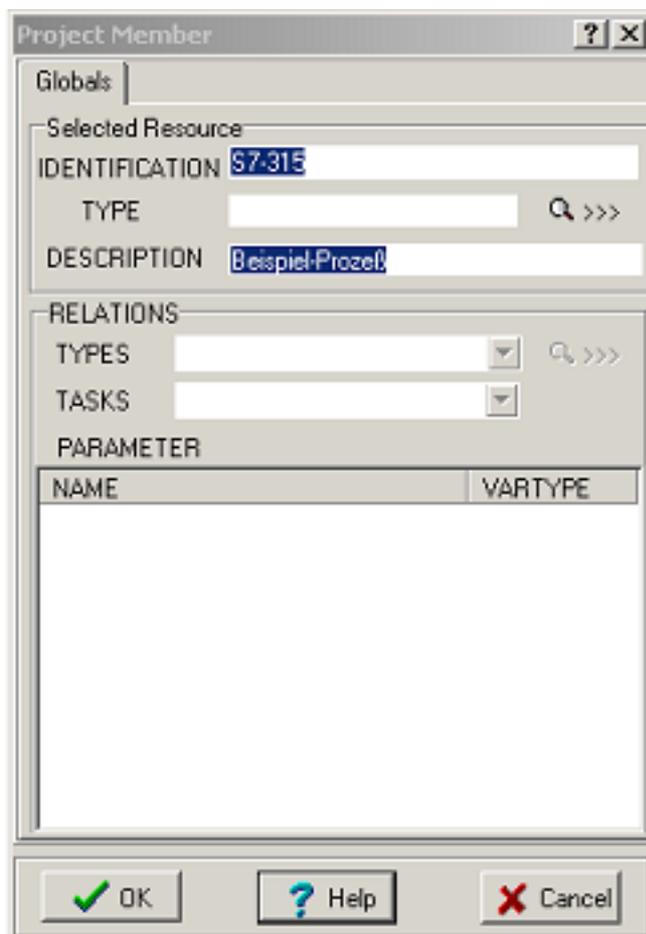
Mit Klick auf  können Sie die Bezeichnung eingeben, die möglichst technologie-orientiert sein soll, also den Bereich der technologischen Anlage benennen soll, für die das nachfolgende Automatisierungssystem gilt:



In ähnlicher Weise ist die Ressource mittels  einzufügen. Es öffnet ein Fenster,



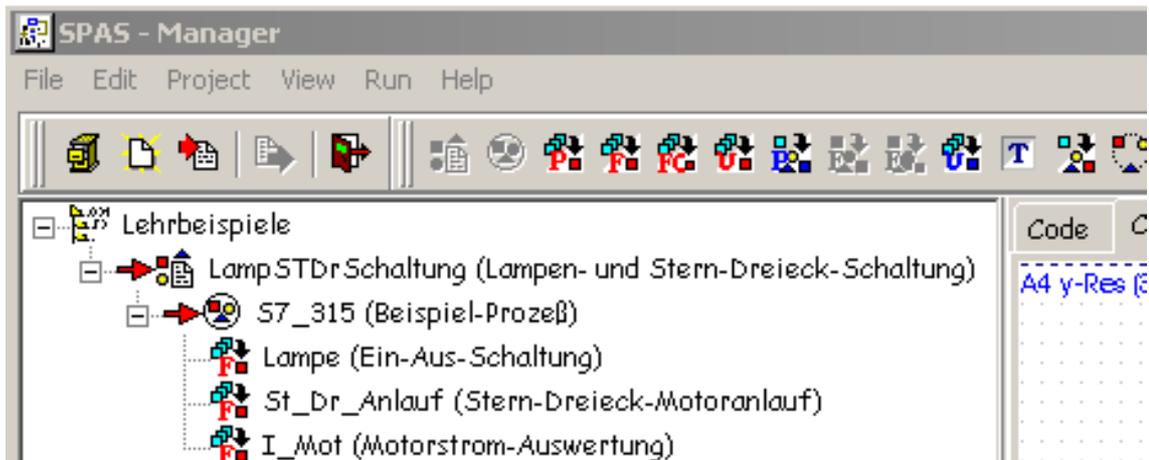
in dem Sie entweder den Standard-Text stehen lassen und "OK" klicken, oder Sie tragen die Bezeichnung der Steuerungs-Einrichtung vorher ein. Danach sofort entweder mit Doppelklick auf den Standardtext oder anders, z.B. Klick auf , den "Member Dialog" starten. Auch hier gilt die Regel, die Ressource so zu bezeichnen, daß vollkommen klar ist, welche Steuereinheit eingesetzt wird. Der hier passende Eintrag könnte so aussehen:



Damit erhält man einen Projektbaum:

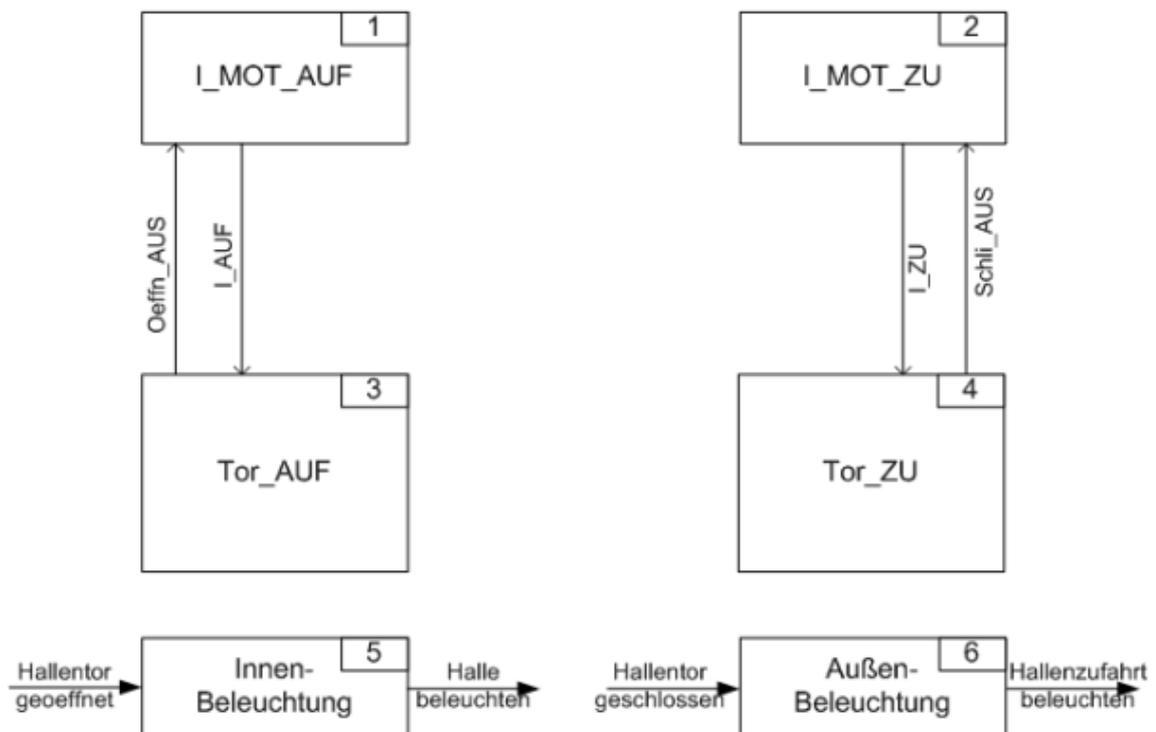


Die beiden PRAP-Typen sind hier im Beispiel die Lampensteuerung und die Stern-Dreieck-Schaltung. Gemäß der Beschreibung PRAP-Typen einfügen [siehe Seite 56] wurde der Projektbaum ergänzt:



Die Entwicklung der PRAP siehe: PRAP-Editor | Entwicklung eines PRAP, Beispiele [siehe Seite 66]

Wie schon in den Vorbemerkungen V [siehe Seite 73] steht, ist ein zusätzlicher Typen-PRAP hinzugekommen, der den Motorenstrom im Rahmen des Anlaufes erfaßt. Damit muß auch der PZG erweitert werden, wobei auch die Kopplungen mit eingetragen sind:



Anmerkung

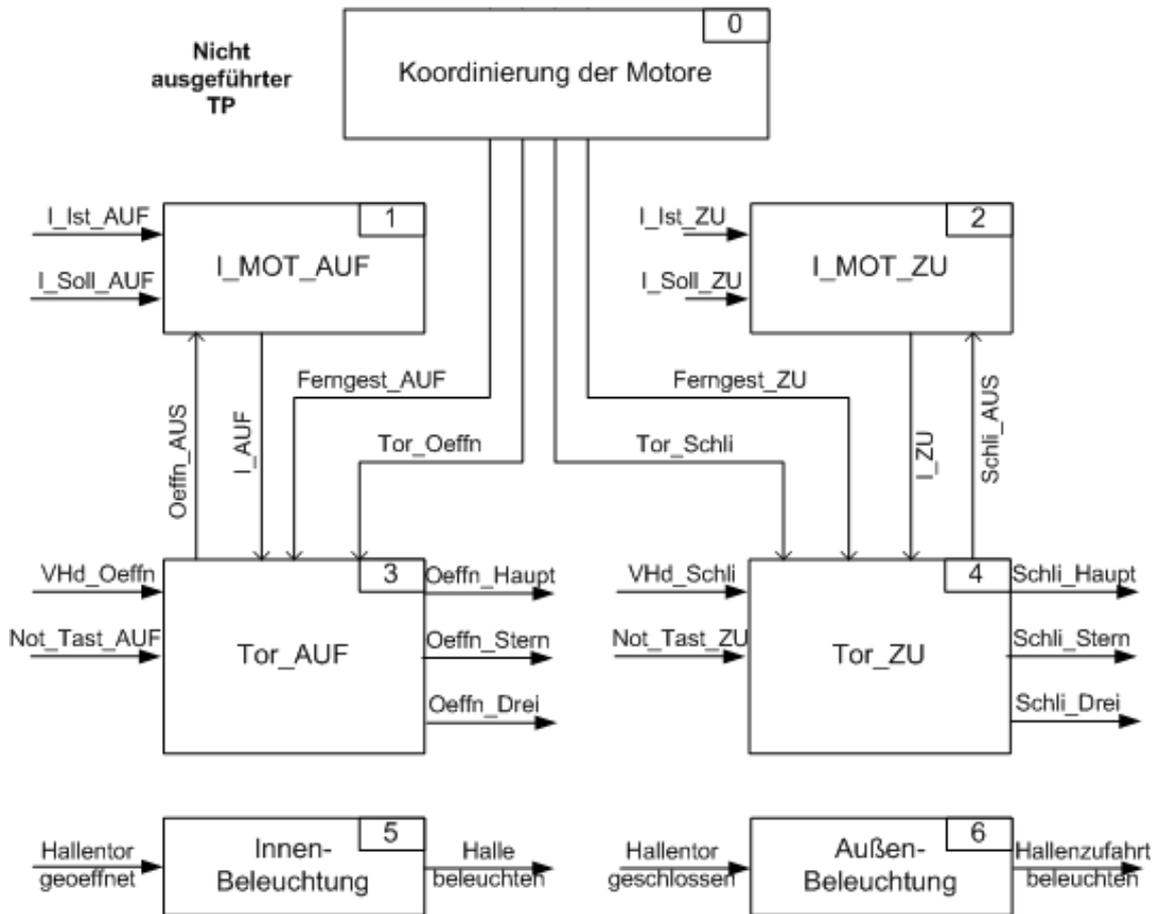
Das Beispielprojekt ist für die Ziel-Sprache STEP 7 vorgesehen, weshalb es zu jedem PRAP auch eine FB-Nummer geben muß. Wählen Sie diese nach eigenem Ermessen.

Nachdem alle Typen-PRAP der Ressource eingefügt sind, ist mindestens eine TASK [siehe Seite 87] einzutragen. Hier im Beispiel soll es eine SPS-zyklische Task geben. Sie hat die Priorität 1 und keine Angabe zu SINGLE/INTERVAL. Schließlich ergibt sich daraus für STEP 7 der OB1.

Im Thema Ressource einfügen [siehe Seite 55] stand der Hinweis, globale Variable im Rahmen der Ressource zu deklarieren. Dazu bedarf es der Entscheidung, ob es Variable gibt, die "oberhalb" der Programme gelten. Im Beispielprojekt ist nur ein Programm enthalten. Damit brauchen hier keine Globaldeklarationen zu erfolgen, und alle globalen Variablen können im PROGRAM [siehe Seite 79] "lokal" (Local Satic) deklariert werden.

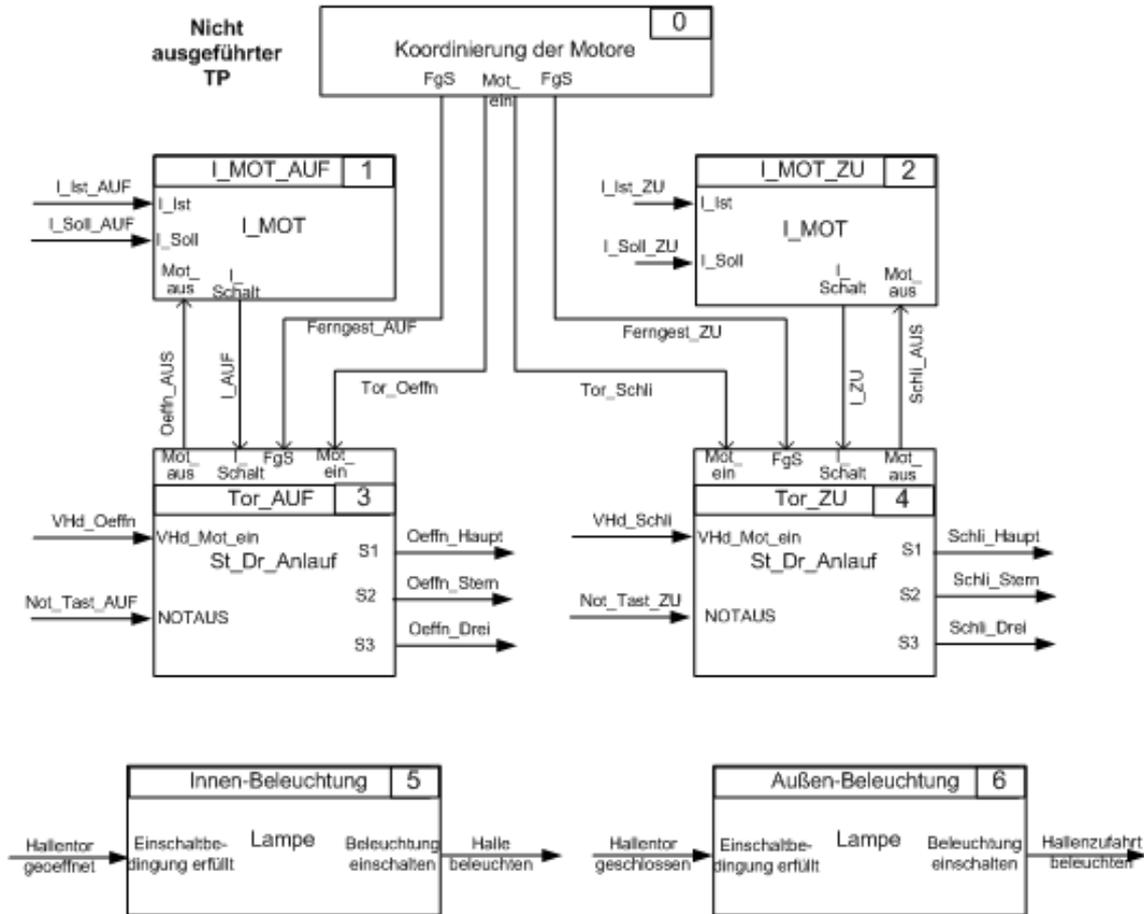
Aus dem Verdrahtungs-Projekt, das hierzu natürlich nicht existiert, oder im Rahmen des Übergangs vom PZG zur Funktionsblock-Darstellung (FBD) [siehe Seite 23] kann eine Liste der Variablen [siehe Seite 48] und ihrer Adressen gewonnen werden. Tragen Sie jetzt oder nach dem Einfügen der Instanzen

die Aktualparameter der Koppelvariablen ein. Bedienen Sie sich der im Formblatt angebotenen Hilfe!
Die Variablen sind auch im neuen PZG zu erkennen:



Prozess-Zerlegungs-Graph zum Lehrbeispiel
Lampen-Stern-Dreieck-Schaltung

Das erstrebte Endergebnis FBD erhalten Sie schließlich durch Ergänzen des PZG mit den zu den Instanzen gehörigen Typen mit deren Variablen. Die soeben erwähnte Exceltabelle benötigt dann nur noch Eintragungen einschließlich der TP-Namen; wirklich importiert werden ohnehin nur die Daten ohne TP. In der nachfolgend dargestellten Funktionsblock-Darstellung können Sie erkennen, wie der "Endstand" aussehen sollte.



Funktionsblock-Darstellung zum Lehrbeispiel
Lampen-Stern-Dreieck-Schaltung

Zugegeben, es ist eine etwas mühsame Arbeit, die jedoch auch eine Entwurfssicherheit gewährt, und das ist allemal die Mühe wert.

Weiter mit der Instanziierung der PRAP-Typen [siehe Seite 79].

12.1.3 Typ-Programm mit Instanzen der Typ-PRAP

Nach den drei Typen-PRAP ist ein Member "PROGRAM" (idh_PRAPTypes_Insert [siehe Seite 56]) in das Projekt einzufügen, dem alle Instanzen der PRAP-Typen unterzuordnen sind. Und es müssen alle direkten Adressen der Aktualparameter eingetragen werden (Projectmember_PROGRAM [siehe Seite ?]). Dazu später mehr.

Markieren Sie die Ressource und klicken auf  oder nutzen das Menue mit Klick auf die rechte Maustaste: Insert|PROGRAM Type. Tragen Sie z.B. "Programm" als Identification ein und "FB-Instanzen" als Description.

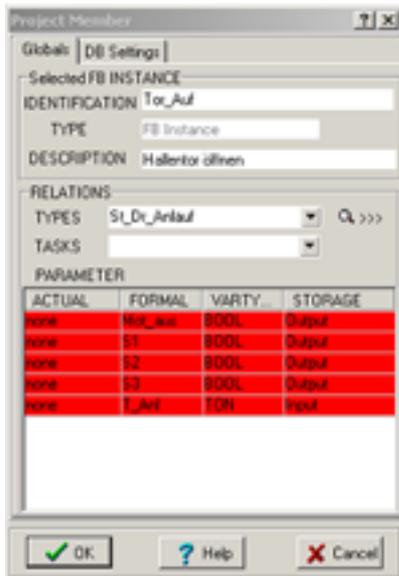


Tip

Es kann sein, das PROGRAM ist nicht als letztes Member zu sehen. Dann markieren, rechte Maustaste und Klick auf  Build Later, bis die gewünschte Stelle erreicht ist.

Nun müssen die Instanzen in das Programm eingefügt werden:

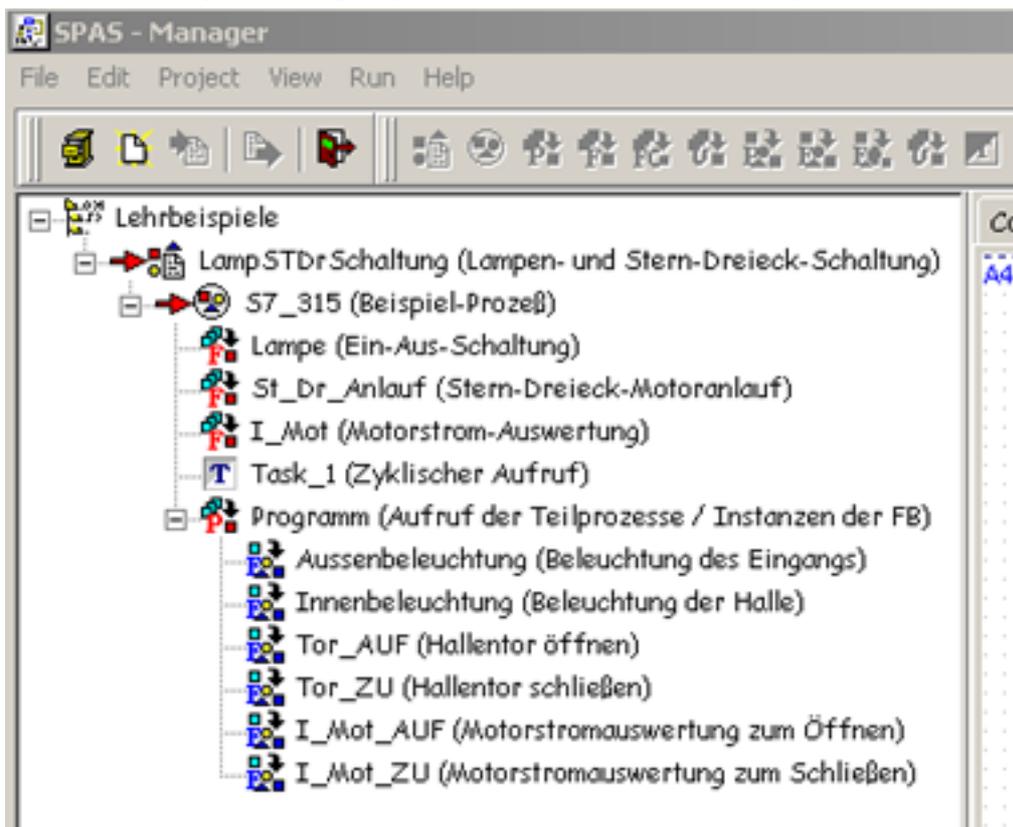
Markieren Sie also das Member "Programm", klicken auf  und tragen die Texte für "Identification" und "Description" ein. Außerdem müssen Sie zu jeder Instanz den richtigen Typen-PRAP wählen. Das Ergebnis sehen Sie im folgenden Bild:



Als Beispiel sehen Sie die Instanz "Tor_Auf", die zum Typ

"St_Dr_Anlauf" gehört. Rot eingetragen sind alle Formalparameter des Typen, neben denen "none" steht, solange keine Aktualparameter zugewiesen sind. (siehe Teilprozesse fertigstellen, Instanzen komplettieren [siehe Seite 84])

Nachdem alle Teilprozesse eingetragen sind, hat der Projektbaum das folgende Aussehen:



Damit haben Sie erst die Instanzen im Projekt angelegt. Es gibt noch zwei Probleme zu lösen:

- Die Reihenfolge, in der die Teilprozesse im PROGRAM eingetragen sind, bestimmt die Aufruf-Folge. Also entsprechend ordnen mit Markieren, rechte Maustaste und Klick auf  oder 
- Die Variablen der Instanzen müssen noch ihre Aktualparameter den Formalparametern [siehe Seite ?] der zugehörigen Typen zuordnen.
Also sind spätestens jetzt durch Sie für jeden Teilprozeß (= jede Instanz) die Aktualparameter zu benennen mit den zugehörigen Adressen. (siehe Variablenliste der Instanzen [siehe Seite 83] oder Variablen-Excel-Liste [siehe Seite 82])

Weiterführung im Thema Variable in das Programm [siehe Seite 81] eintragen.

12.1.4 Variable in das Programm eintragen

Zu bedenken ist, ob diese Variablen und deren Adressen global in "RESOURCE" oder "PROGRAM" eingetragen werden.

Die Grundregel lautet: Variable, die in mehreren Programmen benötigt werden, sollten in der Ressource als globale Variable deklariert werden. Ausschließlich im PROGRAM benötigte Variable sollten nur hier und damit lokal deklariert werden.

Im Beispielprojekt sind alle Variablen der Instanzen in das Programm eingetragen.

Externe Deklarationen sind erforderlich, wenn im PROGRAM Variable aus der Ressource benötigt und dort global deklariert wurden.

In der SPAS-Version 1 ist es zwar möglich, in der Ressource globale Variable zu deklarieren, aber es fehlen die External-Deklarationen. Dennoch kann zur Instanzierung auf diese Variablen zugegriffen werden.

Eintragen der Variablen in das Programm oder in die Ressource:

- Doppelklick auf "Program-Type" bzw. "Resource" im Projektbaum.
- Klick auf  öffnet die Datenbanktabelle / Access Table
- Tragen Sie der Reihe nach die Daten aus der Variablenliste [siehe Seite 83] für alle Variablen ein. Benutzen Sie die angebotenen Hilfemöglichkeiten. Lesen Sie eventuell beim Projektmember PROGRAM [siehe Seite 211] nochmal nach.

Importieren der Excel-Liste:

- "Program-Type" bzw. "Resource" markieren und Rechte-Maustaste-Klick. Es öffnet das Projektmenue.
- Klick auf die letzte Zeile vor Help: "Import Variable List"
- Folgen Sie den Hilfemöglichkeiten, die in diesem Formular angeboten werden
- Sie finden das Beispiel hier. [siehe Seite 82]

Das Ergebnis des Listen-Importes zeigt das folgende Bild:

Project Member [?] [X]

Globals | **FB Settings**

Selected PROGRAM Type

IDENTIFICATION Programm

TYPE Program (Type) [Q >>>]

DESCRIPTION FB-Instanzen

RELATIONS

TYPES [v] [Q >>>]

TASKS [v]

PARAMETER

NAME	VARTYPE	STORAGE
ferngest_AUF	BOOL	Address Memory ▲
ferngest_ZU	BOOL	Address Memory
Halle_beleucht	BOOL	Address Memory
Hallenzuf_bel	BOOL	Address Memory
Halltor_geoeffn	BOOL	Address Memory
Halltor_geschl	BOOL	Address Memory
I_AUF	BOOL	Address Memory
I_AUF1	BOOL	Address Memory
I_LIST_AUF	INT	Address Memory
I_LIST_ZU	INT	Address Memory ▼

[OK] [Help] [Cancel]

Die Variablen sind alphabetisch geordnet. Das Feld enthält drei Spalten: NAME, VARTYPE und USAGE. Damit ist der Programm-Typ bis auf "FB Settings" komplett. Dieses Sheet liegt nur bei, da bei STEP 7 eine Multiinstanz erzeugt werden kann. Bei C++ oder C ist das Sheet nicht dabei. Hier im Beispiel wurde die Zahl 10 eingetragen.

12.1.5 Excel-Variablenliste zum Beispielprojekt

Die nachstehend gezeigte Liste wird in das Programm importiert.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Act_Name	Actual Statement	Prefix	Addr.	Dt_Type	USAGE	Init_Val.	Point of Automation	TP_Name	PRAP_Name	Form_Param.	USAGE	Formal Comment
2	Halltor_geoeffn	Halltor geöffnet	E	10	BOOL	Address Input	G5 0301	Innenbel	Lampe		L1_einschalten	Input	Schaltbedingung erfüllt:
3	Halle_beleuchtet	Halle beleuchtet	A	10	BOOL	Address Output	EU 0302	Innenbel	Lampe		L1_EN	Output	Lampe einschalten
4													
5	Halltor_geschl	Halltor geschlossen	E	00	BOOL	Address Input	G5 0601	Aussenbel	Lampe		L1_einschalten	Input	Schaltbedingung erfüllt:
6	Hallenzuf_bel	Hallenzuf beleuchten	A	00	BOOL	Address Output	EU 0602	Aussenbel	Lampe		L1_EN	Output	Lampe einschalten
7													
8	Tor_oeffn	Tor öffnen	M	10.0	BOOL	Address Memory	KOP von Koordination	Tor_AUF	St_Dr_An1		Mot_ein	Input	Motor einschalten
9	ferngest_AUF	Öffnen ferngesteuert	M	10	BOOL	Address Memory	KOP von Koordination	Tor_AUF	St_Dr_An1		FgS	Input	Ferngesteuert schalten
10	VHd_oeffn	Öffnen von Hand	E	12	BOOL	Address Input	H5 0301	Tor_AUF	St_Dr_An1		VHd_Mot_ein	Input	Motor von Hand einschalten
11	NOT_Tast	Öffnen NOT-AUS	E	3.0	BOOL	Address Input	H2 0202	Tor_AUF	St_Dr_An1		NOTAUS	Input	NOT-Aus
12	I_AUF	Schaltwert zum Öffnen erreicht	M	0.0	BOOL	Address Memory	KOP von LMOT_AUF	Tor_AUF	St_Dr_An1		I_Schalt	Input	Strom hat Weicherschaltwert erreicht
13	Oeffn_Haupt	Hauptschütz Öffnen EIN	A	11	BOOL	Address Output	EU 0203	Tor_AUF	St_Dr_An1		S1	Output	Hauptschütz einschalten
14	Oeffn_Stern	Sternschütz Öffnen EIN	A	12	BOOL	Address Output	EU 0203	Tor_AUF	St_Dr_An1		S2	Output	Sternschütz einschalten
15	Oeffn_Drei	Dreieckschütz Öffnen EIN	A	13	BOOL	Address Output	EU 0203	Tor_AUF	St_Dr_An1		S3	Output	Dreieckschütz einschalten
16	Oeffn_AUS	Öffnen ausgeschaltet			BOOL	Local Static	KOP nach LMOT_AUF	Tor_AUF	St_Dr_An1		Mot_aus	Output	Motor ausgeschaltet
17	T_An1_AUF	Anlauf-Timer Öffnen		T1	TON	Address Memory		Tor_AUF	St_Dr_An1		T_An1	Input	Anlauf-TIMER
18													
19	Tor_Schli	Tor schließen	M	10.1	BOOL	Address Memory	KOP von Koordination	Tor_ZU	St_Dr_An1		Mot_ein	Input	Motor einschalten
20	ferngest_ZU	Schließen ferngesteuert	M	11	BOOL	Address Memory	KOP von Koordination	Tor_ZU	St_Dr_An1		FgS	Input	Ferngesteuert schalten
21	VHd_Schli	Schließen von Hand	E	0.2	BOOL	Address Input	H5 0502	Tor_ZU	St_Dr_An1		VHd_Mot_ein	Input	Motor von Hand einschalten
22	NOT_Tast1	Schließen NOT-AUS	E	3.0	BOOL	Address Input	H2 0502	Tor_ZU	St_Dr_An1		NOTAUS	Input	NOT-Aus
23	I_ZU	Schaltwert zum Schließen erreicht	M	0.1	BOOL	Address Memory	KOP von LMOT_ZU	Tor_ZU	St_Dr_An1		I_Schalt	Input	Strom hat Weicherschaltwert erreicht
24	Schli_Haupt	Hauptschütz Schließen EIN	A	0.1	BOOL	Address Output	EU 0901	Tor_ZU	St_Dr_An1		S1	Output	Hauptschütz einschalten
25	Schli_Stern	Sternschütz Schließen EIN	A	0.2	BOOL	Address Output	EU 0901	Tor_ZU	St_Dr_An1		S2	Output	Sternschütz einschalten
26	Schli_Drei	Dreieckschütz Schließen EIN	A	0.3	BOOL	Address Output	EU 0901	Tor_ZU	St_Dr_An1		S3	Output	Dreieckschütz einschalten
27	Schli_AUS	Schließen ausgeschaltet			BOOL	Local Static	KOP nach LMOT_ZU	Tor_ZU	St_Dr_An1		Mot_aus	Output	Motor ausgeschaltet
28	T_An1_ZU	Anlauf-Timer Schließen		T2	TON	Address Memory		Tor_ZU	St_Dr_An1		T_An1	Input	Anlauf-TIMER
29													
30	L_AUF	Schaltwert zum Öffnen erreicht	M	0.0	BOOL	Address Memory	KOP nach Tor_AUF	LMOT_AUF	LMOT		I_Schalt	Output	Weicherschaltwert erreicht
31	(Dreh_AUS)	Öffnen ausgeschaltet			BOOL	Local Static	KOP von Tor_AUF	LMOT_AUF	LMOT		Mot_aus	Input	Motor ausgeschaltet
32	L_Soll_AUF	Soll-Stromwert beim Öffnen			INT	Constant	12345	LMOT_AUF	LMOT		I_Soll	Input	Weicherschaltwert
33	LIST_AUF	Ist-Stromwert beim Öffnen			PIE/252 INT	Address Input	EF 0101	LMOT_AUF	LMOT		LIST	Input	Motorstrom
34													
35	L_ZU	Schaltwert zum Schließen erreicht	M	0.1	BOOL	Address Memory	KOP nach Tor_ZU	LMOT_ZU	LMOT		I_Schalt	Output	Weicherschaltwert erreicht
36	(Schli_AUS)	Schließen ausgeschaltet			BOOL	Local Static	KOP von Tor_ZU	LMOT_ZU	LMOT		Mot_aus	Input	Motor ausgeschaltet
37	L_Soll_ZU	Soll-Stromwert beim Schließen			INT	Constant	12333	LMOT_ZU	LMOT		I_Soll	Input	Weicherschaltwert
38	LIST_ZU	Ist-Stromwert beim Schließen			PIE/256 INT	Address Input	EF 0201	LMOT_ZU	LMOT		LIST	Input	Motorstrom
39													

Es sind zwei Möglichkeiten bezüglich der Koppelvariablen, in der Spalte "Point of Automation" mit KOP [Sender-TP] nach [alle Empfänger-TP] gekennzeichnet, gangbar:

- Koppelvariable mit direkten Adressen, die alle mit USAGE = Address Memory deklariert wurden.
- Koppelvariable mit symbolischen Adressen, die alle in einem Programm gelten, können als "Local Static" deklariert werden. Damit entfallen die direkten Adressen aus der Variablenliste. Der Maschinen-Compiler ordnet den Symbolen interne Adressen zu. Diese Adressierung ist für den User die einfachste.

Achten Sie bei beiden Adressierungen darauf, daß Koppelungen [siehe Seite 170] nur funktionieren, wenn Sender und Empfänger den gleichen Namen haben. Direkte Adressen niemals verschiedenen Namen zuordnen, weil dies zu Konflikten führt!

Für das Beispielprojekt soll die folgende Liste importiert [siehe Seite 198] werden:

12.1.6 Variablenliste der Instanzen mit direkten Adressen

Koppelvariable sind mit "KopE" - Empfänger - und "KopS" - Sender - in der Adressenspalte gekennzeichnet.

....Teilprozeß....|...Symbolname...|.Adresse...|.Formalparam...|.Typ-PRAP

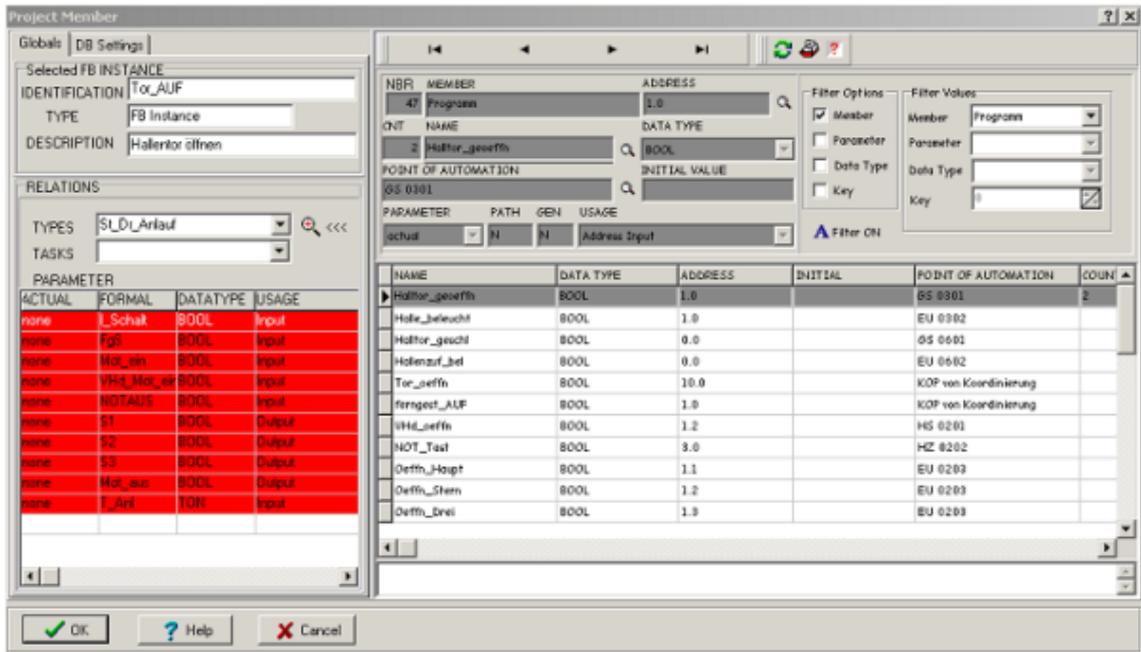
- ```
=====
```
- .Innenbel....|Halltor\_geoeffn...|.E1.0....|.La\_einschalten|...Lampe
  - .Innenbel....|.Halle\_beleuchtet...|.A1.0....|...La\_EIN.....|.Lampe
  - .Aussenbel...|Halltor\_geschl...|.E0.0....|.La\_einschalten|...Lampe
  - .Aussenbel...|.Hallenzuf\_bel...|.A0.0....|...La\_EIN.....|.Lampe
  - .Tor\_AUF.....|...Tor\_oeffn....|KopE::M10.0|...Mot\_ein....|.St\_Dr\_An1
  - .Tor\_AUF.....|.ferngest\_AUF..|KopE::M1.0|.....FgS.....|.St\_Dr\_An1
  - .Tor\_AUF.....|...VHd\_oeffn....|.E1.2....|.VHd\_Mot\_ein..|.St\_Dr\_An1
  - .Tor\_AUF.....|.NOT\_Tast\_AUF..|.E10.0...|...NOTAUS.....|.St\_Dr\_An1

- .Tor\_AUF.....|.....I\_AUF.....|KopE:M0.0|..I\_Schalt....|.St\_Dr\_AnI
- .Tor\_AUF.....|..Oeffn\_Haupt...|...A1.1....|.....S1.....|.St\_Dr\_AnI
- .Tor\_AUF.....|..Oeffn\_Stern...|...A1.2....|.....S2.....|.St\_Dr\_AnI
- .Tor\_AUF.....|..Oeffn\_Drei....|...A1.3....|.....S3.....|.St\_Dr\_AnI
- .Tor\_AUF.....|...T\_AnI\_AUF...|....T1.....|.....T\_AnI.....|.St\_Dr\_AnI
- .Tor\_AUF.....|...Oeffn\_AUS...|KopS:M0.2|..Motor\_aus....|.St\_Dr\_AnI
- .Tor\_ZU.....|...Tor\_Schli....|KopE:M10.1|...Mot\_ein....|.St\_Dr\_AnI
- .Tor\_ZU.....|...ferngest\_ZU..|KopE:M1.1|.....FgS.....|.St\_Dr\_AnI
- .Tor\_ZU.....|...VHd\_schli....|...E0.2....|.VHd\_Mot\_ein..|.St\_Dr\_AnI
- .Tor\_ZU.....|..NOT\_Tast\_ZU...|...E10.0...|...NOTAUS.....|.St\_Dr\_AnI
- .Tor\_ZU.....|.....I\_ZU.....|KopE:M0.1|..I\_Schalt....|.St\_Dr\_AnI
- .Tor\_ZU.....|..Schli\_Haupt...|...A0.1....|.....S1.....|.St\_Dr\_AnI
- .Tor\_ZU.....|..Schli\_Stern...|...A0.2....|.....S2.....|.St\_Dr\_AnI
- .Tor\_ZU.....|..Schli\_Drei....|...A0.3....|.....S3.....|.St\_Dr\_AnI
- .Tor\_ZU.....|...T\_AnI\_ZU.....|....T2.....|.....T\_AnI.....|.St\_Dr\_AnI
- .Tor\_ZU.....|...Schli\_AUS...|KopS:M0.3|..Motor\_aus....|.St\_Dr\_AnI
- .I\_MOT\_AUF...|.....I\_AUF.....|KopS:M0.0|..I\_Schalt....|...I\_MOT
- .I\_MOT\_AUF...|...Oeffn\_AUS...|KopE:M0.2|...Mot\_aus....|...I\_MOT
- .I\_MOT\_AUF...|...I\_Soll\_AUF...|...12345...|...I\_SOLL.....|...I\_MOT
- .I\_MOT\_AUF...|...I\_IST\_AUF...|...PEW.?...|.....I\_IST.....|...I\_MOT
- .I\_MOT\_ZU...|.....I\_ZU.....|KopS:M0.1|..I\_Schalt....|...I\_MOT
- .I\_MOT\_ZU...|...Schli\_AUS...|KopE:M0.3|...Mot\_aus....|...I\_MOT
- .I\_MOT\_ZU...|...I\_Soll\_Zu...|...12345...|...I\_SOLL.....|...I\_MOT
- .I\_MOT\_ZU...|...I\_IST\_ZU.....|...PEW.?...|.....I\_IST.....|...I\_MOT

## 12.1.7 Die Instanzen - Realisierung der Teilprozesse

Eine Instanz ist angelegt, wenn sie einem Member "PROGRAM" zugeordnet ist, wie es im PROGRAM und Instanzen [siehe Seite 79] dargestellt wurde. Vorausgesetzt, die erforderlichen Aktualparameter sind im PROGRAM-Type, hier Programm(FB-Instanzen), eingetragen, dann öffnen Sie mit Klick auf  den "Vorrat" an Variablen.

Am Beispiel des TP "Tor\_AUF", das bereits im Thema "Typ-Programm mit Instanzen ..." vorgestellt wurde, soll der Weg gezeigt werden. Im folgenden Bild sehen Sie die geöffnete ACCESS TABLE, Aktual-Variablentabelle, die zum Programm gehört (siehe Filter-Einstellung).



**Handeln Sie bitte wie folgt:** Die erste Formalvariable links oben heißt "FgS". Führen Sie den Cursor dahin und markieren mit Klick diese Zeile. Sehen Sie in Ihrer Excel-Variablen-tabelle<sup>20</sup> nach, unter den Spalten "H" bei "Tor\_AUF" und "J" bei "FgS" finden Sie in der Spalte "A" die Variable "ferngest\_AUF". Ebendiese Variable suchen Sie in der ACCESS TABLE.



**Tip**

ENTWEDER:

Den Cursor auf die ACCESS TABLE führen, rechte Maustaste und "Order By Name" klicken. Dann sind die Variablen alphabetisch geordnet, zuerst die Groß-, dann die Kleinbuchstaben. Das Suchen fällt dadurch leichter.

ODER:

Klick auf neben "NAME" und "ferngest" eingeben, um schnellstens die Variable zu finden.

Sie können natürlich auch nach dem Ordnen die Such-Hilfe verwenden.

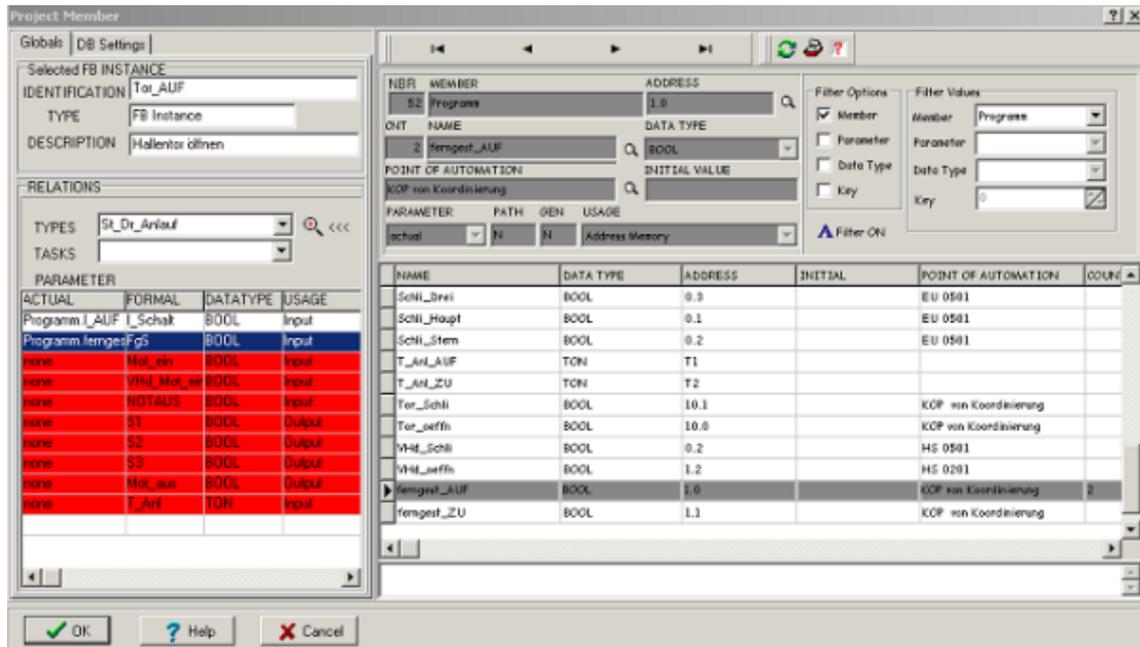
<sup>20</sup> Hinweis:

Die unter Spalte "A" Namen der aktuellen Parameter dürfen nur einmal importiert werden, weshalb die Wiederholungen eingeklammert wurden! Genauer: Nur die "Empfänger" der Koppelsignale sind eingeklammert, soweit es "Sender" gibt.

|    | A               | B                                 | C      | D     | E        | F              | G                    | H                   | I         | J              | K            | L                                    | M              |
|----|-----------------|-----------------------------------|--------|-------|----------|----------------|----------------------|---------------------|-----------|----------------|--------------|--------------------------------------|----------------|
|    | Act. Name       | Actual Statement                  | Prefix | Addr. | Qt. Type | USAGE          | Init. Val.           | Point of Automation | TP Name   | PRAP Name      | Form. Param. | USAGE                                | Formal Comment |
| 1  | Halltor_gewoefn | Halltor geöffnet                  | E      | 10    | BOOL     | Address Input  | GS 0308              | Innenbel            | Lampe     | Lx_einschalten | Input        | Schließbedingung erfüllt             |                |
| 2  | Halle_beleucht  | Halle beleuchtet                  | A      | 10    | BOOL     | Address Output | EU 9302              | Innenbel            | Lampe     | Lx_EIN         | Output       | Lampe einschalten                    |                |
| 3  | Halltor_geschl  | Halltor geschlossen               | E      | 00    | BOOL     | Address Input  | GS 0608              | Aussenbel           | Lampe     | Lx_einschalten | Input        | Schließbedingung erfüllt             |                |
| 4  | Hallenrout_bel  | Hallenrout beleuchten             | A      | 00    | BOOL     | Address Output | EU 9602              | Aussenbel           | Lampe     | Lx_EIN         | Output       | Lampe einschalten                    |                |
| 5  | Tor_oeffn       | Tor öffnen                        | M      | 10.0  | BOOL     | Address Memory | KDP von Koordination | Tor_AUF             | St_Dr_Anl | Mot_ein        | Input        | Motor einschalten                    |                |
| 6  | ferngest_AUF    | Öffnen ferngesteuert              | M      | 10    | BOOL     | Address Memory | KDP von Koordination | Tor_AUF             | St_Dr_Anl | FgS            | Input        | Ferngesteuert schalten               |                |
| 7  | YH4_oeffn       | Öffnen von Hand                   | E      | 12    | BOOL     | Address Input  | HS 0206              | Tor_AUF             | St_Dr_Anl | YH4_Mot_ein    | Input        | Motor von Hand einschalten           |                |
| 8  | NOT_Tast        | Schließen NOT-AUS                 | E      | 30    | BOOL     | Address Input  | HZ 0202              | Tor_AUF             | St_Dr_Anl | NOTAUS         | Input        | NOT-Aus                              |                |
| 9  | (I_AUF)         | Schaltwert zum Öffnen erreicht    | M      | 0.0   | BOOL     | Address Memory | KDP von L_MOT_AUF    | Tor_AUF             | St_Dr_Anl | I_Schalt       | Input        | Strom hat Weicherschaltwert erreicht |                |
| 10 | Oeffn_Haupt     | Hauptschütz Öffnen EIN            | A      | 11    | BOOL     | Address Output | EU 9203              | Tor_AUF             | St_Dr_Anl | S1             | Output       | Hauptschütz einschalten              |                |
| 11 | Oeffn_Stern     | Sternschütz Öffnen EIN            | A      | 12    | BOOL     | Address Output | EU 9203              | Tor_AUF             | St_Dr_Anl | S2             | Output       | Sternschütz einschalten              |                |
| 12 | Oeffn_Drei      | Dreieckschütz Öffnen EIN          | A      | 13    | BOOL     | Address Output | EU 9203              | Tor_AUF             | St_Dr_Anl | S3             | Output       | Dreieckschütz einschalten            |                |
| 13 | Oeffn_AUS       | Öffnen ausgeschaltet              |        |       | BOOL     | Local Status   | KDP nach L_MOT_AUF   | Tor_AUF             | St_Dr_Anl | Mot_aus        | Output       | Motor ausgeschaltet                  |                |
| 14 | T_Anl_AUF       | Anlaufzeit Öffnen                 | T1     |       | TON      | Address Memory |                      | Tor_AUF             | St_Dr_Anl | T_Anl          | Input        | Anlauf-TIMER                         |                |
| 15 | Tor_Schli       | Tor schließen                     | M      | 10.1  | BOOL     | Address Memory | KDP von Koordination | Tor_ZU              | St_Dr_Anl | Mot_ein        | Input        | Motor einschalten                    |                |
| 16 | ferngest_ZU     | Schließen ferngesteuert           | M      | 11    | BOOL     | Address Memory | KDP von Koordination | Tor_ZU              | St_Dr_Anl | FgS            | Input        | Ferngesteuert schalten               |                |
| 17 | YH4_Schli       | Schließen von Hand                | E      | 0.2   | BOOL     | Address Input  | HS 0608              | Tor_ZU              | St_Dr_Anl | YH4_Mot_ein    | Input        | Motor von Hand einschalten           |                |
| 18 | (NOT_Tast)      | Schließen NOT-AUS                 | E      | 30    | BOOL     | Address Input  | HZ 0602              | Tor_ZU              | St_Dr_Anl | NOTAUS         | Input        | NOT-Aus                              |                |
| 19 | (I_ZU)          | Schaltwert zum Schließen erreicht | M      | 0.1   | BOOL     | Address Memory | KDP von L_MOT_ZU     | Tor_ZU              | St_Dr_Anl | I_Schalt       | Input        | Strom hat Weicherschaltwert erreicht |                |
| 20 | Schli_Haupt     | Hauptschütz Schließen EIN         | A      | 0.1   | BOOL     | Address Output | EU 9501              | Tor_ZU              | St_Dr_Anl | S1             | Output       | Hauptschütz einschalten              |                |
| 21 | Schli_Stern     | Sternschütz Schließen EIN         | A      | 0.2   | BOOL     | Address Output | EU 9501              | Tor_ZU              | St_Dr_Anl | S2             | Output       | Sternschütz einschalten              |                |
| 22 | Schli_Drei      | Dreieckschütz Schließen EIN       | A      | 0.3   | BOOL     | Address Output | EU 9501              | Tor_ZU              | St_Dr_Anl | S3             | Output       | Dreieckschütz einschalten            |                |
| 23 | Schli_AUS       | Schließen ausgeschaltet           |        |       | BOOL     | Local Status   | KDP nach L_MOT_ZU    | Tor_ZU              | St_Dr_Anl | Mot_aus        | Output       | Motor ausgeschaltet                  |                |
| 24 | T_Anl_ZU        | Anlaufzeit Schließen              | T2     |       | TON      | Address Memory |                      | Tor_ZU              | St_Dr_Anl | T_Anl          | Input        | Anlauf-TIMER                         |                |
| 25 | I_AUF           | Schaltwert zum Öffnen erreicht    | M      | 0.0   | BOOL     | Address Memory | KDP nach Tor_AUF     | L_MOT_AUF           | L_MOT     | I_Schalt       | Output       | Weicherschaltwert erreicht           |                |
| 26 | (Oeffn_AUS)     | Öffnen ausgeschaltet              |        |       | BOOL     | Local Status   | KDP von Tor_AUF      | L_MOT_AUF           | L_MOT     | Mot_aus        | Output       | Motor ausgeschaltet                  |                |
| 27 | L_Soll_AUF      | Soll-Stromwert beim Öffnen        | INT    |       | Constant |                | 12345                | L_MOT_AUF           | L_MOT     | L_Soll         | Input        | Weicherschaltwert                    |                |
| 28 | I_LIST_AUF      | Ist-Stromwert beim Öffnen         | PEV252 |       | INT      | Address Input  | EF 0301              | L_MOT_AUF           | L_MOT     | I_LIST         | Input        | Motorstrom                           |                |
| 29 | L_ZU            | Schaltwert zum Schließen erreicht | M      | 0.1   | BOOL     | Address Memory | KDP nach Tor_ZU      | L_MOT_ZU            | L_MOT     | I_Schalt       | Output       | Weicherschaltwert erreicht           |                |
| 30 | (Schli_AUS)     | Schließen ausgeschaltet           |        |       | BOOL     | Local Status   | KDP von Tor_ZU       | L_MOT_ZU            | L_MOT     | Mot_aus        | Output       | Motor ausgeschaltet                  |                |
| 31 | L_Soll_ZU       | Soll-Stromwert beim Schließen     | INT    |       | Constant |                | 12323                | L_MOT_ZU            | L_MOT     | L_SOLL         | Input        | Weicherschaltwert                    |                |
| 32 | I_LIST_ZU       | Ist-Stromwert beim Schließen      | PEV250 |       | INT      | Address Input  | EF 0201              | L_MOT_ZU            | L_MOT     | I_LIST         | Input        | Motorstrom                           |                |

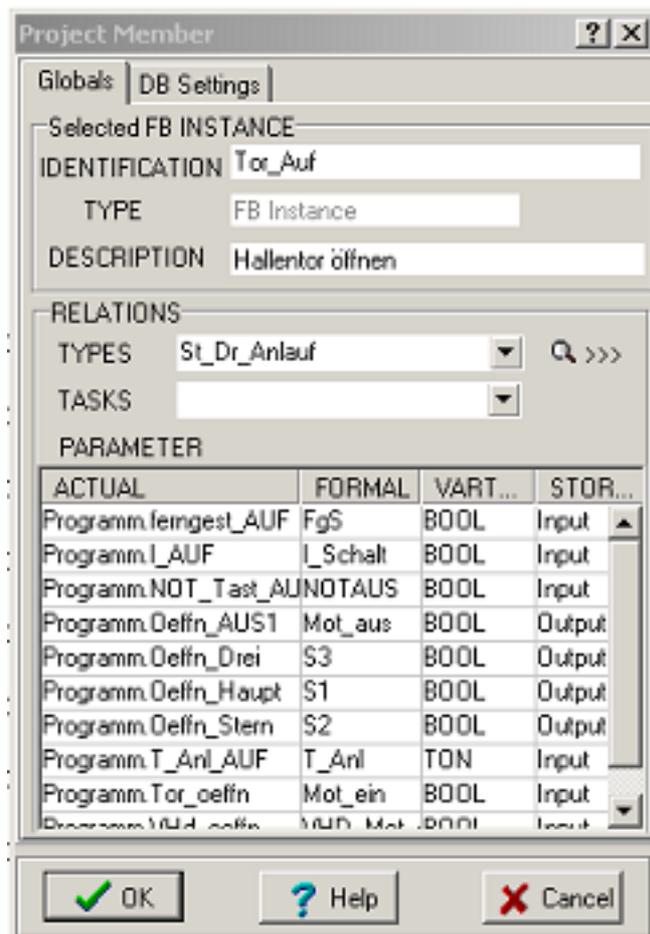
Nun geht es wie folgt weiter:

Nachdem Sie die Aktual-Variable "ferngest\_AUF" gefunden haben, Klick auf . Im folgenden Bild



sehen Sie den Erfolg der Handlung. Beachten Sie, es steht "Programm.ferngest\_AUF" eingetragen. Die Zuordnung zum Member [siehe Seite 172] Programm(FB-Instanzen) ist hier wichtig. Sie können auch Variable der Ressource einfügen. Dann würde deren Identification vor dem Punkte stehen.

Führen Sie diese Zuordnungsprozedur für alle Formal-Größen aus, dann ergibt sich das nächste Bild:



Übrigens erfolgt beim Markieren einer fertigen Zuordnungszeile, jede ist jetzt weiß, automatisch das Suchen der Variablen und das Anzeigen mit grüner Hinterlegung.

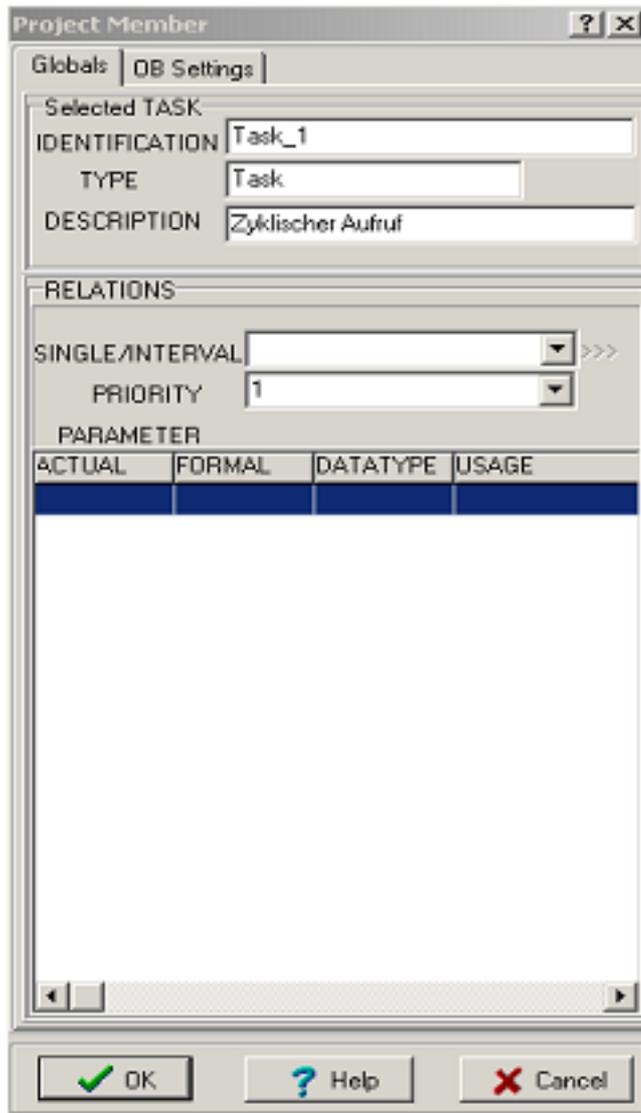
Schließlich sehen Sie noch ein Sheet "DB Settings", dort müssen Sie nichts hineinschreiben, wenn ein

Multiinstanzbaustein - wie hier im Beispiel - erzeugt werden soll.

## 12.1.8 Einfügen der Task in das Projekt

Allgemeines erfahren Sie unter Task einfügen [siehe Seite 55], demnach ist die Ressource zu markieren, danach Klick auf  und es ist das Formblatt auszufüllen (Klick auf "Help" hilft dabei!).

Das fertige Bild sehen Sie nachstehend:



The screenshot shows the 'Project Member' dialog box with the 'OB Settings' tab selected. The 'Selected TASK' section has the following fields:

- IDENTIFICATION: Task\_1
- TYPE: Task
- DESCRIPTION: Zyklischer Aufruf

The 'RELATIONS' section has the following fields:

- SINGLE/INTERVAL: (empty dropdown menu)
- PRIORITY: 1

The 'PARAMETER' section is a table with the following columns: ACTUAL, FORMAL, DATATYPE, USAGE. The table is currently empty.

At the bottom of the dialog box are three buttons: OK, Help, and Cancel.

Da das Beispielprojekt für die Sprache STEP 7 konzipiert ist, finden Sie noch das Blatt "OB Settings", aber zunächst müssen Sie die Task-Daten "Single/Interval" und "Priority" ausfüllen.

Nehmen Sie die direkte Hilfe in Anspruch (Fragezeichen) und wählen Sie für den OB1 die Priorität 1. Die richtige OB-Nummer wird automatisch eingetragen mit Klick auf "OK", dies beendet auch den Vorgang.



### Anmerkung

"OB Settings" dient auch zum Eintragen der Nummer, dennoch müssen die Task-Daten eingetragen werden. Aber bei OB mit gleicher Priorität muß noch die OB-Nummer angegeben werden.

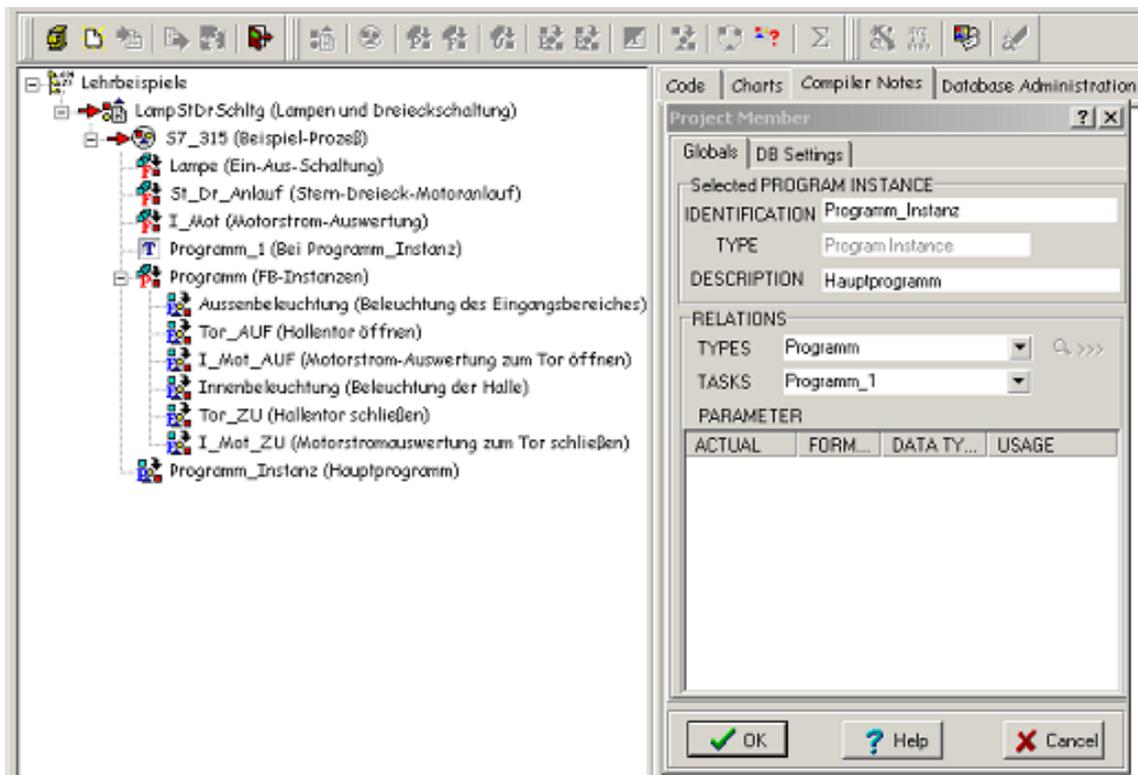
## 12.1.9 Die Programm-Instanz im Projekt

Allgemeines lesen Sie unter: Programm-Instanz einfügen [siehe Seite 60]

Hier für das Beispielprojekt ist alles sehr einfach.

- Ressource markieren
- Klick auf 
- In das Formblatt des Projekt Member "PROGRAM INSTANCE" tragen Sie Identification und Description ein. Beachten Sie bitte die zugelassenen Zeichen Hilfe mit  oder Klick auf 
- Wählen Sie den zugehörigen Programm-Typ und die Task. Hier kommen nur "Programm" und "Programm\_1" in Frage
- Klick auf "DB Settings" und Eintragen der Datenbaustein-Nummer, hier im Beispiel "10", weil FB10 für den Multi-Instanzbaustein eingesetzt wurde.

Das Ergebnis sehen Sie im folgenden Bild:



Auf der linken Seite erkennen Sie den kompletten Projektbaum, auf der rechten Seite das Formblatt für die Programm-Instanz.

## 12.2 Stern-Dreieck-Aufschaltung

### 12.2.1 Stern-Dreieck-Anlaufschaltung (Einführung)

Sie können von hier aus zu allen Teilen dieses Themas springen:

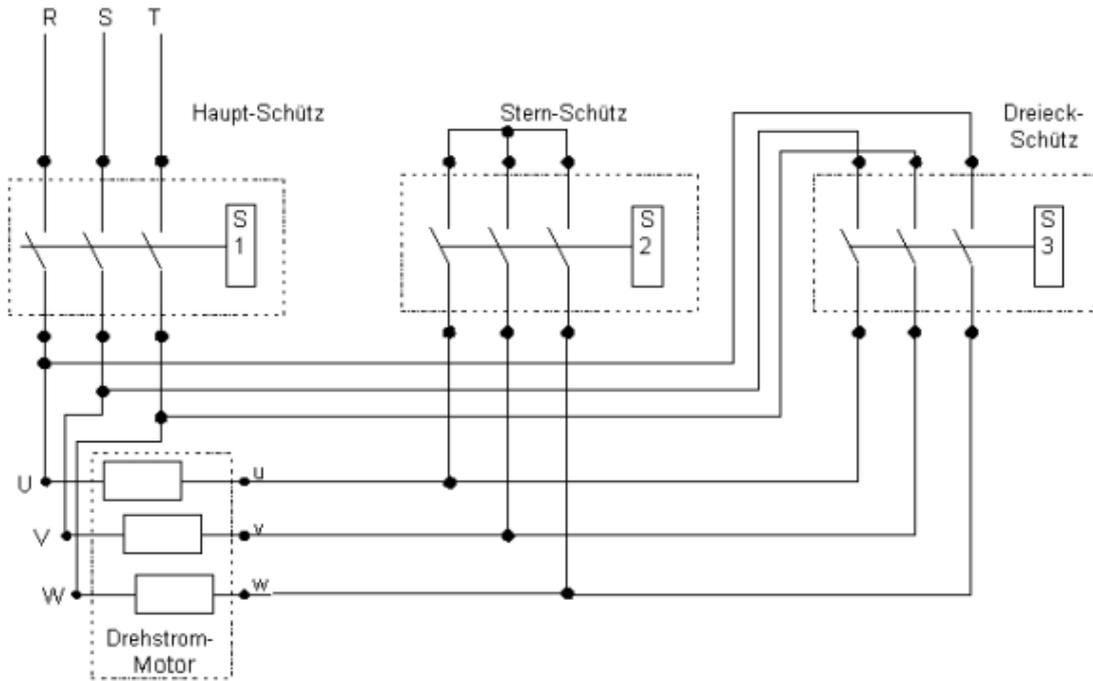
PRAP Stern-Dreieck-Steuerung (Teil 1) [siehe Seite 90]

PRAP Stern-Dreieck-Steuerung (Teil 2) [siehe Seite 92]

PRAP Stern-Dreieck-Steuerung (Teil 3) [siehe Seite 94]

Projektieren des Zeitgliedes "Wartezeit" [siehe Seite 98]

Motorstrom-Verarbeitung (Projektierung der Vergleichsfunktion) [siehe Seite 102]



Aus der Schaltung sind das Hauptschütz, das Sternschütz und das Dreieckschütz zu erkennen. Für Anwender des PRAP-Editors, denen Sinn und Zweck dieser Schaltung fremd sind, ist eine kurze Erklärung<sup>21</sup> beigefügt. Ebenso wie bei dem Thema Lampenschaltung [siehe Seite 105] beginnt die binäre Prozeßanalyse damit, die Variablen zu bestimmen.

#### Formulierung der Operationsvariablen.....Kurzzeichen

- Hauptschütz einschalten/ausschalten.....S1
- Sternschütz einschalten/ausschalten.....S2
- Dreieckschütz ein-/ausschalten.....S3

**Bemerkung:** Aus dem Schaltbild sind nur diese drei zu entnehmen. Sollten noch z.B. Signalisierungen an die Warte gewünscht werden, so sind entsprechende Ergänzungen erforderlich. Das Beispiel soll aber einfach bleiben.

#### Formulierung der Prozeßvariablen.....Kurzzeichen

- Ferngesteuert schalten.....FgS
- Motor einschalten.....Mot\_ein
- Motor von Hand einschalten.....VHd\_Mot\_ein
- Strom hat Weiterschaltwert erreicht..I\_schalt
- NOT-Aus.....NOTAUS

#### Bemerkungen:

<sup>21</sup>

Ein Drehstrom-Motor enthält drei Spulen-Systeme, drei Wicklungen, drei Phasen. Die Anschlüsse der Wicklungen am Motor sind in der Schaltung mit U - u, V - v und W - w gekennzeichnet, die Wicklungen selbst durch drei Rechtecke. Da Drehstrom aus drei um 120 Grad versetzten, gleich hohen Wechselspannungen besteht, entsteht in einem ebenso um 120 Grad versetzt gewickelten Motorenhäuser durch die drei Wicklungen ein Dreh-Magnetfeld. Die Stärke des Magnetfeldes hängt von der Höhe (Größe) des Stromes ab, der durch jede Wicklung fließt.

Die Drehstrom-Netzanschlüsse R, S und T werden mit den Motoranschlüssen U, V und W verbunden.

Im **Sternbetrieb** sind die Anschlüsse u, v und w miteinander verbunden, wodurch die Wechselspannung zwischen R und S einen Strom durch die Wicklungen U-u und V-v treibt, die zwischen S und T einen Strom durch V-v und W-w, die zwischen T und R durch W-w und U-u.

Im **Dreieckbetrieb** sind u mit S, v mit T und w mit R verbunden, die Wechselspannung zwischen R und S treibt jetzt einen Strom **nur** durch U-u, der deshalb ca. 73% höher ist.

Um das Netz beim Einschalten nicht mit einem zu hohen Anlaufstrom zu belasten, wird zunächst mit Sternbetrieb eingeschaltet und nach Abklingen des Einschaltstromstoßes auf Dreieckbetrieb umgeschaltet.

Die ersten beiden Variablen "liefern" andere Teilprozesse (oder auch nur einer), sie stellen damit eine Verbindung zu diesen "Informationsquellen" her. Solche Variable werden Koppelvariable [siehe Seite 170] genannt.

Wenn nicht ferngesteuert geschaltet wird, dann vor Ort von Hand z.B. mit einem Taster. Zu diesem Taster und seiner Dynamik [siehe Seite 150] ist später noch zu sprechen.



### Anmerkung

Die Weiterschaltbedingung vom Stern- in den Dreieckbetrieb liefert ein Strommesser, der allerdings nicht im Schaltbild eingetragen ist. Sie müssen als Entwerfer des PRAP mit Ihrem Auftraggeber daher verhandeln! (siehe Verhandlungs-Themen<sup>22</sup>)

Vielleicht will der Auftraggeber eine Zeitsteuerung haben, also nach z.B. 5 Sekunden Anlauf einfach weiterschalten wollen. Dann muß er aber dafür die Verantwortung übernehmen, falls die Weiterschaltung zu Störungen führt.

Die NOT-Aus-Variable ist mit eingefügt, weil es Notsituationen geben kann, die zum sofortigen Abschalten zwingen. Auch sie kann eine Koppelvariable sein aber auch ein Taster.

Eine dem **Stern-Dreieck-Anlauf übergeordnete Steuerung** (siehe Vorbemerkungen zum Beispielprojekt [siehe Seite 73]) soll bewirken, daß das Hallentor zwei Motore steuern: Antrieb "Tor\_AUF" (Torbewegung z.B. von links nach rechts) und der Antrieb "Tor\_ZU" (Torbewegung z.B. von rechts nach links). Natürlich müssen dort verschiedene Probleme bedacht werden wie z.B. das gleichzeitige Einschalten beider Motore zu verhindern. Das soll hier nicht behandelt werden, vielmehr sind die Variablen zu den

#### Instanzen "Tor\_AUF" und "Tor\_ZU"

zu benennen:

...St\_Dr\_Anlauf..|.Tor\_AUF...|.Tor\_ZU

=====

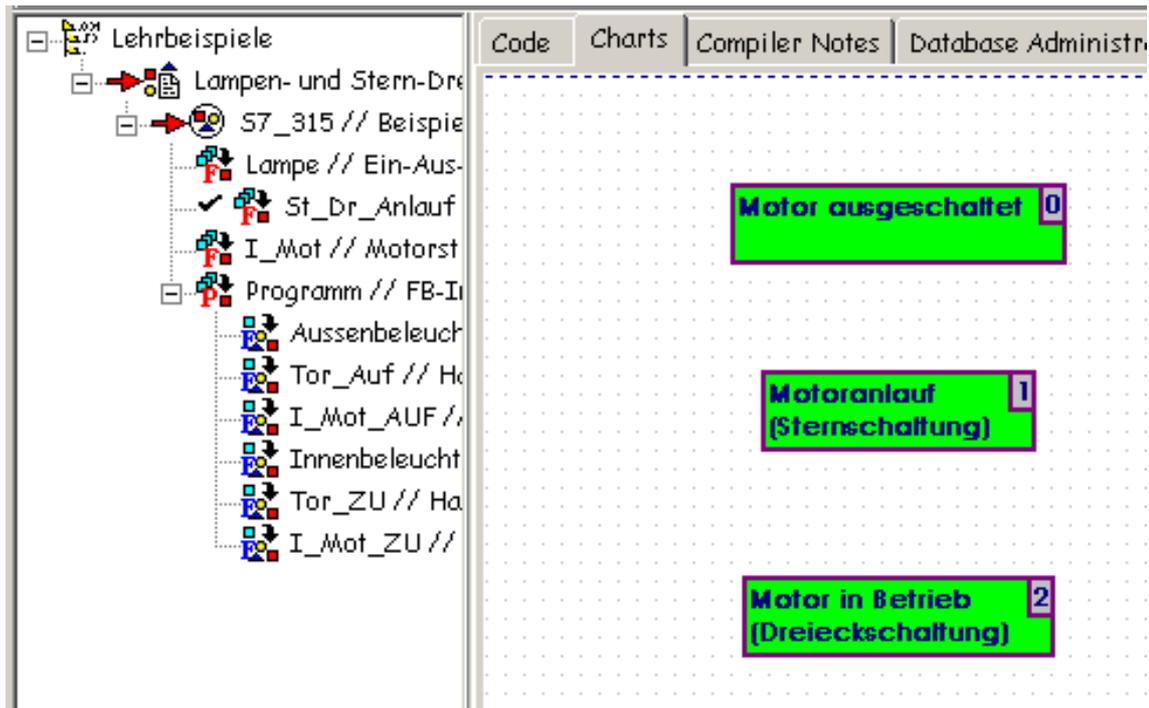
- ...Mot\_ein...|.Tor\_oeffn..|.Tor\_schli
- .....FgS.....|.ferngest\_AUF|.ferngest\_ZU
- .VHd\_Mot\_ein..|.VHd\_oeffn..|.VHd\_schli
- ...I\_Schalt...|.I\_AUF.....|.I\_ZU
- ...NOTAUS.....|.NOT\_Tast\_AUF|.NOT\_Tast\_ZU
- .....S1.....|.Oeffn\_Haupt|.Schli\_Haupt
- .....S2.....|.Oeffn\_Stern|.Schli\_Stern
- .....S3.....|.Oeffn\_Drei..|.Schli\_Drei
- ..Motor\_aus...|.Oeffn\_Aus..|.Schli\_Aus

Die Weiterschaltbedingungen beim Öffnen (I\_AUF) und beim Schließen (I\_ZU) des Tores sollen von der Stromerfassung der Motore stammen. Dieses Problem wird nochmals gegen Ende des Themas PRAP Stern-Dreieck-Steuerung (Teil 3) [siehe Seite 94] aufgegriffen.

Weiter mit PRAP Stern-Dreieck-Steuerung (Teil 1) [siehe Seite 90]

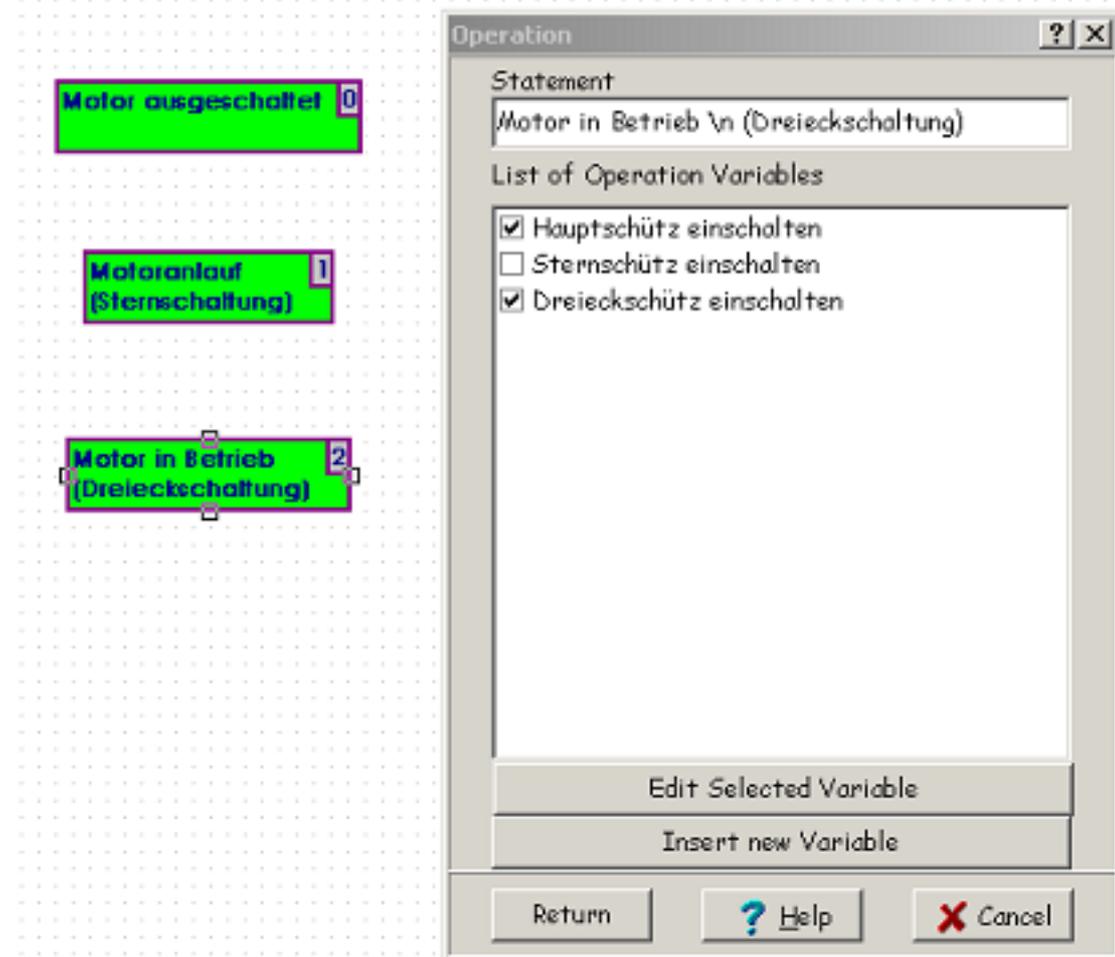
## 12.2.2 PRAP Stern-Dreieck-Steuerung (Teil 1)

<sup>22</sup> Die Strom-Meßeinrichtung muß Ihnen entweder einen Analogwert (Einheits-Strom oder Einheitsspannung) liefern, den Sie dann über einen Analog-Digitalwandler weiterverarbeiten oder Sie erhalten schon das Vergleichsergebnis mit dem Soll-Stromwert als Binärgröße. Im ersten Fall müssen Sie eine Vergleichsfunktion mit im PRAP projektieren, im zweiten Fall genügt ein Binär-Eingang.  
Aber damit ist es nicht getan! Beachten Sie diese Diskussion in allen Teilen zu diesem Beispiel.



Die Handhabung des PRAP-Editors zum Herstellen der drei Operationen entnehmen Sie bitte aus der Lampenschaltung (Teil 1/2) [siehe Seite 106].

Im folgenden Bild sehen Sie die Wertebelegung, den Binärvektor, der Operation "Motor in Betrieb".



Die Handhabung der Datenbank, um zu den Eintragungen der Operationsvariablen zu gelangen, finden Sie in dem Beispiel Lampenschaltung. [siehe Seite 110]

Der Datenbankeintrag für alle bisher im Projekt Lampen- und Stern-Dreieck-Schaltung eingetragenen

Variablen finden Sie im nächsten Bild. Allerdings ist die Operationsvariable "Hauptschütz einschalten" ausgewählt, zu der das Kurzzeichen (=Name) S1 gehört.



### Anmerkung

Es ist streng darauf zu achten, daß die richtigen Kurzzeichen den Statements zugeordnet werden.

| NBR | MEMBER       | ADDRESS | DATA TYPE | INITIAL VALUE |
|-----|--------------|---------|-----------|---------------|
| 82  | St_Dr_Anlaut |         | BOOL      |               |

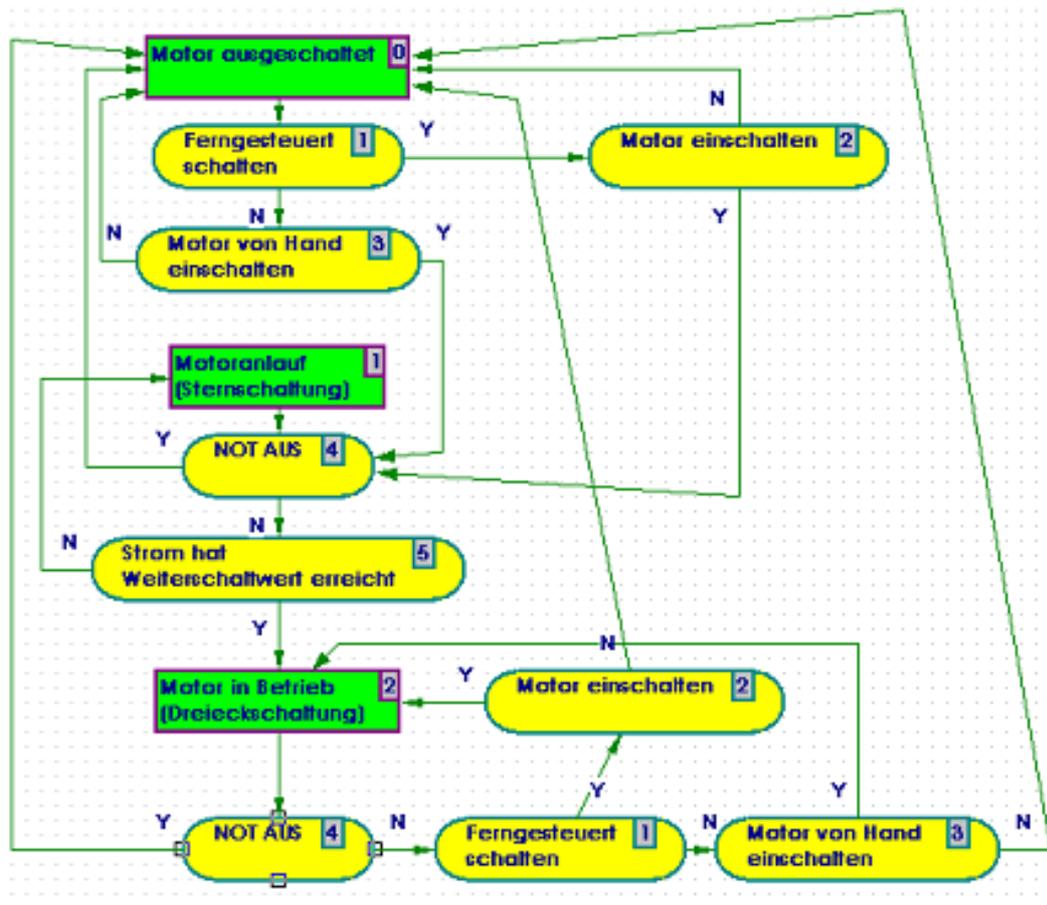
| PARAMETER | PATH | GEN | USAGE  |
|-----------|------|-----|--------|
| Formal    | N    | N   | Output |

| NAME | DATA TYPE | ADDRESS | INITIAL | POINT OF AUTOMATION | COUNTER |
|------|-----------|---------|---------|---------------------|---------|
| S1   | BOOL      |         |         |                     |         |

Sie sehen einen Eintrag EU xxxx in "Point of Automation" (=EMSR-Stellen nach DIN 19227). EU bedeutet E=Elektrisch, U=Antrieb von mehreren Variablen beeinflusst. Weil der Typ-PRAP keinen Bezug zur konkreten Anlage hat, sind xxxx anstelle einer konkreten EMSR-Stelle geschrieben. Instanzen müssen dann echte Bezeichner haben.

Weiter mit PRAP Stern-Dreieck-Steuerung (Teil 2) [siehe Seite 92]

## 12.2.3 PRAP Stern-Dreieck-Steuerung (Teil 2)



Die schrittweise Weiterentwicklung des PRAP erfolgt genau so wie in Lampenschaltung (Teil 2) [siehe Seite 110] und Lampenschaltung (Teil 3) [siehe Seite 113] beschrieben wurde, nur daß jetzt fünf P-Variable den Steuerungsablauf bestimmen, sie wurden im Einführungsteil [siehe Seite 88] beschrieben. Das Bild des PRAP ist so gezeichnet, daß jede Kante [siehe Seite 217] in ihrem Verlauf einzeln betrachtet bzw. verfolgt werden kann. In der Endausführung kann man parallele Kanten übereinanderlegen, wie das Bild im Teil 3 zeigt. Zunächst aber zur Steuerungs-Funktion des PRAP im oberen Teil. Der Motor bleibt ausgeschaltet, solange weder von Hand noch ferngesteuert der Motor eingeschaltet werden soll. Der Motor-Anlauf wird erreicht, sobald entweder bei Fernsteuerung der Motor eingeschaltet werden soll oder von Hand (Taster, wie oben erwähnt).

Aber hier tritt die erste Besonderheit auf:

Beide Yes-Ausgänge der P-Variablen führen zu "NOT-Aus", und falls dort Y gilt, bleibt der Motor ebenfalls ausgeschaltet.



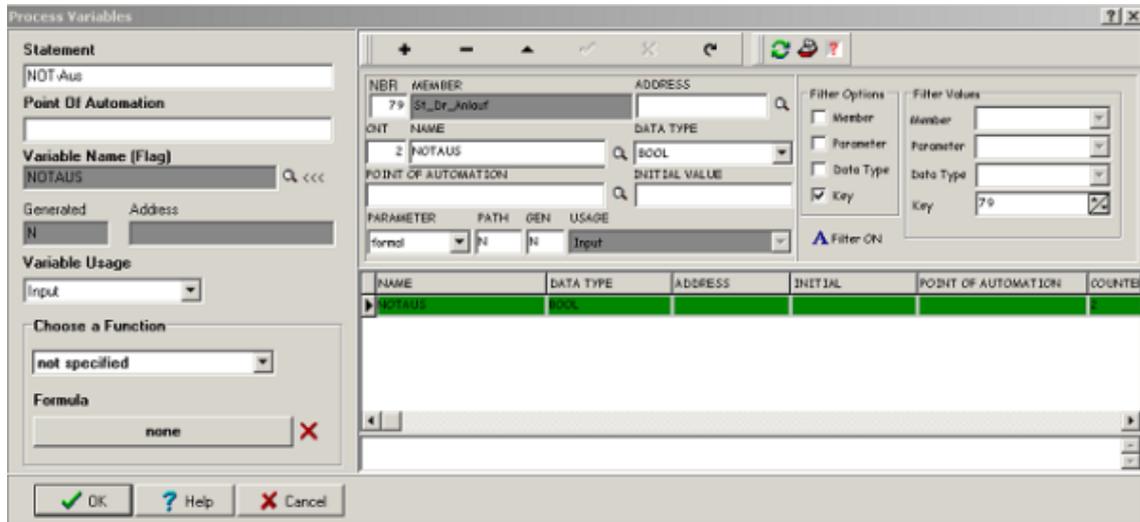
### Anmerkung

Würden die beiden Kanten nicht zu NOT-Aus führen sondern zur Operation "Motor-Anlauf (Sternschaltung)", dann würde zwar der Motor auch nicht anlaufen, aber dieser Endeffekt führte stets erst über die Operation "Motor-Anlauf", die dann doch nicht stabil erreicht [siehe Seite 139] wird!

Natürlich kann die P-Variable "NOT-Aus" auch VOR der Operation "Motor-Anlauf" eingetragen werden und mit ihrer No-Kante den Anlauf erwirken. Dann sollte man aber dennoch diese P-Variable nochmal - wie sie schon jetzt steht - einsetzen, weil während des Anlaufes das "NOT-Aus" auch funktionieren muß. Es handelt sich also bei der vorgelegten Lösung um eine Vereinfachung des PRAP, aber strukturstabil [siehe Seite 189] ist die zweite Situation {Op. Motor-Anlauf mit Stabilitätspfad: (NOT\_Aus = N) & (Strom hat Weiterschaltwert erreicht = N)} nicht, weil die Erfüllung des Stabilitätspfad von der No-Aussage des Stromgrenzwertes abhängt. Im Teil 3 wird das Problem weiterbehandelt! Lesen Sie eventuell

nochmal über die Prüfungen des PRAP [siehe Seite 64] nach.

Die zu den drei O-Variablen S1, S2 und S3 hinzugekommenen fünf P\_Variablen (vergl. <sup>23</sup>) sehen Sie im nachfolgenden Bild, allerdings fehlt S1, da der Darstellungsausschnitt dies verdeckt.



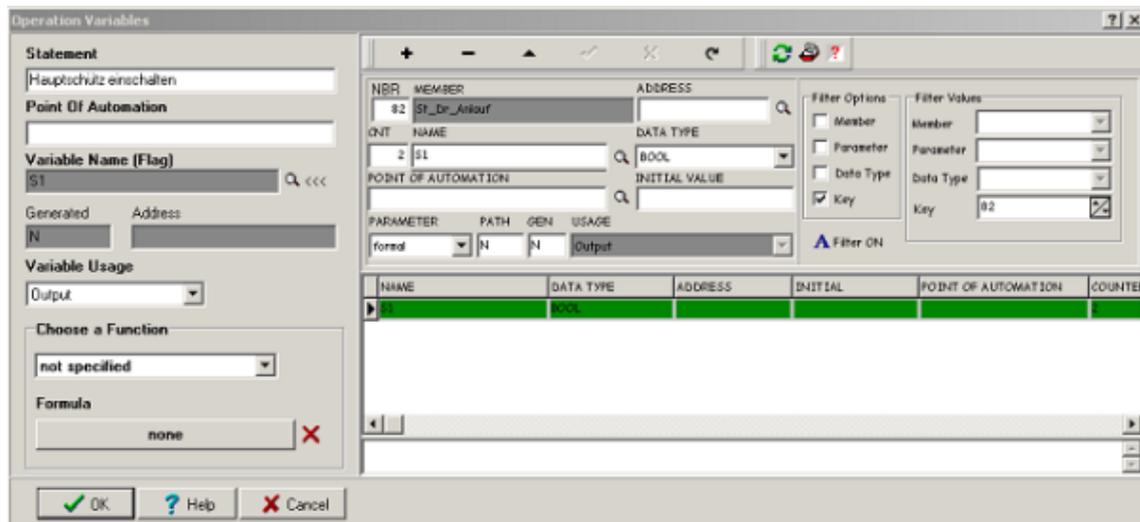
Es sind drei EMSR-Stellenbezeichnungen dazugekommen, HZ xxxx und HS xxxx für NOT-AUS und Handsteuerung. EF xxxx bedeutet gemäß DIN 19227 Elektrische Größe E und Durchfluß F erfassen, so daß F hier nicht z.B. als Gasströmung sondern als elektrischer Strom aufgefaßt wird. Die xxxx stehen anstelle der konkreten EMSR-Stellen in den Instanzen. Es sei hier an das Informationsfluß-Modell [siehe Seite 14] erinnert, in dem die Informations-Eingabe-Einheit genau das meint, was hier bezüglich Strommessung zu erfüllen ist. Sie können sich auch im Einführungsabschnitt [siehe Seite 88] die Überlegungen hierzu ansehen.

Weiter zu PRAP Stern-Dreieck-Steuerung (Teil 3) [siehe Seite 94]

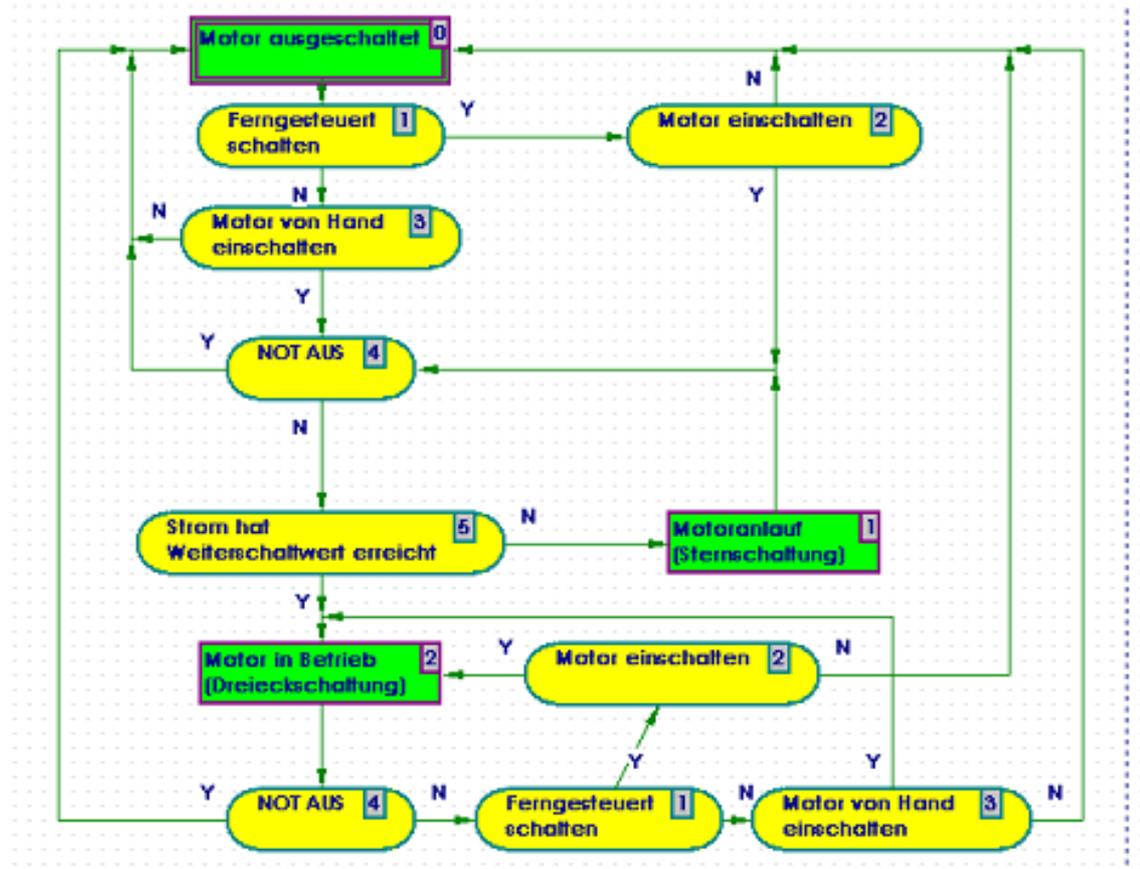
## 12.2.4 PRAP Stern-Dreieck-Steuerung (Teil 3)

In diesem Teil soll zuerst die Umzeichnung des PRAP gegenüber dem Bild in Teil 2 besprochen werden. Danach folgt eine Betrachtung zur Dynamik der P-Variablen, wie im Einführungsteil angekündigt, und damit auch die Sinnfälligkeit der Einführung einer weiteren P-Variablen.

23



94



Es gibt zwei beachtliche Unterschiede in der Gestaltung des PRAP gegenüber dem im Teil 2. Zuerst fällt natürlich auf, daß alle Kanten rechteckig und so angeordnet sind, daß sich alle überdecken, die das gleiche Ziel erreichen.



### Tipp

Wenn Sie Schwierigkeiten mit dem Geradezeichnen haben, dann halten Sie sich an das Raster-Gitter, und eventuell schalten Sie mit Zoom auf 200% (rechte Maustaste im freien Feld).

Schwerwiegender ist die Veränderung bei der Operation "Motor-Anlauf". Die Stabilitätsbetrachtungen im Teil 2 [siehe Seite 92] können zum Umzeichnen des PRAP genutzt werden. Nun ist deutlich zu sehen, daß die P-Variable "NOT-Aus" mit zur ersten Situation [siehe Seite 183] gehört. Die Durchlaufgefah zur Operation "Motor in Betrieb" im Fall einer falschen Aussage zum Stromwert ist nun noch deutlicher zu erkennen.

### Dynamik der Variablen "Strom hat Weiterschaltwert erreicht:"

Immerhin ist darüber nachzudenken, daß mit dem Einschalten des Haupt- und des Sternschützes der Stromanstieg erst mit Zeitverzögerung durch die Motorwicklung und die Meßeinrichtung übertragen wird. Jede Steuereinrichtung (SPS) ist schneller und damit ist die Fehlfunktion bereits abzusehen. Erst dann, wenn der Stromwert die Größe des Weiterschaltwertes überschritten hat, und das kann je nach Motorgröße einige zig Millisekunden dauern, beginnt die Aussage der P-Variablen wichtig zu werden.

### Lösungsmöglichkeiten des Durchlaufproblems

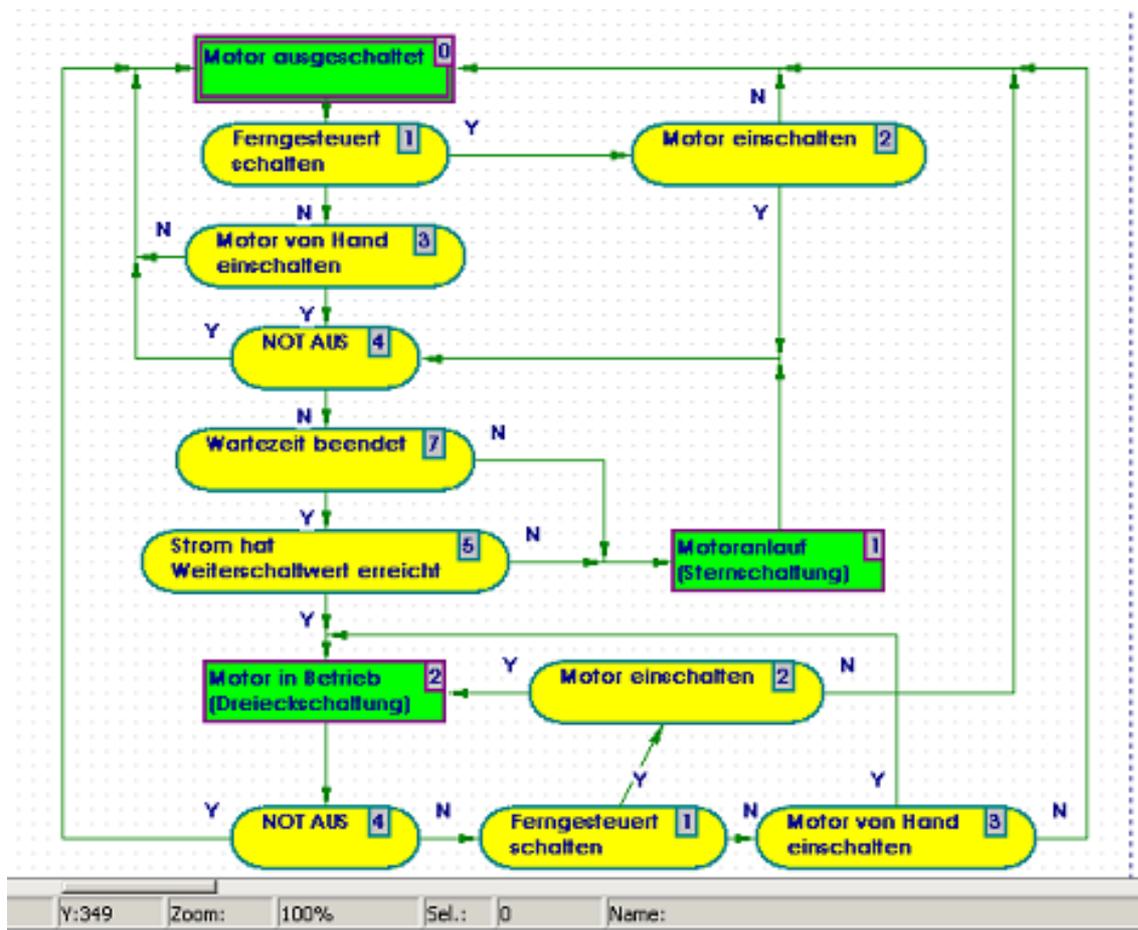
- Sie vereinbaren mit Ihrem Auftraggeber die Bereitstellung des Analogwertes und projektieren eine Eingangsbaugruppe zur Analog-Digital-Wandlung. Diese Möglichkeit wurde bereits im Einführungsteil erwähnt. Sie programmieren im PRAP nicht nur den Weiterschaltwert sondern zusätzlich einen Maximalwert als Vergleichsfunktion und führen eine neue P-Variable<sup>24</sup> "Maximalstrom erreicht" (Kurzzeichen:  $I_{max}$ ) ein.

<sup>24</sup> Sie schätzen die Anlaufdauer, bis zur Überschreitung des Weiterschaltwertes auf z.B. 250ms und  
Neue P-Variable "Maximalstrom erreicht" in der Situation Motor-Anlauf

programmieren ein Zeitglied so, daß erst nach Ablauf dieser Zeit nach dem Unterschreiten des Weiterschaltstromwertes "gefragt" wird. (Diese Lösungsvariante wird weiterverfolgt!)

- Sie verzichten darauf, den relativ unsicheren Wert des Maximalstromes einzuführen. Dafür verfolgen Sie den Stromverlauf einfach mit einem PRAP-Programm, das dann an sich zur Informations-Eingabe-Einheit [siehe Seite 14] gehört. Aber es muß natürlich hier im Projektbaum erscheinen.
- Sie sind sich mit Ihrem Auftraggeber einig, daß überhaupt keine Erfassung des Anlaufstromverhaltens erforderlich ist und schalten nach z.B. 2s weiter auf Dreieck-Betrieb.

Der Lösungsvorschlag 2. hat zum Vorteil gegenüber Vorschlag 1., daß Sie für die Erfassung des Weiterschaltwertes auf ein Binärsignal zugreifen können (siehe auch in der Einführung [siehe Seite 88]), falls Sie das so mit dem Auftraggeber vereinbart haben.



Sie sehen im PRAP die einfache Veränderung durch Einschieben der P\_Variablen<sup>25</sup> "Wartezeit beendet". Solange dies nicht der Fall ist, bleibt der Stromwert unberücksichtigt. Nach der Wartezeit wird angenommen, daß der Weiterschaltwert überschritten ist und auf das Erreichen "von oben her" gewartet wird. Es ist dazu eine Vergleichsfunktion zu projektieren. Im Thema Motorstrom-Verarbeitung [siehe Seite 102] finden Sie eine Lösung ganz anderer Art. Dann kann auch auf das Zeitglied verzichtet werden. "Strom hat Weiterschaltwert erreicht" wird dort ebenfalls durch Vergleich des Motorstromes mit dem Weiterschaltwert als Sollgröße ermittelt, bitte sehen Sie sich das selbst dort an.

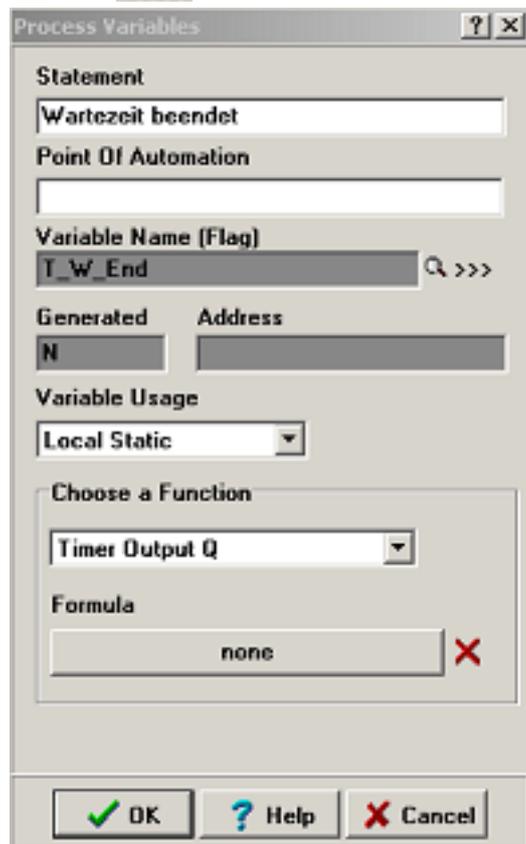
Die dritte Situation "Motor in Betrieb (Dreieckschaltung)" ist nur dann stabil, wenn "Motor einschalten" = Yes oder "Motor von Hand einschalten" = Yes gilt. Bei der Einführung dieser Variablen wurde schon auf die Dynamik dieser Signale verwiesen. Sind beide P\_Variablen statisch [siehe Seite 188], dann sollten sie auch während des Motor-Anlaufes ihre Wertung Yes behalten, andererseits erfolgt mit dem Übergang vom Anlauf in den Betrieb das Abschalten statt! Das will aber niemand. Für das Signal "Motor von Hand einschalten" bedeutet das, es MUSS ein rastender Taster verwendet werden (oder ein Ein-Aus-Schalter)! Falls Sie das nicht wollen, müssen Sie eine weitere Variable "Motor von Hand ausschalten" einführen

und in der dritten Situation anstelle "... einschalten" verwenden. Dann dürfen beide Hand-Taster als dynamisch [siehe Seite 150] gelten.

Zu Ihrer eigenen Sicherheit und zur Information Dritter hat sich sehr bewährt, die **PRAP-Funktion** zu **beschreiben**. Stellen Sie den Cursor auf ein freies Feld außerhalb der PRAP-Symbole und klicken auf die rechte Maustaste; danach Klick auf "Descriptions". Nun können Sie Ihren Beschreibungstext eintragen. Als Beispiel sehen Sie den Text des PRAP-Autors:

<sup>25</sup> Das Einführen der P-Variablen "Wartezeit beendet" beginnt damit, zunächst "Statement"="Wartezeit beendet" einzutragen. "Variable Usage"="Local Static" ist hier sinnvoll zu wählen. Die Zuordnung zum Timer - Klick auf "nothing" - hat natürlich nur Sinn, wenn ein dazugehöriges Zeitglied existiert, aber "Choose a Function"="Timer Output Q" kann schon erfolgen.

Mit Klick auf   können Sie "Name"="T\_W\_End" als Boolesche Variable eintragen. Das Ergebnis sehen Sie im folgenden Bild:



Process Variables

Statement  
Wartezeit beendet

Point Of Automation

Variable Name (Flag)  
T\_W\_End  

| Generated | Address |
|-----------|---------|
| N         |         |

Variable Usage  
Local Static

Choose a Function  
Timer Output Q

Formula  
none 

 OK  Help  Cancel

Zum erfolgreichen Abschluß der Tätigkeit unbedingt  anklicken!

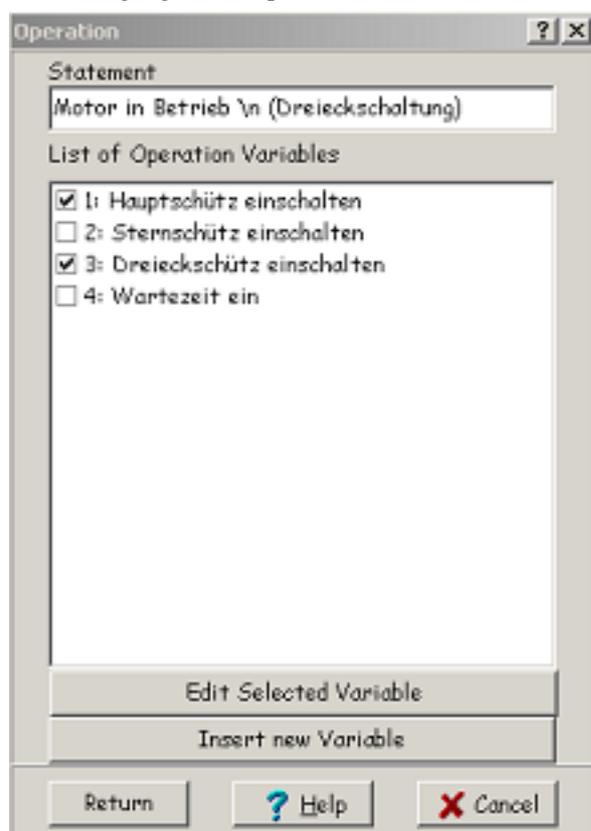


Weiter mit Projektieren des Zeitgliedes "Wartezeit" [siehe Seite 98]

## 12.2.5 Projektieren des Zeitgliedes "Wartezeit"

Lesen Sie hierzu die Ausführungen zu Zeitgliedern/Timern [siehe Seite 26].

Die Zuordnung der P-Variablen "Wartezeit beendet" zum Timer - Klick auf "nothing" - hat natürlich nur Sinn, wenn ein dazugehöriges Zeitglied existiert. Deshalb sollten Sie zuerst die O-Variable "Wartezeit ein" im PRAP einführen und das Zeitglied konfigurieren. Im nachfolgenden Bild sehen Sie die Wertebelegung in der Operation "Motor-Anlauf (Sternschaltung)".

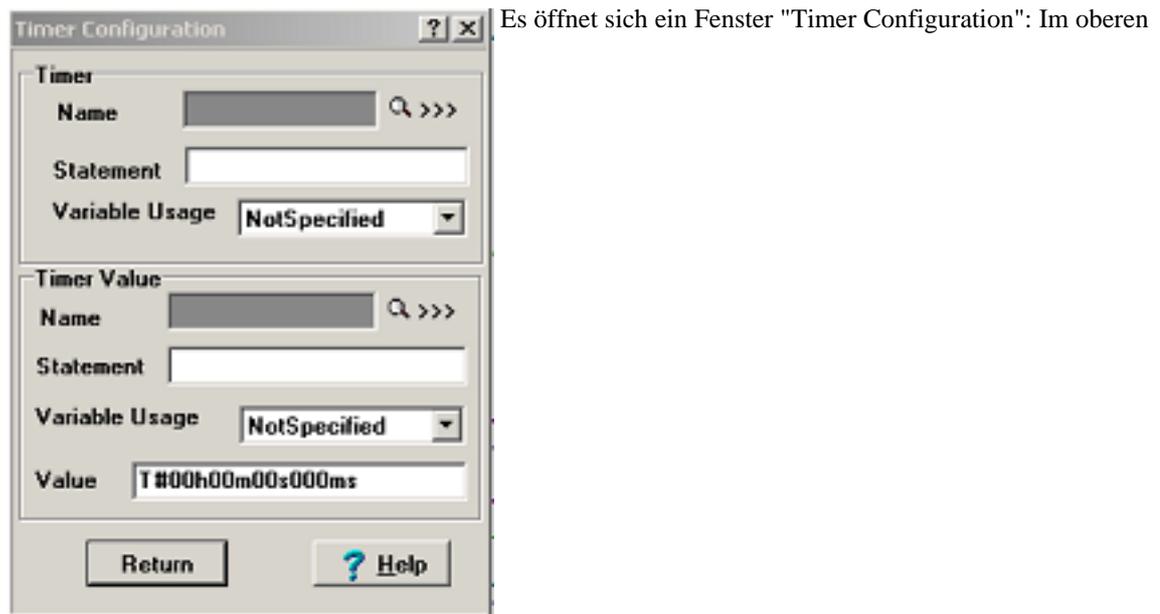


Natürlich beginnt man mit einem Klick auf "Insert

New Variable" und erhält die Möglichkeiten der Eingaben, wie Sie es schon kennen (eventuell in der Lampensteuerung [siehe Seite 106] nachlesen).

Nach dem Doppelklick auf "New\_Operation\_Variable" oder Klick auf "Edit Selected Variable" öffnet sich das Fenster "Operation Variables" zur Eingabe der notwendigen Daten. Hier handelt es sich um die Operationsvariable "Wartezeit ein", die im statischen Merkerbereich liegen soll, also "Local Static" bei Variable Usage wählen. Schaffen Sie sich Zugang zur Datenbank mit Klick auf  >>>. Tragen Sie bei Name z.B. "T\_W" ein als Kurzzeichen zu "Wartezeit ein". Der Datentyp muß Boole sein. Danach wieder  <<< anklicken.

Im unteren Teil sehen Sie den Bereich "Choose a Function" ( eine Funktion wählen), das erste Feld bietet Ihnen alle möglichen an. Wählen Sie "Create Timer". Danach ein Klick auf das Feld "nothing" oder "none" (soviel wie Formel oder Formelausdruck nicht vorhanden).



Bereich "Timer" wird "Name" - der Bezeichner - erwartet, darunter (Statement) die Beschreibung.

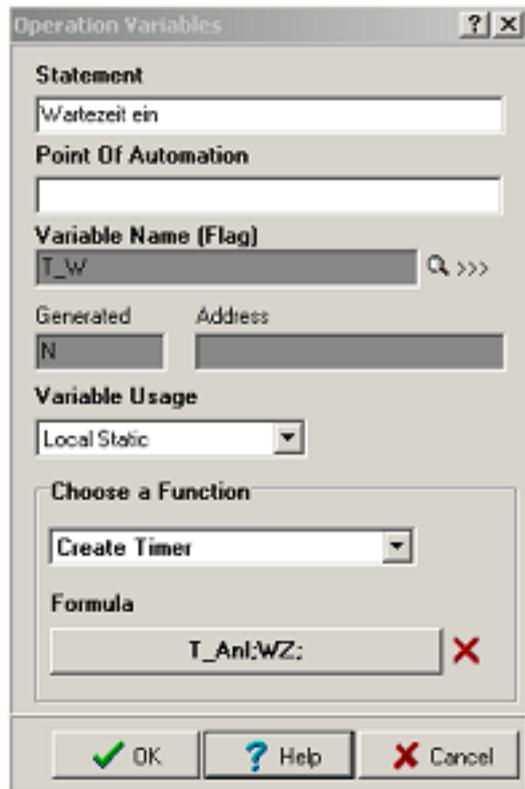
Im unteren Bereich "Timer Value" soll bei "Name" der Bezeichner des Zeitwertes stehen. Im Statement soll wieder ein technologisch sinnvoller Text eingetragen werden. Die beiden Felder "Variable Usage" dienen der Eindeutigkeit der Nutzung, oben "Input" wegen der Step-7-Spezialität, unten "Constant", weil eine feste Zeit vorgegeben werden soll. Die entsprechenden Hilfetexte geben ebenfalls Auskunft.



Klicken Sie nun auf  >>> im oberen Teil. Im Editierfeld der Datenbank tragen Sie bei Name "T\_An1" ein. Danach müssen Sie den Datentyp "Timer" wählen. Anschließend auf  klicken. Betrachten Sie sich das Ergebnis: T\_An1 steht bei "Timer/Name" eingetragen. Vielleicht haben Sie es schon erfolgreich probiert: Das Feld "Statement" ist zum Editieren vorgesehen, also "Anlauf-Timer" eintragen! Tragen Sie im unteren Teil als Statement "Wartezeit" ein und wählen Sie "Constant" bei Variable Usage. Den Zeitwert tragen Sie in das vorgesehene Raster ein, z.B. 250ms. Klicken Sie nun auf  >>> im unteren Teil. Die Zeit erhält noch einen Namen, hier z.B. "WZ". Sie ist vom Datentyp "Time". Am Ende der



Klicken Sie auf "Return" und gelangen zum nächsten Bild:



Hier müssen Sie mit "OK" abschließen, um den konfigurierten Timer endgültig in der Datenbank abzuspeichern. Mit "Cancel" ist jedenfalls bis auf die Variablen alles verloren. Die Variablen müssen Sie getrennt löschen, weil es bei einer Mehrfachnutzung ansonsten zu unliebsamen Überraschungen kommen könnte, wenn eine Variable fehlt, wo sie genutzt wird.

Damit ist das Zeitglied funktionell konfiguriert aber noch nicht fertig projiziert, weil es noch die P-Variable "Wartezeit beendet" einzubinden gilt. Also den Cursor im PRAP auf diese Variable stellen und mit Rechte-Maus-Klick und anschließend mit Klick auf "Data Access" in das Menue "Process Variables" gelangen(auch Doppelklick möglich). Falls beim Entwickeln des PRAP schon "T\_W\_End" als Variablen-Identifikator, kurz: NAME, eingetragen wurde, sehen Sie das nachstehende Fenster:

Process Variables

Statement  
Wartezeit beendet

Point Of Automation

Variable Name (Flag)  
T\_W\_End

Generated Address  
N

Variable Usage  
Local Static

Choose a Function  
Timer Output Q

Formula  
none

OK Help Cancel

Anderenfalls fügen Sie den Namen nach KClick auf und neu ein. Beachten Sie dabei die Hilfeangebote!

Nun bei Formula Klick auf "nothing" oder "none" und bei "Name" den betreffenden Timer eintragen, indem Sie den Filter nur für Member ST\_DR\_Anlauf einschalten - Klick auf "A Filter ON" - und in diesem Fall 'T\_AnI' auswählen, weil dies für den Anlauf-Timer steht.

Process Variables

Statement  
Wartezeit beendet

Point Of Automation

Variable Name (Flag)  
T\_W\_End

Generated Address  
N

Variable Usage  
Local Static

Choose a Function  
Timer Output Q

Formula  
none

Filter Options  
 Member  
 Parameter  
 Data Type  
 Key  
 Filter OFF

Filter Values  
 Member: ST\_DR\_Anlauf  
 Parameter:  
 Data Type: TON  
 Key: 00

| NAME      | DATA TYPE | ADDRESS | INITIAL | POINT OF AUTOMATION | COUNT |
|-----------|-----------|---------|---------|---------------------|-------|
| T_AnI_AUF | TON       | T1      |         |                     |       |
| T_AnI_ZU  | TON       | T2      |         |                     |       |
| T_AnI     | TON       |         |         |                     | 1     |

OK Help Cancel

Mit Klick auf übertragen Sie die Timer-Zuordnung in die P-Variable, und erhalten

Nach einem Klick auf öffnet sich die Variablen-tabelle (Access Table, Zugriff auf die Datenbank) und zeigt Ihnen die ausgewählte Variable:

| NBR | MEMBER       | NAME  | DATA TYPE | ADDRESS | INITIAL VALUE | POINT OF AUTOMATION | PARAMETER | PATH | GEN | USAGE |
|-----|--------------|-------|-----------|---------|---------------|---------------------|-----------|------|-----|-------|
| 09  | St_Dr-Anlauf |       |           |         |               |                     |           |      |     |       |
| 2   |              | T_Anl | TON       |         |               |                     | forward   | N    | N   | Input |

| NAME  | DATA TYPE | ADDRESS | INITIAL | POINT OF AUTOMATION | COUNTER |
|-------|-----------|---------|---------|---------------------|---------|
| T_Anl | TON       |         |         |                     |         |

Mit Klick auf "OK" schließen Sie das Zeitglied-Projektieren ab.

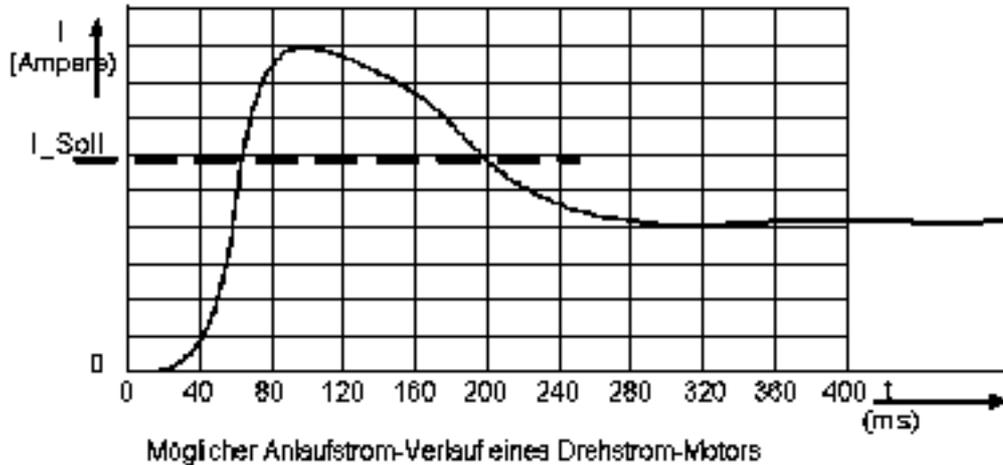
Zurück zu Stern-Dreieck-Anlaufschaltung (Einführung) [siehe Seite 88]

## 12.2.6 Motorstrom-Verarbeitung (Projektierung der Vergleichsfunktion)

Blicken Sie zurück auf die Ausführungen im Thema PRAP Stern-Dreieck-Steuerung [siehe Seite 94], um die hier gezeigte Problematik zu erkennen und zu akzeptieren.

Mit dem Einschalten des Stern-Schützes steigt der Motorstrom innerhalb von 50 bis 200ms auf sein Maximum und fällt dann mit Beginn der Drehbewegung langsam ab. Es wird vorausgesetzt, daß das Betriebssystem der Ressource, also einer SPS, so schnell arbeitet, daß es eine Zykluszeit von unter 25ms

aufweist. Das bedeutet, das Strommaximum ist erfassbar - die Verzögerung durch die Strommessung eingeschlossen - und damit auch auf der abfallenden Stromkurve über der Zeitachse der dort zu erwartende "Weiterschaltwert" des Stromes. Die konkrete Höhe dieses Wertes ist hier nicht von Interesse.



Der Weiterschaltwert  $I_{Soll}$  ist eine Konstante im Vergleich mit dem Motorstrom-Istwert.

Im zu entwickelnden PRAP gibt es fünf Variable:

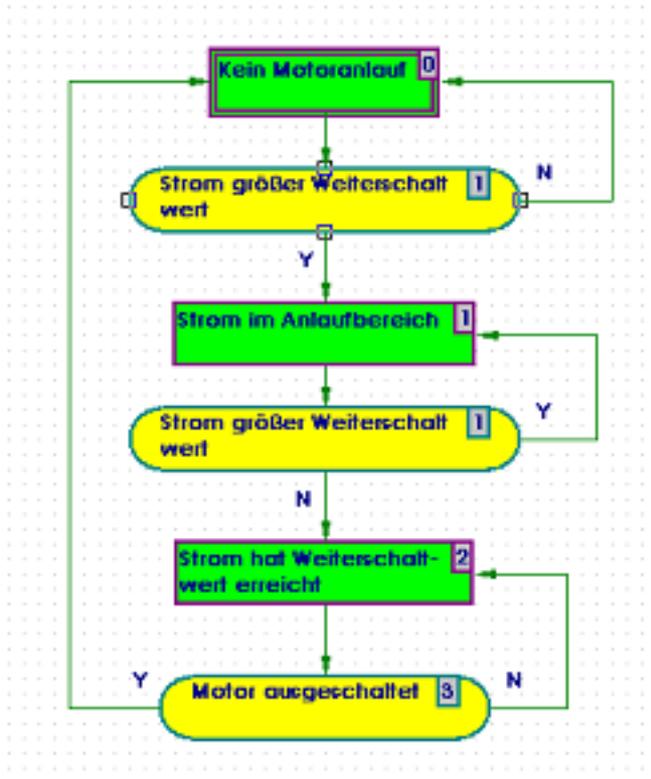
.....**Statement**.....**Identification**..(**Bemerkung**)

=====

- Weiterschaltwert erreicht..... $I_{Schalt}$ .(Kopplung zur Stern-Dreieck-Schaltung)
- Strom größer Weiterschaltwert... $I_{WSW}$ ....(Interne, lokale Boolesche Variable)
- Motorstrom..... $I_{IST}$ ....(Vom Meßgeber oder AD-Wandler)
- Weiterschaltwert..... $I_{SOLL}$ ...(Konstante gemäß Code AD-Wandler)
- Motor ausgeschaltet..... $Mot_{aus}$ ..(Kopplung von Stern-Dreieck-Schaltung)

Typen-PRAP Motorstrom-Auswertung

Die zweite Situation [siehe Seite 183] ist



struktur stabil???

die dritte Situation ist es nicht. Da "Motor ausgeschaltet" als Koppelvariable nicht "falsch" auftreten kann, ist kein Durchlauf [siehe Seite 150] zu erwarten.

Die dritte Operation im PRAP enthält die Koppelvariable gleich Operations-Variable "Wefterschaltwert erreicht" mit "≠"), die "mitteilt", der Wefterschaltwert sei erreicht (ohne diese Markierung: ... NICHT erreicht).



### Tipp

Tragen Sie alle diese Variablen in die Access Table ein. Dadurch haben Sie es leichter, die Vergleicherfunktion zu formulieren.

Zu Ihrer eigenen Sicherheit und zur Information Dritter hat sich sehr bewährt, die **PRAP-Funktion** zu **beschreiben**. Stellen Sie den Cursor auf ein freies Feld außerhalb der PRAP-Symbole und klicken auf die rechte Maustaste; danach Klick auf "Descriptions". Nun können Sie Ihren Beschreibungstext eintragen. Als Beispiel sehen Sie den Text des PRAP-Autors:



Weiter mit: Projektierung der Vergleichsfunktion. [siehe Seite 215]

## 12.3 Lampenschaltung

### 12.3.1 Lampenschaltung

Sie können von hier aus zu allen Teilen dieses Themas springen:

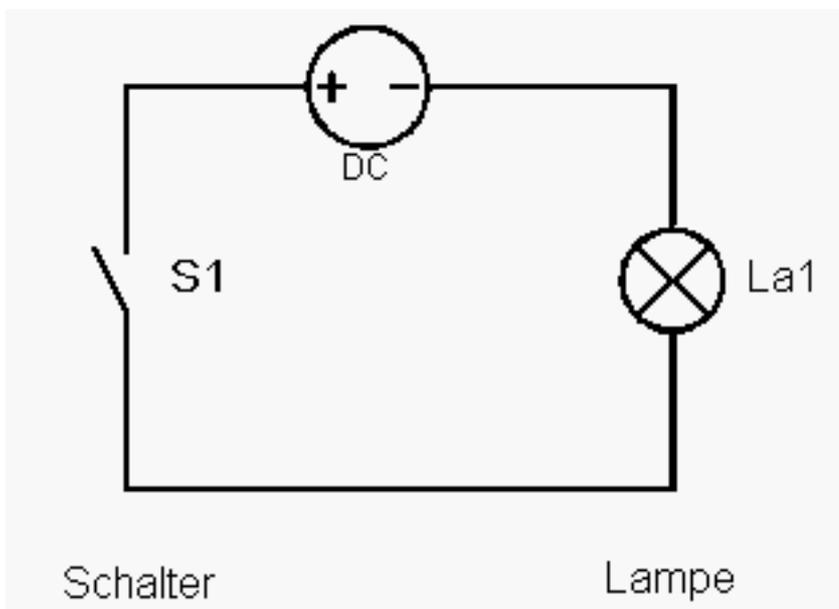
Editieren von Operationen (Teil 1) [siehe Seite 106]

Editieren von Prozeß-Variablen (Teil 2) [siehe Seite 110]

Eintragen der Kanten (Teil 3) [siehe Seite 113]

PRAP Lampensteuerung, Prüfung (Teil 4) [siehe Seite 114]

PRAP Lampensteuerung - Strukturfragen (Teil 5) [siehe Seite 115]



Das vorliegende Schaltbild zeigt ein gewöhnliches Lampensymbol und einen allgemeinen Schalter. Es ist sicherlich das denkbar einfachste Beispiel. In dieser Form ist es ein Typ für das Ein- und Ausschalten einer Lampe. Diesen Typ als PRAP [siehe Seite 126] zu entwerfen wird die erste Aufgabe sein. Die folgenden Aufgaben sind das Instanzieren dieses Typs [siehe Seite 126], z.B. die Beleuchtung bei einem

geschlossenem und die bei geöffnetem Hallen-Tor zu definieren. Bei geschlossenem Hallen-Tor soll die Lampe außen, bei geöffnetem innen, also hinter dem Tor angebracht sein. Diese beiden Teilprozesse "Außenbeleuchtung" und "Innenbeleuchtung" sind hier einfach erfunden, weil keine Prozeß-Zerlegung [siehe Seite 18] eines konkreten Prozesses stattgefunden hat.

Die folgende Beschreibung hält sich an die Texte bei Entwicklung eines PRAP [siehe Seite 66] und Grundsätze zum Gestalten eines PRAP. [siehe Seite 63]  
Formulierung der Prozeß-Zustandsvariablen:

- Für den Typ als PRAP [siehe Seite 126] gilt, dem Schaltzeichen S1 eine Benennung zu geben wie z.B. "Einschaltbedingung erfüllt"
- Die erste "Anwendung" dieses Typs soll sein "Hallen-Tor geschlossen" im TP "Außenbeleuchtung".
- Es soll eine weitere Einschaltmöglichkeit geben "Hallen-Tor geöffnet" im TP "Innenbeleuchtung".

S1 ist nur ein Schaltzeichen-Symbol, niemals geeignet zur Formulierung einer Prozeßvariablen. Beide TP sind Instanzen vom Typ-PRAP "Lampe".

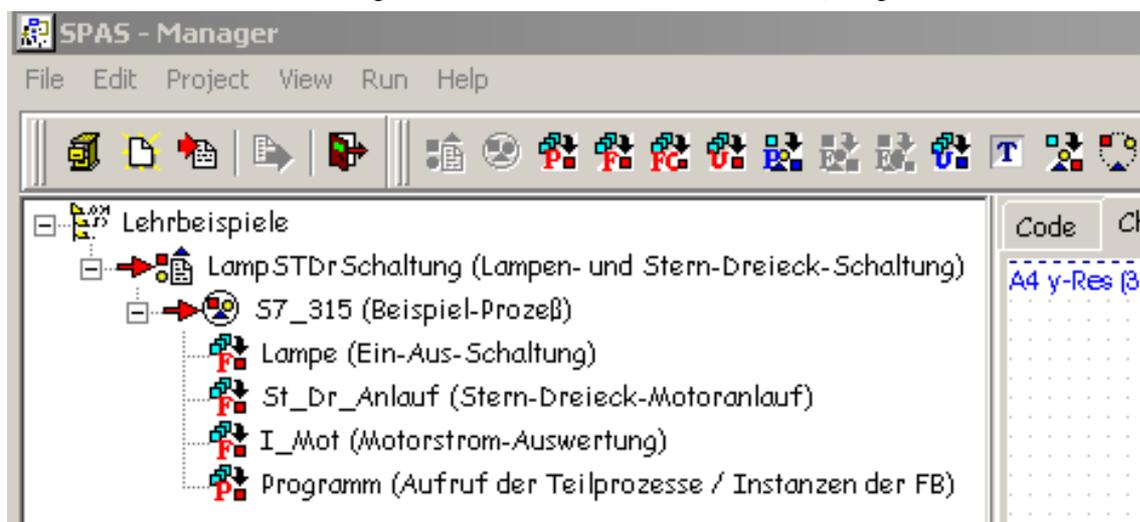
Formulierung der Operationsvariablen:

- Das Symbol La1 soll die Bezeichnung "Beleuchtung einschalten" erhalten.
- Im TP "Außenbeleuchtung" soll "Hallenzufahrt beleuchten" formuliert werden.
- Im TP "Innenbeleuchtung" soll die Variable "Halle beleuchten" heißen.

Damit sind die Variablen zu diesen Teilprozessen benannt, die direkten Adressen könnten hier noch dazugeschrieben werden. Sie müssen spätestens beim Instanzieren bekannt sein. Der Entwurf des Typen-PRAP kann beginnen.

Falls Sie mit dem Projekt-Editor [siehe Seite 51] arbeiten und noch keinen PRAP-Knoten als Member in Ihr Projekt eingefügt haben, dann erledigen Sie das bitte jetzt, weil es sonst keinen Zugang zum PRAP-Editor gibt. Gehen Sie gemäß Projekt-Editor vor.

Dort ist das Beispiel soweit beschrieben, daß sowohl der PRAP für die Lampensteuerung als auch der für die Stern-Dreieck-Steuerung eingefügt sind. Im nachfolgenden Bild wurde der Lampentyp markiert, mit der rechten Maustaste das Menue geöffnet und "Edit Flow Chart (PRAP-FB)" angeklickt:



Weiter mit Aufbau des PRAP zur Lampensteuerung (Teil 1) [siehe Seite 106]

## 12.3.2 PRAP zur Lampensteuerung (Teil 1)

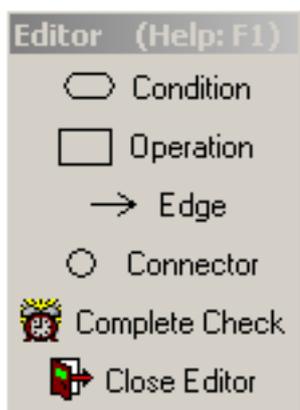
### Editieren von Operationen

Die Registerkarte "CHARTS" muß aktiviert sein (siehe hier [siehe Seite 105]), dann führen Sie den

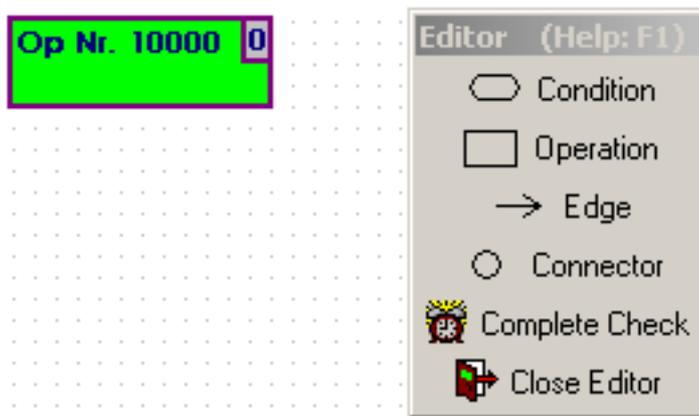
Cursor auf dieses (noch leere) Feld und drücken auf die rechte Maustaste. Sie erkennen ein Menü,



in dem eine Zeile "Graphic Editor" lautet. Ein Klick mit der linken Maustaste gibt Ihnen das Menue frei, mit dem Sie einen PRAP aufbauen können:



Klicken Sie nun die Zeile "Operation" und führen anschließend den Cursor an die Stelle, wohin die erste Operation - sie trägt die Inschrift Op Nr 10000 - gelangen soll. So erhalten Sie ein Bild:

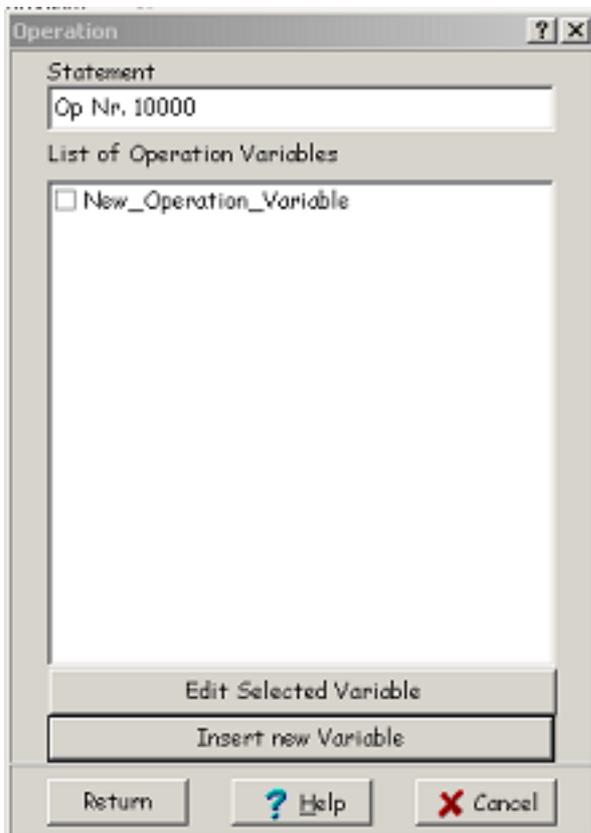


Führen Sie den Cursor auf das Operationssymbol und betätigen Sie die rechte Maustaste, dann klicken



auf die Zeile "Data Access" und erhalten den Zugriff auf alle Daten

der betreffenden Operation:



Dies ist das sogenannte Operations-Variablen-Fenster, in das die Operationsvariablen eingetragen werden. Zuvor ersetzen Sie den Text in "Statement" am besten durch "Keine Beleuchtung", weil das den Anfangs-Stand der Schaltung zum Ausdruck bringt. Das trifft auch den Satz 1 der Grundsätze zum Gestalten [siehe Seite 63]!

Klicken Sie nun auf "Insert New Variable", um eine neue Operationsvariable (es ist die erste!) in die Operation einzutragen. Es erscheint der Text "New\_Operation\_Variable", auf dem Sie mit Doppelklick ein Fenster der bearbeiteten Operation öffnen. (Entfernen einer Operationsvariablen<sup>26</sup>)

<sup>26</sup> Markieren Sie die betreffende Operationsvariable und mit rechtem Mausklick wird Ihnen das Entfernen angeboten.

Operation Variables

Statement  
Operation Variable

Point Of Automation

Variable Name (Flag)

Generated Address

Variable Usage  
Output

Choose a Function  
not specified

Formula  
none

OK Help Cancel

"Statement" bedeutet den Variablennamen, hier "Lampe

einschalten", weil diese Aussage dem entspricht, was eigentlich im "aktiven Fall" geschehen soll. Nach diesem Eintrag als "Comment" klicken Sie bitte auf neben dem Text "Variable Name (Flag)1)". Damit erhalten Sie Zugang zur Variablen- Datenbank:

Operation Variables

Statement  
Lampe einschalten

Point Of Automation

Variable Name (Flag)  
Lampe\_ein

Generated Address  
N

Variable Usage  
Output

Choose a Function  
not specified

Formula  
none

OK Help Cancel

| NBR | MEMBER | NAME      | DATATYPE | ADDRESS | INITIAL VALUE | PARAMETER | PATH | GEN | USAGE  |
|-----|--------|-----------|----------|---------|---------------|-----------|------|-----|--------|
| 1   | Lampe  | Lampe_ein | BOOL     |         |               | formal    | N    | N   | Output |

Access Table : unfiltered

NUR HIER UND NUR IM OBEREN TEIL (also nicht in der Tabelle - unten - selbst) können Sie die weiteren Variableneinträge tätigen. In das Feld "Name" tragen Sie eine Kurzform ein. Hier steht kurz "La\_ein". Wichtig ist nun, die Datenbank-Symbole richtig zu bedienen (automatische Hinweise beachten):



Von links nach rechts:

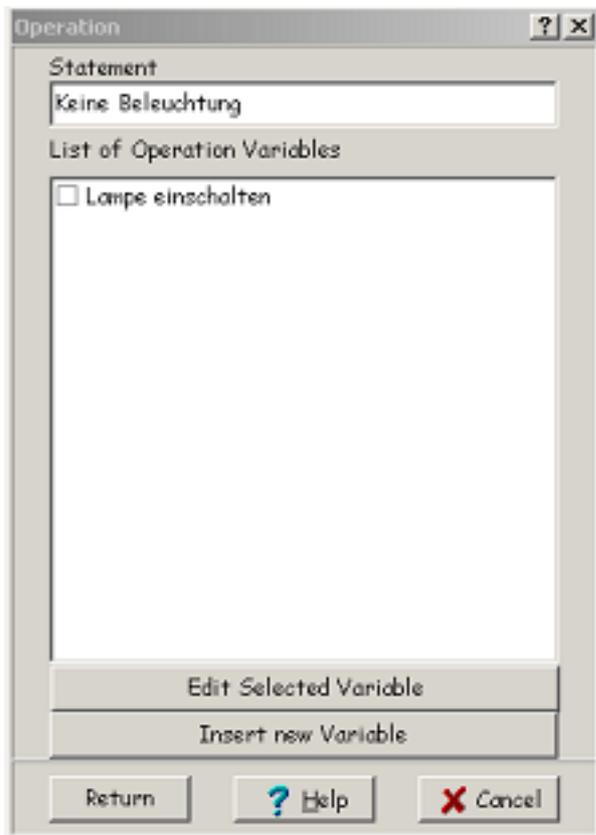
- Neuen Datensatz einer Variablen hinzu
- Datensatz zur markierten Variablen löschen

-  Datenbank-Inhalt editieren
-  alle Änderungen in Datenbank sichern
-  Editiervorgang ungeschehen beenden
-  Datenbankinhalt erneut sichern
-  Daten übernehmen, zugleich in Datenbank sichern
-  alle Variablen-Records anzeigen

 Hilfetext für die Variablen-Behandlung in dem großen, rechten Feld. Das Fragezeichen  des oberen Feldes bringt weiterhin Hilfetexte zu einzelnen Feldern.

Danach, und das ist bei mehreren eingetragenen Variablen wichtig, die richtige Variable in der Datenbankzeile markieren und  anklicken. Dadurch erfolgt der Eintrag in das Feld "Variable Name (Flag)". Klick auf  läßt die Datenbank verschwinden, Klick auf "RETURN" führt zurück zum Operations-Fenster mit dem Statement (der Erklärung) der Variablen:

Nach weiterem RETURN erhält man:



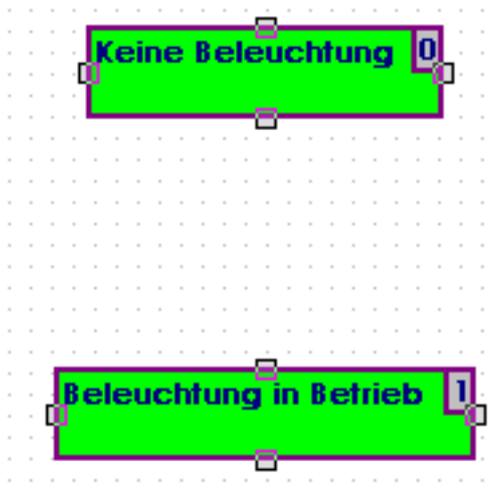
Das Markierungsquadrat bleibt LEER, weil

"Lampe ausschalten" gilt! Weiter mit Aufbau des PRAP zur Lampensteuerung (Teil 2) [siehe Seite 110]  
Zurück zum Anfang [siehe Seite 105]

### 12.3.3 PRAP zur Lampensteuerung (Teil 2)

Nachdem die erste Operation erstellt wurde, ist zu überlegen, welche Folgeoperation(en) es gibt, für dieses Beispiel natürlich nur eine: Beleuchtung in Betrieb.

Gehen Sie ebenso vor wie beim Herstellen der ersten Operation, nur eine neue Operationsvariable benötigen Sie nicht. Zunächst tragen Sie bei "Statement" den Text "Beleuchtung in Betrieb" ein, weil dies in der zweiten Operation geschehen soll. Danach markieren Sie durch Klick das Quadrat neben dem Variablennamen "Lampe einschalten", womit klar ist, daß hier die andere Wertung gilt gegenüber der Operation 0. Mit Klick auf RETURN verschwindet das Operationfenster und die beiden Operationen



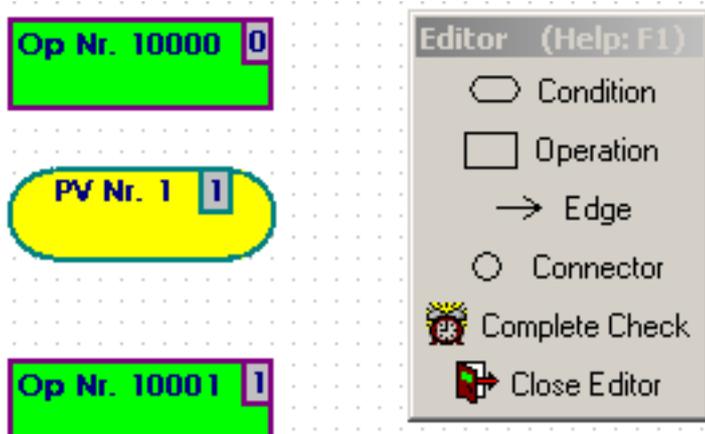
des PRAP sind vorhanden.

#### Editieren von Prozeßzustands-Variablen / Conditions

Nun ist zu überlegen, welche Ursache den Übergang von der Operation 0 zur Operation 1 (die Nummern finden Sie am unteren Bildrand der Registerkarte Charts in Klammern hinter dem Comment-Text) bestimmt. Hier im Beispiel ist das sehr einfach, weil es nur eine Prozeßvariable gibt - siehe oben.

Das Einfügen einer Prozeßvariablen zwischen die beiden Operationen geschieht wie folgt:

Mit Klick auf die rechte Maustaste öffnet sich das bekannte Menue, in dem Sie "Graphic Editor" anklicken. Jetzt klicken Sie auf Condition (=Bedingung, hier P-Variable), bewegen den Cursor zwischen die Operationen und klicken erneut. Es ergibt sich das Bild:



Sie können die Position des Symbols beliebig verschieben:

Blieben Sie mit dem Cursor auf dem Symbol und betätigen die linke Maustaste und halten Sie fest.

Zur Spezifikation der Variablen:

Blieben Sie mit dem Cursor auf dem Symbol und betätigen die rechte Maustaste; Sie werden nach der Feststellung (englisch), daß es noch keine P-Variable gibt, gebeten, eine neue zu erzeugen (Yes-Button), oder eine vorhandene wieder zu verwenden (RETRY-Button [siehe Seite 218]).

Für das Beispiel ist eine neue zu erzeugen: Ein Klick auf "Yes" führt zu einem bekannten Fenster, nur daß es eben für "Process Variables" geöffnet wurde. "New\_PVar\_Statement" ersetzen Sie durch "Schaltbedingung erfüllt" und klicken danach wieder auf , womit sich der Datenbank-Zugang öffnet.

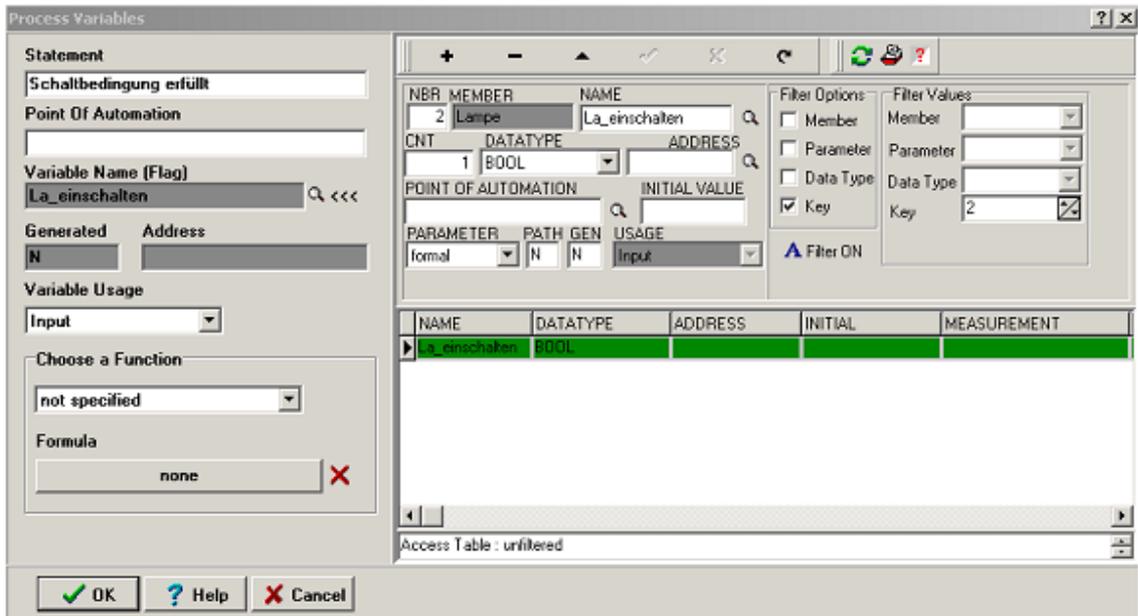


#### **Anmerkung**

Wundern Sie sich nicht über den Eintrag der O-Variablen, hier werden alle Variablen gehalten!

Klicken Sie auf das .

Sie erhalten eine neue Zeile mit dem Eintrag "pVar", den Sie z.B. durch "La\_einschalten" ersetzen. Damit erhalten Sie:



Durch Klick auf  wird der Variablen-Kurzbezeichner "La\_einschalten" in das Feld "Variable Name (Flag)" übernommen, ABER achten Sie darauf, daß die zu übertragende Variable auch markiert ist:

|   | NAME           | DATATYPE | ADDRESS |
|---|----------------|----------|---------|
| ▶ | La_einschalten | BOOL     |         |
|   | La_ein         | BOOL     |         |

Klicken Sie nun entweder auf  und/oder auf RETURN, um zum jetzigen PRAP-Bild



zu gelangen.

Zum Schluß dieses Abschnittes noch eine Bemerkung:



### Anmerkung

Jedes PRAP-Element enthält seine Nummer automatisch. Es ist bei mehreren P-Variablen auch völlig unerheblich, ob aus irgend einem Grund die fortlaufende Zählreihe unterbrochen wird. Die Prüfergebnisse "Check Graphic" enthalten diese Nummern.

Wollen Sie eine fortlaufende Zählung, dann spezifizieren Sie jede neue Condition sofort nach dem Platzieren.

Zurück zu PRAP zur Lampensteuerung (Teil 1). [siehe Seite 106]

Weiter mit PRAP zur Lampensteuerung (Teil 3). [siehe Seite 113]  
Zurück zum Anfang [siehe Seite 105]

## 12.3.4 PRAP zur Lampensteuerung (Teil 3)

### Eintragen der Kanten / Edges

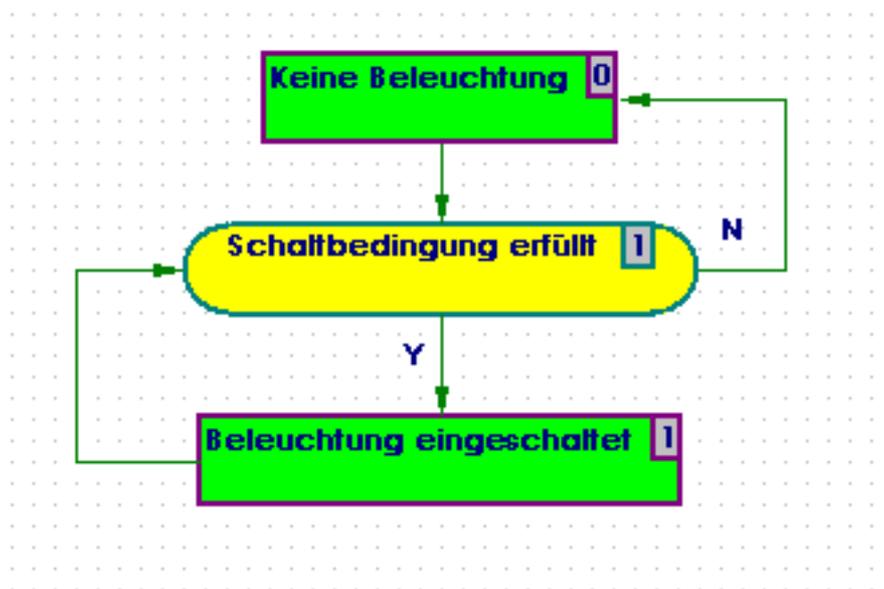
Nun müssen die drei PRAP-Elemente durch Edges (=Kanten, gerichtet) verbunden werden.

### Der Zieh-Vorgang:

Zu diesem Zweck klicken Sie im Graph-"Editor" auf "Edge", führen den neuen Cursor in die obere Operation, klicken die linke Maustaste und führen den Cursor in die P-Variable, und klicken Sie wiederholt. Nun sehen Sie eine Verbindung zwischen der Operation und dem Condition-Symbol (ein sog. Langrund) mit Pfeil auf die Condition gerichtet. Man sagt auch, die wegführende Kante beginnt an einem Symbol, die hinführende Kante endet mit einem Pfeil am Symbol.

Es gelten folgende Regeln: [Click here](#) [siehe Seite 217]

Sie erkennen im Bild des nun fertigen PRAP das Ergebnis des Ziehens aller Verbindungen.



Die Kanten sollen jetzt erläutert werden:

Die mit "Y" bewertete Kante (Yes=Ja) führt ganz selbstverständlich zur Folgeoperation.

Die mit "N" bewertete Kante führt zurück zur Operation "Keine Beleuchtung", weil dies der Fall ist, solange "Schaltbedingung erfüllt" mit No=nein bewertet wird. Die dadurch entstehende "Schleife" garantiert, ganz praktisch gedeutet, daß die Beleuchtung nicht eingeschaltet wird: Diese Situation [siehe Seite 183] ist stabil, solange die Aussage "N" bleibt.

Die Operation "Beleuchtung in Betrieb" bleibt natürlich solange stabil, wie "Schaltbedingung erfüllt" mit "Y" gilt. Folglich muß die wegführende Kante dieser Operation an der P-Variablen enden, dort ist sie hinführende Kante. Damit ist die zweite Situation des PRAP definiert.

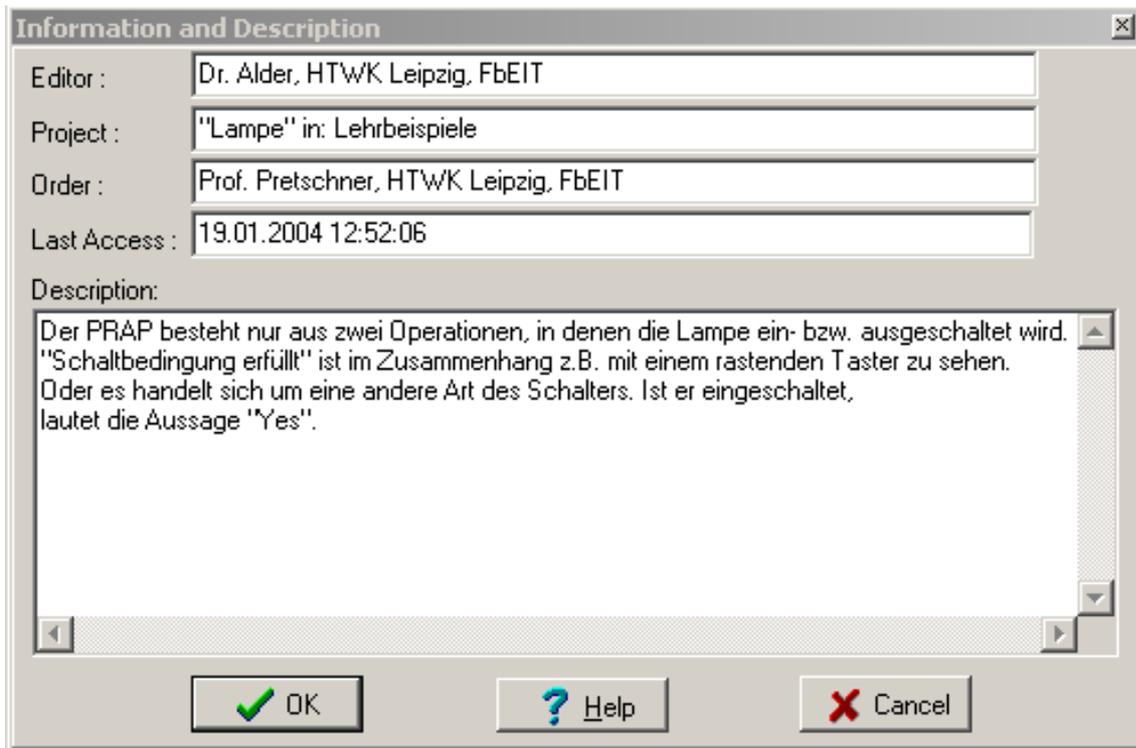
Lesen Sie auch Punkt 6 aus Grundsätze zum Gestalten eines PRAP. [siehe Seite 63]



### Tipp

Übrigens "Save Graphic" sollte immer mal wieder angeklickt werden, damit Ihnen der PRAP erhalten bleibt. Dazu gehören auch alle zum PRAP gehörigen Variablen!!

Zu Ihrer eigenen Sicherheit und zur Information Dritter hat sich sehr bewährt, die **PRAP-Funktion** zu **beschreiben**. Stellen Sie den Cursor auf ein freies Feld außerhalb der PRAP-Symbole und klicken auf die rechte Maustaste; danach Klick auf "Descriptions". Nun können Sie Ihren Beschreibungstext eintragen. Als Beispiel sehen Sie den Text des PRAP-Autors:



Zurück zu

PRAP zur Lampensteuerung (Teil 2) [siehe Seite 110]

Weiter mit PRAP zur Lampensteuerung (Teil 4) [siehe Seite 114]

Zurück zum Anfang [siehe Seite 105]

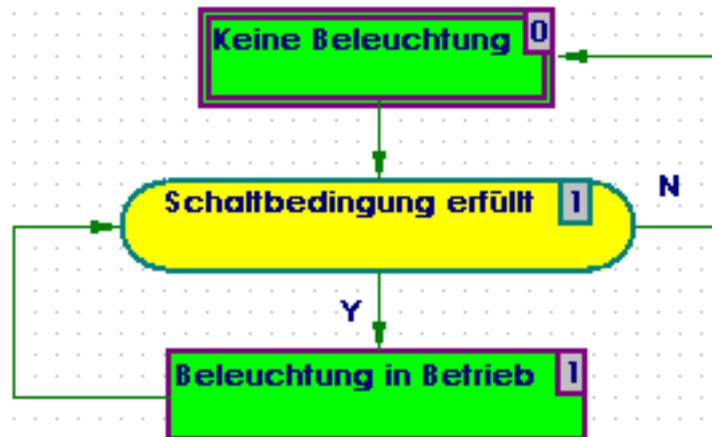
## 12.3.5 PRAP zur Lampensteuerung (Teil 4)

### PRAP-Prüfungen

Im Kapitel über Prüfungen des PRAP [siehe Seite 64] sind alle notwendigen Erläuterungen enthalten. Hier soll gezeigt werden, wie am Beispiel die Prüfungspraxis aussieht.

Der PRAP im letzten Abschnitt (Teil 3) liegt fertig erstellt vor, den Cursor an irgend eine Stelle des freien Blatt-Teils positionieren und rechte Maustaste drücken führt zu einem Menü, in dem es die Zeile "Check Graphic" gibt. Gehen Sie nun in den unteren Teil des SPAS-Manager, dem sog Aktionsfeld, auch Meldungsfeld. Dort könnte der letzte Eintrag sinngemäß sagen, daß der rechte Mausklick zu nichts geführt hat. Drücken Sie in diesem Feld die rechte Maustaste, dann haben Sie die Möglichkeit, die Action List zu leeren. Falls Sie die Zeilen verfolgt und die Empfehlungen durchgeführt haben, ist das Menü verschwunden.

Öffnen Sie das Menü eventuell wiederholt und klicken auf "Check Graphic", dann erhalten Sie unter anderen eine Mitteilung, die besagt, daß **keine Initialoperation** vorhanden ist. Der PRAP zur Lampensteuerung hat wenig Auswahl, es gibt nur zwei Operationen, und man könnte z.B. "Keine Beleuchtung" wählen. Führen Sie den Cursor in dieses Operations-Symbol, drücken die rechte Maustaste und klicken auf "Set Launch Operation". Der nun ergänzte doppelte Rahmen kennzeichnet die Initialoperation.



Eine andere Mitteilung sagt, daß die Variablen geprüft wurden. Zum Test nehmen Sie bitte eine der Kanten heraus und lassen wieder "Check Graphic" arbeiten, dann erkennen Sie sehr schnell, wo der Fehler liegt und vor allem wie er angezeigt wird. Übrigens können Sie durch Klick auf "Reload Graph" immer wieder den alten Zustand der Graphik herstellen.

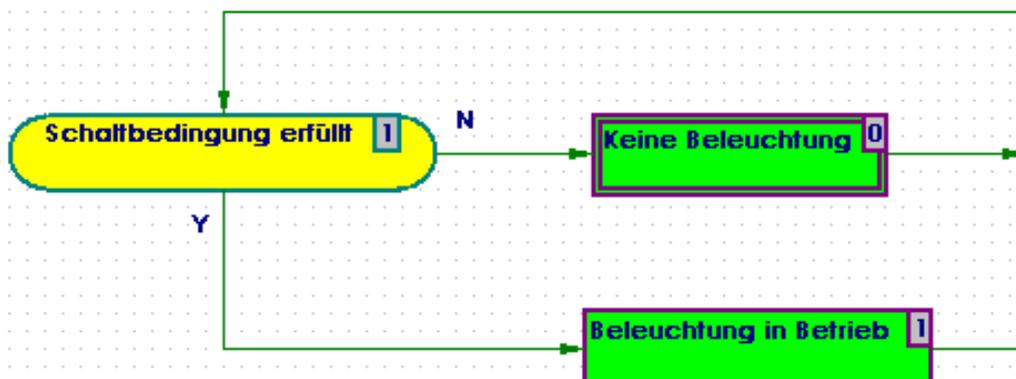
Zurück zu PRAP Lampensteuerung (Teil 3) [siehe Seite 113]

Weiter mit PRAP Lampensteuerung (Teil 5) [siehe Seite 115]

Zurück zum Anfang [siehe Seite 105]

## 12.3.6 PRAP zur Lampensteuerung (Teil 5)

### Eigenschaften des PRAP Lampensteuerung



Wenn Sie sich diesen Prap betrachten, werden Sie sicher überrascht sein, daß er tatsächlich funktionell dasselbe beschreibt wie im vorangegangenen Abschnitt. Im Kapitel Prüfungen und Optimieren eines PRAP [siehe Seite 64] ist bereits im Punkt 5. auf die hier gezeigte Möglichkeit des Umzeichnens hingewiesen worden. Nur liegt hier ein sehr kleines Beispiel vor, so daß der Effekt auch gering ist.

Aber es geht um das Prinzip: Es handelt sich hier um eine sogenannte reine kombinatorische Struktur [siehe Seite 168], weil die beiden Operationen keine gemeinsame Folgeoperation haben, die nicht zur kombinatorischen Struktur gehört.

Der PRAP ist ein Typen-Prozeßablaufplan [siehe Seite 193], weil er keine direkt adressierten Variablen [siehe Seite 123] enthält.

Zurück zu PRAP Lampensteuerung (Teil 4) [siehe Seite 114]

[Zurück zum Anfang](#) [siehe Seite 105]

## **12.4 Anwendungsbeispiele**

### **12.4.1 Beispielprojekt**

An einem kleinen Ein-Ausschalt-Vorgang und einem Stern-Dreieck-Schaltproblem werden die Handhabung des Projekt-Editors und des PRAP-Editors zur Vorbereitung der Übersetzung in ein Steuerprogramm für S7-300 (Fa. Siemens) behandelt.

Das Beispielprojekt finden Sie hier. [\[siehe Seite 73\]](#)

Die PRAP-Entwicklungen erreichen Sie hier. [\[siehe Seite 66\]](#)

Ein Timer aus dem Beispiel-Projekt: [Klicken Sie hier.](#) [\[siehe Seite 116\]](#)

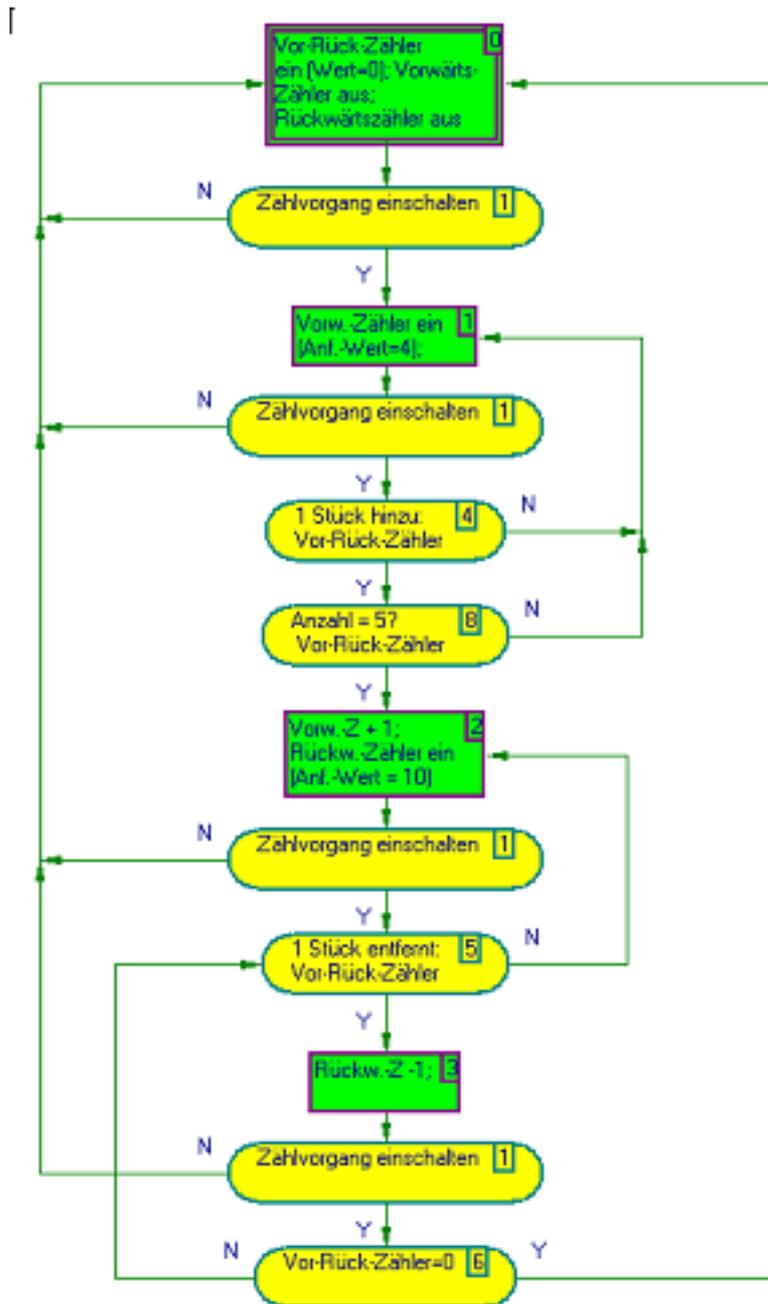
Eine Vergleich-Funktion aus dem Beispielprojekt: [Stromvergleich](#) [\[siehe Seite 215\]](#)

### **12.4.2 Beispiel Timer**

Der folgende Timer ist mit einer konstanten Dauer entworfen: [Beispiel aus Stern-Dreieck-Schaltung](#) [\[siehe Seite 98\]](#)

### **12.4.3 Zähler-Beispiel**

Im PRAP sind drei Zähler



[siehe Seite 29] enthalten, ein Vorwärtszähler, ein Rückwärts-Zähler und ein Vorwärts-Rückwärts-Zähler. Der Vorwärts-Rückwärts-Zähler(V-R-Z)

wird bereits mit dem Anfangswert Null in der Initialsituation eingeschaltet und nie ausgeschaltet. Damit wird erreicht, daß das momentane Zählergebnis während der Gesamt-Funktionszeit zur Verfügung steht. Die PRAP-Funktion wird durch den V-R-Z gesteuert. Der Übergang von der Operation O1 zur Operation O2 erfolgt, sobald das Zählergebnis 5 ist. Der Übergang von O3 zu O0 erfolgt bei der Aussage: Vor-Rück-Zähler = 0 (Prozeßvariable p6).

Die Prozeßvariable 1 (p1), "Zählvorgang einschalten", sei irgendein Schalter, der zu jeder Zeit auch ausgeschaltet werden kann, dann gilt  $p1=N$ .

Das Vorwärts-Zählen des V-R-Z geschieht durch die Prozeß-Variable p4, das Rückwärts-Zählen durch die Prozeß-Variable p5. Stellen Sie sich bitte je einen Sensor vor, um die beiden Ereignisse des Hinzufügens und Entfernens zu "melden".

Der Vorwärtszähler

wird erst nach dem Einschalten des Zählvorganges in Betrieb genommen. Er erhält einen Anfangswert 4, natürlich etwas merkwürdig, weil jede Erhöhung durch Zählen nur höhere Werte liefert. Aber es ist ja nur ein Beispiel. Das Zählen erfolgt in der O2 durch eine Operationsvariable. Nach einem "Sollwert" wird nicht gefragt.

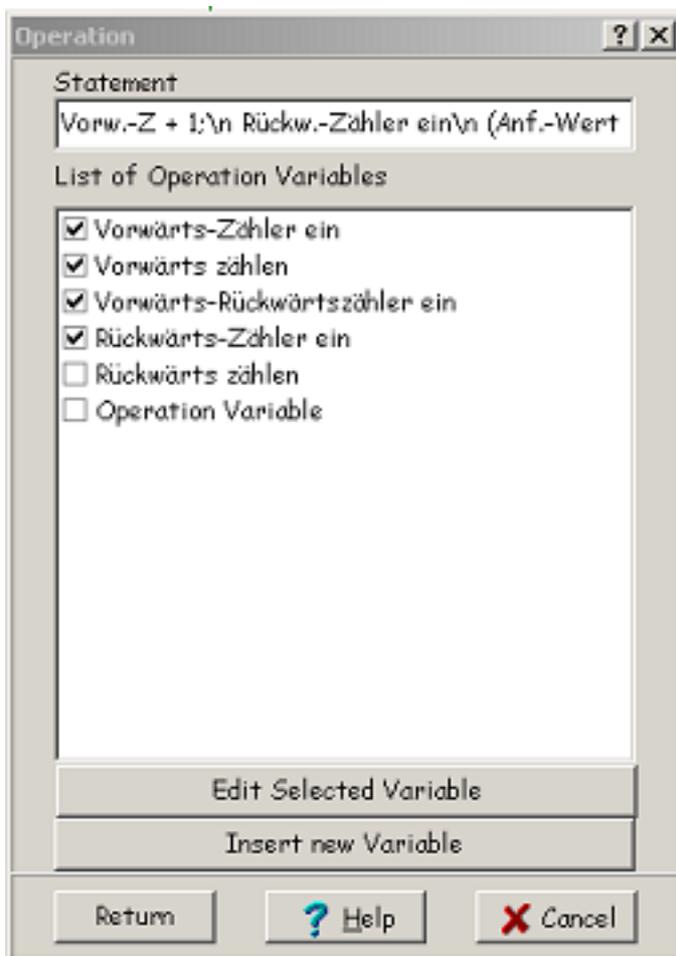
### Der Rückwärtszähler

wird in der Operation 2 eingeschaltet mit einem Anfangswert = 10. Das Rückwärtszählen erfolgt in der O3 durch eine Operationsvariable. Auch hier wird nicht nach einem Sollwert gefragt, z.B. wann der Wert 10 heruntergezählt ist, also der Zählerausgang Null ist.

### Die PRAP-Funktion:

Die Initial-Situation [siehe Seite 183] (Sit0) kann nur mit dem Einschalten (p1) verlassen werden. Die Folge-Situation mit O1 ist so lange stabil, wie der V-R-Z noch nicht den Wert 5 (p8) erreicht hat (Vergleichsfunktion: Zählerstand mit Konstante 5), oder der Zählvorgang wird abgeschaltet (p1). Immer dann, wenn ein Stück hinzu kommt (p4 = Y), wird der Zählerwert um 1 erhöht. Dabei wechselt die Stabilitätsbedingung für O1 von  $p1=Y$  UND  $p4=N$  nach  $p1=Y$  UND  $p4=Y$  UND  $p8=N$ . Ein neues Stück kann erst erfaßt werden, wenn der Sensor vorher wieder  $p4=N$  meldet.

Der Übergang zur Situation 2 (Sit2) mit der Operation 2 führt zur Stabilität nur, wenn in diesem Augenblick kein Stück entnommen wird. Mit dem Übergang zählt der V-Z und der R-Z wird eingeschaltet.



Im Bild sehen Sie die markierte O2 und die Belegung der Operationsvariablen. Im Feld "Statement" können Sie erkennen, wie der Text [siehe Seite 143] im Rechteck der Operation entsteht, beachten Sie "\n" als Symbol für Zeilenwechsel. Darunter, in der "List of Operation Variables", sehen Sie alle Operationsvariable des PRAP und deren Markierung.

Erfolgt nun eine Entfernung eines Stückes, findet der Übergang zur Situation 3 statt, wobei diese nur stabil ist, solange der Sensor  $p5=Y$  meldet.

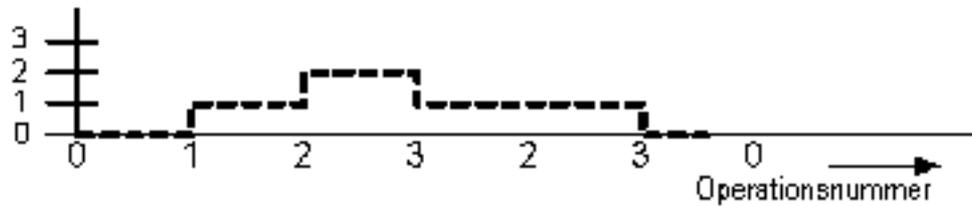
Sobald der Sensor  $p5=N$  meldet, findet der Übergang von Sit3 nach Sit2 statt. Dieses "Hin-Und-Her" des Rückwärtszählens führt bei leerem Zähler ( $p6=Y$ ) zum Übergang in die Sit0, die nicht stabil ist, weil der Zählvorgang ja noch eingeschaltet ist, also  $p1=Y$  anliegt. Entweder jetzt wird ausgeschaltet, oder ein neues Zählversuch kann beginnen.



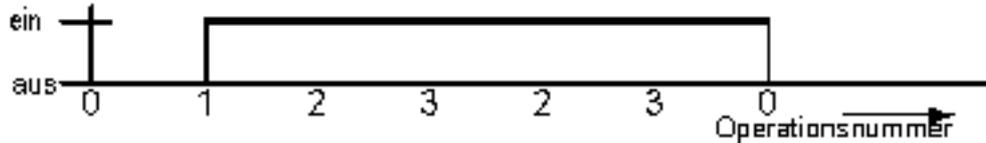
### Anmerkung

Wenn Sie diesen PRAP selbst ausprobieren, dann meldet "Check Graph" per Warnung die mögliche instabile Schleife O2 - O3 - O2.

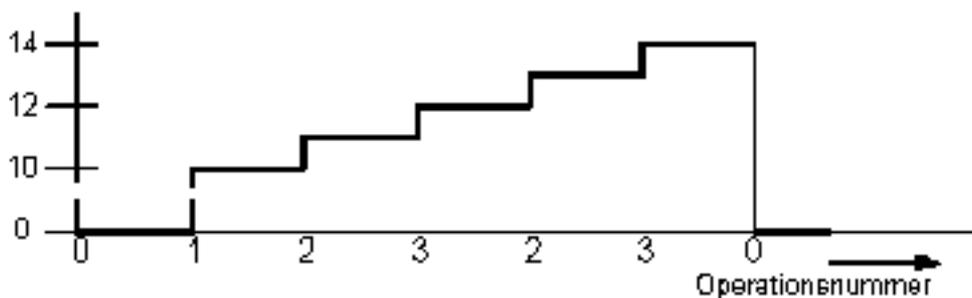
Wert V-R-Zähler



V-Zähler



Wert V-Zähler



Die dargestellten Zählerdiagramme sollen zeigen, in welchen Operationen welche Ereignisse eine Zählwert-Änderung erzielen. Der V-R-Z ist nur von außen durch die Stück-Zugabe oder-Entnahme beeinflusst, während die beiden anderen nur durch den Operationen-Wechsel bzw. Situationen-Wechsel ihren Wert verändern.

Die Projektierung eines Zählers bzw. dieses Beispiels finden Sie im Abschnitt Entwurfsprozeß - Projektierungs-Ablauf. [siehe Seite 29]

## 12.4.4 Beispiel Vergleichler

Die folgende Vergleichlerfunktion ist mit einem konstanten Sollwert entworfen: Beispiel aus Motor-Strom-Auswertung [siehe Seite 215]



## **II Lexikon**



# Kapitel 13

## Begriffe in SPaS

### 13.1 Begriffs-Erklärungen

Ein Werkzeug, wie SPAS es darstellt, benötigt eine zum Teil neue Begriffswelt, weil die Struktur eines Steuerprogramms sowohl den Erfolg der Prozeß-Steuerung bestimmen kann als auch der Projektant insbesondere bei großen Prozessen eine Übersicht benötigt, die er sich durch verschiedene Strukturen schaffen und bewahren kann. Das gilt besonders bei Änderungen in einzelnen Teilprozessen, die normalerweise im PRAP mit dem PRAP-Editor stattfinden. Begriffe aus der Norm IEC 1131-3 werden hier nicht näher erläutert, sondern nur deren Texte aus der Norm zitiert.

Hier finden Sie Begriffe alphabetisch geordnet und kurz erklärt: Wörterbuch [siehe Seite 129]

Herausgehobene Themen aus dem Wörterbuch:

Adresse - allgemein [siehe Seite 123]

Adresse - direkt [siehe Seite 123]

Formalparameter [siehe Seite 125]

Aktualparameter [siehe Seite 124]

Aufruf [siehe Seite 125]

Aufrufbezeichner [siehe Seite 125]

Aufrufgruppe [siehe Seite 125]

Baustein [siehe Seite 125]

Bausteinbezeichner [siehe Seite 126]

Instanz-Baustein [siehe Seite 126]

Prozeß-Ablauf-Plan (PRAP) [siehe Seite 61]

PRAP- und Fremd-Programme(ASCII-Baustein) [siehe Seite 126]

Typ-Baustein / Bausteintyp [siehe Seite 126]

Project (Projekt) [siehe Seite 127]

### 13.2 Adresse - allgemein

Alle von Ihnen eingeführte Variablen, die unter USAGE mit "Local Static" markiert wurden und die daher KEINE ADRESSE erhalten, sind im PRAP- oder PROGRAM-Typ unter "VAR" ... "END\_VAR" deklariert. Der Maschinencode-Compiler erzeugt zu jedem dieser "Symbole" eine für den Benutzer nicht sichtbare Adresse im RAM.

Wollen Sie eine Adresse angeben, dann lesen Sie unter Adresse - direkt [siehe Seite 123] weiter.

### 13.3 Adresse - direkt

In der IEC 1131-3 wird hierzu unter direkte Darstellung (direct representation) ausgeführt: "Mittel zur Darstellung einer Variablen in einem SPS-Programm, von der eine herstelllerspezifische Entsprechung zu einem physikalischen oder logischen Speicherort direkt bestimmt werden kann".



#### Anmerkung

In SPAS ist festgelegt, daß Bit-Adressen mit [Byte-Nr].[Bit-Nr] anzugeben sind, wobei [Bit-Nr] = 0 .. 7 vorgeschrieben ist.

Byte-Adressen sind von 0 an frei wählbar.

Wort-Adressen enthalten immer zwei Byte-Adressen. Also enthält z.B. die Wortadresse 14 die Byte-Adressen 14 und 15. Die nächste Adress-Nummer ist die 16.

Doppelwort-Adressen enthalten folglich zwei Wortadressen.

Unter USAGE sind "Address Input" oder "Address Output" oder "Address Memory" einzutragen, niemals

"Local Static" oder "Local Temporary". SPAS bestimmt aus dieser Angabe das Präfix: Z.B. bei "Address Input" für STEP 7 "E", "EB", "EW" oder "ED" je nach Datentyp. Dagegen müssen Sie Peripherie-Adressen vollständig eintragen (z.B. PEW 352). Konventionelle Steuerungen dürfen generell z.B. mit EW17 oder IW17 für ein Eingangswort ("Address Input") adressiert werden, oder über Pointer wie Adress-Register ("Address Memory").

## 13.4 Aktual-Parameter

Ein Instanzbaustein übergibt aktuelle, d.h. für diesen Baustein gültige Adressen oder Konstante an den für ihn gültigen Typ-Baustein. Deshalb werden diese auch Übergabe-Variable genannt, obwohl es auch feste Werte - Konstante - sein dürfen. Wichtig ist dabei die Reihenfolge der Aktualparameter, weil in der Variablendeklaration des Typ-Bausteins [siehe Seite 126] die Formalparameter "Platzhalter" der funktionellen Verarbeitung darstellen.

Deklariert werden Aktual-Parameter in Members wie RESSOURCE und PROGRAM mit Address Memory, Address Input, Address Output oder Local Static.

## 13.5 Konstante Größen

In SPAS werden weitestgehend die Festlegungen in der IEC 1131-3 (DIN EN 61131-3) zur Anwendung gebracht. Deshalb ist es sinnvoll, folgende Hinweise zu beachten, wenn Sie USAGE = Constant wählen und in INIT VALUE eintragen:

- Jede Größe hat eine Identifikation über ihren Namen (NAME, IDENTIFICATION)
- Jede Konstante ist auch einem Datentyp [siehe Seite 147] zugehörig
- Boolesche Größen  
Sie dürfen true / false (Groß- oder Kleinbuchstaben) oder 0 / 1 verwenden.
- Ganze Zahlen:
  - SINT: -128 ..+127
  - INT: -32768 .. +32767
  - DINT: -2147483648 .. +2147483647

Große Zahlen können Sie mit "\_" aufgliedern, z.B. 343\_562\_110. Aber bitte kein Punkt oder Komma verwenden.

- Gleitkommazahlen:  
REAL: -3.402823\*10 hoch+38 .. 0.0 .. +3.402823\*10 hoch+38  
Verwenden Sie nicht mehr als 7 Ziffern  
Tausender mit "\_" aufteilen oder mit ","  
anstelle 10 hoch.. E+38 bzw. E-38 schreiben, aber die Potenzangabe kann auch unterbleiben. Z.B. ist 234.368 eine Real-Zahl.
- Zeitwerte:  
Gemäß IEC 1131-3 ist T# ..H ..M ..S ..MS gültige Eingabe, aber SPAS gestattet, Zeitwerte verkürzt schreiben zu dürfen. Z.B. werden auch 25,3min angenommen.  
Tageszeiten und Kalenderwerte, also etwa DATE\_AND\_TIME ist späteren Versionen vorbehalten.
- Zählwerte:  
Für alle Zähler gelten Werte vom Datentyp INT. Sie erhalten eine Fehlermeldung, wenn z.B. für Step 7 der Wert 999 überschreitet. Denken Sie daran, daß Sie auch mit Arithmetik alle Zählprobleme lösen können.

## 13.6 Formal-Parameter

Jede Funktion und jeder Funktionsbaustein enthält Variable, die funktionell innerhalb des Bausteins verarbeitet werden. Diese Variablen werden Formalparameter genannt, die in der Variablen-Deklaration des Typen-Bausteins aufgeführt sind. Sie sind "Platzhalter" für Aktualparameter und damit vorläufige Adressen, die zu wirklichen werden, wenn ein Instanzbaustein aktuelle übergibt.

## 13.7 Aufruf

Zum Aufbau eines Steuerungsprojektes wird der Begriff neu beschrieben:

Jeder Aufruf muß eine Ebene nach dem Bezeichner [siehe Seite 125] eines Aufrufes eingefügt werden. Die Reihenfolge der Aufrufe wird im Steuerprogramm eingehalten. Jeder Baustein muß vom Betriebssystem des Prozeßsteuerrechners, also auch von einer SPS, wenigstens einmal aufgerufen werden. Die Form des Systemrufes kann von Hersteller zu Hersteller einer SPS recht unterschiedlich sein. In konventionellen Sprachen, z.B. STEP7-AWL, gibt es der DIN 19239 entsprechend Organisationsbausteine, die als Aufrufbezeichner zu betrachten sind. Ein Aufruf kann aber auch von einem Baustein ausgehen (in dieser SPAS-Version noch nicht!), Unterprogramm-Aufruf wird diese Form des Aufrufes genannt. Gemäß DIN IEC 61131-3 ist Aufruf ein Sprachkonstrukt mit der Zeichenfolge CAL(C oder CN) zur Ausführung einer Funktion oder eines Funktionsbausteines.

## 13.8 Aufruf-Bezeichner

Gemäß der DIN IEC 61131-3 steht jeder Aufruf in einem Programm = PROGRAM, um danach die Programm-Instanz einer Task zuzuordnen. PROGRAM ist folglich im Sinne des Teilprojekt-Aufbaus ein Aufrufbezeichner. Echtzeit-Betriebssysteme verwalten Tasks, jede Task??? ruft ihr zugeordnetes Programm auf (Programm-Instanz).

Betriebssystem-Aufrufe werden auch gemäß DIN 19239 Organisationsbausteine genannt, wie es von SPS-Erzeugnissen der Firma Siemens her bekannt ist. Ein solcher Aufrufbezeichner beginnt mit den Buchstaben OB und bedienen eine große Palette von Tasks.

## 13.9 Aufrufgruppe

Der Begriff der Aufrufgruppe bildet sich aus der Tatsache, Bausteine im Projektbaum in gleicher Ebene untereinander in das PROGRAM einzutragen, um sie in dieser Reihenfolge abarbeiten zu lassen. Falls nur einer in einer Ebene eingetragen ist, besteht die Gruppe eben nur aus diesem Baustein. PreCOM schreibt das Programm so, daß die Folge mehrerer Aufrufe genau der Eintragsfolge entspricht.

Aufrufgruppe

wird jede Aufruffolge genannt, die PreCOM an den Aufrufbezeichner der vorhergehenden Ebene "übergibt", d.h. im AWL-Programm einen Aufruf (CALL) je Baustein einträgt.

Folglich ist das Hauptprogramm (z.B. bei Siemens-SPS mit OB1 als Aufrufbezeichner bekannt) eine Aufrufgruppe, die azyklisch von der SPS abgearbeitet wird. Azyklisch heißt es, weil die Aufrufe beginnen, wenn alle anderen Aufgaben des Betriebssystems erledigt sind. Die Zykluszeit ist also von der Dauer der Verarbeitung abhängig.

## 13.10 Baustein

Baustein heißt jeder Teil eines Steuerprogramms. Die DIN IEC 61131-3 kennt dafür den Begriff Programm-Organisations-Einheit (POE), ein Baustein ist eine POE. PreCOM erzeugt für jeden Teilprozeß einen Baustein, der als PRAP existiert, folglich mittels PRAP-Editor festgelegter Steuerungsablauf eines Teilprozesses ist, auch PRAP-Typ genannt.

Bausteine haben oft die Bezeichnung Funktionsbaustein und daher den Kurzbezeichner FB mit nachgestellter Nummer. Die DIN IEC 61131-3 nennt sie FUNCTION\_BLOCK.

Typ-Baustein [siehe Seite 126]

Instanzbaustein [siehe Seite 126]

ASCII-Baustein [siehe Seite 126]

## 13.11 Bezeichner und Namen

Bezeichner sind meist formale Abkürzungen für Variable und Bausteine, sie dienen der Identifikation [siehe Seite 172]. Hierzu gehören auch Bausteinbezeichner unter den konventionellen Sprachen die z.B. von den Produkten der Firma Siemens bekannten FB, FC (Funktionsbaustein, Funktion) usw. Instanzbausteine [siehe Seite 126] sind auf FB-Typen basierend. Sie sind wirklich ausführbare Funktionen mit wirklichen Adressen, sogenannten Aktual-Parametern.

DIN IEC 61131-3 gestattet eine freie Wahl der Zeichenfolge aus den Ziffern 0 .. 9, den Buchstaben a .. z, A .. Z und Unterstrich "\_", wobei das erste Zeichen ein Buchstabe oder ein Unterstrich sein muß.

## 13.12 Typ-Baustein / Typ-PRAP

Typen-Baustein oder Baustein-Typ kann jeder PRAP sein, der seinen funktionellen Ablauf als typische Steuerungsfunktion für einander gleich zu steuernde technologische Einheiten zur Verfügung stellt. Diese Formulierung weist darauf hin, daß ein Typen-Baustein nicht aufgerufen werden kann, ohne einen konkreten Bezug zu den zu verarbeitenden Variablen herzustellen. Einen solchen Bezug leistet der Instanzbaustein [siehe Seite 126].

Das Schalten eines Elektro-Motors kann eine solche typische Steuerungsfunktion sein. Für mehrere Motoren, die den gleichen Schaltablauf erfordern, kann ein Steuerungstyp entworfen werden. Ein solcher Typbaustein(PRAP-Type) oder Function Block PRAP(Type) erhält dann eine ihm zugehörige Identifikation.

Jede dieser PRAP-Typen ist zugleich Daten-Typ unter der "IDENTIFICATION" bzw. des dort eingetragenen Bezeichner-Textes.

Die DIN IEC 61131-3 definiert:

"SPS-Sprachelement, bestehend aus :

- (i) Der Definition einer Datenstruktur, geteilt in Eingangs-, Ausgangs- und interne Variablen und
- (ii) einer Menge von Operationen, die mit den Elementen der Datenstruktur durchgeführt werden, wenn eine Instanz des Funktionsbaustein-Typs aufgerufen wird."

## 13.13 Instanz-Baustein

Ein Instanzbaustein ist eine Programm-Organisationseinheit. Sie entnimmt ihre funktionellen Eigenschaften genau dem Typ-Baustein, dem sie ihre individuellen, technologisch relevanten Variablen-Parameter zur Verarbeitung übergibt.

Der Prozeßerlegungsgraph enthält ausschließlich Instanzen oder Unikate.

Nach DIN IEC 61131-3 ist der Instanzbaustein ein "individuelles, benanntes Exemplar einer Datenstruktur, verknüpft mit einem Funktionsbaustein-Typ oder Programm-Typ, das von einem Aufruf der zugehörigen Operationen bis zum nächsten erhalten bleibt."

Falls diese Zeilen zur Klärung nicht beitragen, könnten Sie unter dem Begriff Aufrufgruppe [siehe Seite 125] fündig werden.

## 13.14 PRAP- und Fremdprogramme (ASCII-Bausteine)

- **PRAP-Programme** sind

Anweisungslisten bzw. Befehlsfolgen (C/C++), die einem PRAP entsprechen, der mit dem PRAP-Editor erzeugt wurde. Zur Eigenschaft eines jeden PRAP-Programmes (Member markieren: rechte Maustaste | Eigenschaften oder Doppelklick mit linker Maustaste) gehören der Name und eine Bezeichnung des PRAP-Programmes (Identification, Statement/Description).

PRAP-Typen stellen ihre PRAP-Struktur (funktionelle Verknüpfung aller Parameter) und Formalparameter zur Verfügung, die PRAP-Instanzen (=Teilprozesse) durch Aktualparameter ersetzen.

Unikat- bzw. TP-PRAP sind Teilprozesse, die einmalig im durch die Prozeßerlegung vorliegenden

Steuerungs-System sind.

- **Fremd-Programme** sind Anweisungslisten oder andere Befehlsfolgen, die als Text-Datei (ASCII\_Baustein) in das Steuerprogramm eingebunden und beim Compilieren in den Maschinencode (Herstellere Software!) mit übersetzt werden (In dieser SPAS-Version noch nicht).

## 13.15 Project (Projekt)

Der Begriff Projekt ist mit einem Bezeichner versehen, einem Namen also, der für das gesamte Prozeß-Steuerungsvorhaben steht, das als Steuerungsprogramm erzeugt werden soll. Dieser Name wird einmalig beim Anlegen des Projektes vom User gegeben.

Ein Projekt beinhaltet im System SPAS mindestens eine Configuration (Konfiguration) [siehe Seite 144] und zu jeder Konfiguration gehört mindestens eine Resource (Ressource) [siehe Seite 187]. Zu jeder Resource gehören mindestens ein PROGRAM und eine TASK.

Der weitere Aufbau ist im Detail im Thema Projekt-Editor [siehe Seite 51] beschrieben.

## 13.16 Standard-Funktionen und -Funktionsbausteine

### Standard-Funktionen

sind in der DIN IEC 61131-3 im Abschnitt 2.5.1.5 beschrieben. SPAS verwendet davon die Typumwandlungen sowohl durch automatisches Einsetzen als auch durch manuelles. Typumwandlungen, die eindeutig durchführbar sind, z.B. INTEGER nach REAL, werden automatisch in das Compilat geschrieben. Eine Typumwandlung von INTEGER nach SHORTINTEGER oder WORD muß zwingend der Projektant verantworten und von Hand einsetzen.

Numerische Funktionen wie ABS, SQRT, LN oder alle trigonometrischen Funktionen sind vorbereitet. Da dies auch sprachabhängig ist, ob diese Funktionen verfügbar sind, ist der Precompiler noch nicht damit ausgestattet.

Arithmetische Funktionen sind außer MOD und EXPT verfügbar.

Bitschiebe-Funktionen sind alle verfügbar.

Bitweise boolesche Funktionen sind außer NOT verfügbar.

Auswahlfunktionen sowie Zeichenfolgefunktionen sind nicht verfügbar.

Vergleichsfunktionen sind vollständig verfügbar.

Funktionen für Datentypen der Zeit sind NUR für TIME verwendbar, alle anderen NICHT.

### Standard-Funktionsbausteine

Bistabile Bausteine wie RS-Flip-Flop werden grundsätzlich nicht benötigt und sind daher nicht verfügbar.

Flankenerkennung ist vollständig einsatzbereit.

Zählerbausteine werden alle mit dem Typen CTUD projektiert. Der Precompiler erkennt anhand der Zähl-Variablen CU bzw. CD, welcher Zähler benötigt wird.

Zeitbausteine werden grundsätzlich mit dem Typen TON projektiert. Falls die Ausschaltverzögerung benötigt wird, ist dies im PRAP zu beachten.

### Weitere Themen:

Allgemeine Funktionen, insbesondere selbst definierte (Funktionen [siehe Seite 154]) und (Einfügen einer Funktion in die Ressource [siehe Seite 58]) oder (der Funktions-Aufruf [siehe Seite 60])



# Kapitel 14

## Wörterbuch

### 14.1 Alphabet-Buttons

#### Hinweis auf eine Fehlfunktion:

Wenn Sie in den Abschnitt der Alphabet-Buttons durch Klick hineinspringen, dann die Maustaste erst loslassen, wenn Sie mit dem Cursor außerhalb der Buttons sind, ansonsten springt die Darstellung dorthin, wo der Cursor gestanden hat!

A [siehe Seite 129] B [siehe Seite 129] C [siehe Seite 129]  
 D [siehe Seite 130] E [siehe Seite 130] F [siehe Seite 130]  
 G [siehe Seite 130] H [siehe Seite 130] I [siehe Seite 130]  
 J [siehe Seite 131] K [siehe Seite 131] L [siehe Seite 131]  
 M [siehe Seite 131] N [siehe Seite 131] O [siehe Seite 131]  
 P [siehe Seite 131] Q [siehe Seite 132] R [siehe Seite 132]  
 S [siehe Seite 132] T [siehe Seite 132] U [siehe Seite 132]  
 V [siehe Seite 132] W [siehe Seite 133] X [siehe Seite 133]  
 Y [siehe Seite 133] Z [siehe Seite 133]



#### **Tip**

Falls Sie lieber die Buchstaben anwählen wollen, dann gehen Sie zurück zum "Handbuch" und finden unter "Begriffe in SPAS" das "Wörterbuch" mit Buchstabenwahl.

### 14.2 -A-

Access Table - Datenzugriff auf Datenbank [siehe Seite 148]  
 Adresse - allgemein [siehe Seite 123]  
 Adresse - direkt [siehe Seite 123]  
 Aktual-Parameter [siehe Seite 124]  
 ASCII-Baustein [siehe Seite 126]  
 Aufruf [siehe Seite 125]  
 Aufruf-Bezeichner [siehe Seite 125]  
 Aufrufgruppe [siehe Seite 125]  
 Arithmetik-Funktionen [siehe Seite 37]  
 Ausdruck [siehe Seite 138]  
 Ausgabe an Adresse (Kopier-Funktion) [siehe Seite 145]  
 Zurück zu den Buttons [siehe Seite 129]

### 14.3 -B-

Baustein [siehe Seite 125]  
 Bezeichner, Name, Identification [siehe Seite 126]  
 Bedingungsstabil [siehe Seite 139]  
 Beschreibung - Zulässige Zeichenfolge [siehe Seite 143]  
 Binärer Prozeß-Zustand [siehe Seite 182]  
 Binäre Prozeßanalyse [siehe Seite 14]  
 Binär-Vektor [siehe Seite 197]  
 Zurück zu den Buttons [siehe Seite 129]

### 14.4 -C-

Code [siehe Seite 69]  
Comment - Sequence of Letters and Digits [siehe Seite 143]  
Compilat [siehe Seite 69]  
Connectors, Sinn und Zweck und verwenden im PRAP [siehe Seite 168]  
Configuration [siehe Seite 144]  
Conversions of data types [siehe Seite 148]  
Zurück zu den Buttons [siehe Seite 129]

## 14.5 -D-

Data Type - Datentyp [siehe Seite 147]  
Datentyp-Wandel - Data type conversion [siehe Seite 148]  
Description - Sequence of Letters and digits [siehe Seite 143]  
Durchlauf [siehe Seite 150]  
dynamische Prozeßzustandsvariable [siehe Seite 150]  
Zurück zu den Buttons [siehe Seite 129]

## 14.6 -E-

Ebene [siehe Seite 150]  
Einzelübersetzung???  
Entwurfsprozeß [siehe Seite 13]  
Erzeugte Koppelung [siehe Seite 170]  
Zurück zu den Buttons [siehe Seite 129]

## 14.7 -F-

Fach-Sprache [siehe Seite 7]  
Funktionsbaustein-Bezeichner [siehe Seite 126]  
Formal-Parameter [siehe Seite 125]  
Function Block [siehe Seite 125]  
Formelobjekt [siehe Seite 153]  
Funktion [siehe Seite 154]  
Zurück zu den Buttons [siehe Seite 129]

## 14.8 -G-

Globale Variable [siehe Seite 154]  
Gerichteter Graph [siehe Seite 154]  
Zurück zu den Buttons [siehe Seite 129]

## 14.9 -H-

Halteprozeßzustand [siehe Seite 154]  
Hilfe-Texte, wie damit umgehen ... [siehe Seite 155]  
Zurück zu den Buttons [siehe Seite 129]

## 14.10 -I-

Identification - Zulässige Zeichen/Sequence of letters and digits [siehe Seite 172]  
Initial-Operation/Situation [siehe Seite 166]  
Instabile Operation/Situation [siehe Seite 150]  
Instabiler Zyklus [siehe Seite 166]  
Instanz-Baustein [siehe Seite 126]

Zurück zu den Buttons [siehe Seite 129]

## 14.11 -J-

(kein Stichwort)

Zurück zu den Buttons [siehe Seite 129]

## 14.12 -K-

Kombinatorische Struktur [siehe Seite 168]

Kommentar - Zeichen [siehe Seite 143]

Kompilat [siehe Seite 69]

Konfiguration [siehe Seite 144]

Konnektoren im PRAP [siehe Seite 168]

Konstante Größen [siehe Seite 124]

Koppel-Variable [siehe Seite 170]

Zurück zu den Buttons [siehe Seite 129]

## 14.13 -L-

Lokale Variable (Local Static, Local Temporary) [siehe Seite 195]

Zurück zu den Buttons [siehe Seite 129]

## 14.14 -M-

Member [siehe Seite 172]

Mensch-Anlagen-Kommunikation [siehe Seite 172]

Zurück zu den Buttons [siehe Seite 129]

## 14.15 -N-

NAME - Zulässige Zeichen/Sequence of letters and digits [siehe Seite 172]

Natürliche Koppelung [siehe Seite 170]

Nichtbinäre Funktionen [siehe Seite 25]

Zurück zu den Buttons [siehe Seite 129]

## 14.16 -O-

Operation [siehe Seite 174]

Operationen-Folge [siehe Seite 175]

Operationsvariable [siehe Seite 176]

Operationen-Zyklus [siehe Seite 176]

Zurück zu den Buttons [siehe Seite 129]

## 14.17 -P-

Parameter [siehe Seite 177]

Paarweise Steuerrelationr [siehe Seite 177]

Pfad [siehe Seite 177]

POE - Programmorganisationseinheit [siehe Seite 177]

PRAP [siehe Seite 61]

PRAP- und Fremdprogramme (ACSII-Bausteine) [siehe Seite 126]

PreCOM [siehe Seite 178]

PreCompiler-Ergebnis [siehe Seite 69]

Programm (PROGRAM)???

Project (Projekt) [siehe Seite 127]  
Projektbaum [siehe Seite 182]  
Prozeß-Schritt [siehe Seite 183]  
Prozeß-Situation [siehe Seite 183]  
Prozeß-Zerlegung [siehe Seite 17]  
Prozeßzustands-Variable [siehe Seite 183]  
Zurück zu den Buttons [siehe Seite 129]

## **14.18 -Q-**

(kein Stichwort)  
Zurück zu den Buttons [siehe Seite 129]

## **14.19 -R-**

Ressource [siehe Seite 187]  
Zurück zu den Buttons [siehe Seite 129]

## **14.20 -S-**

Situation [siehe Seite 183]  
Sprache, wählbare Sprache [siehe Seite 7]  
SPAS???  
Stabilitäts-Prozeßzustand [siehe Seite 154]  
Starker Zusammenhang [siehe Seite 190]  
Statement - Sequence of Letters and digits [siehe Seite 143]  
Statische Prozeßvariable [siehe Seite 188]  
Steuerung [siehe Seite 188]  
Steuerungs-System [siehe Seite 189]  
Strukturierte Datentypen - User Defined Types [siehe Seite 189]  
Struktur-Stabilität, strukurstabil [siehe Seite 189]  
Zurück zu den Buttons [siehe Seite 129]

## **14.21 -T-**

Task???  
Teil-AWL / Teil-Steuerung [siehe Seite 190]  
Teilprozeß [siehe Seite 191]  
Transition [siehe Seite 192]  
Typenteilprozeß [siehe Seite 126]  
Typenprozeßablaufplan [siehe Seite 193]  
Zurück zu den Buttons [siehe Seite 129]

## **14.22 -U-**

Übergangs-Prozeßzustand [siehe Seite 193]  
Umwandlung von Datentypen [siehe Seite 148]  
Unikat [siehe Seite 193]  
Zurück zu den Buttons [siehe Seite 129]

## **14.23 -V-**

Variable [siehe Seite 194]  
Variablen-tabelle - Access Table [siehe Seite 148]  
Variable Usage, Verwendung von Variablen [siehe Seite 195]

Vergleicherfunktion [siehe Seite 196]  
Vollständigkeit [siehe Seite 197]  
Zurück zu den Buttons [siehe Seite 129]

## **14.24 -W-**

Wandelung der Datentypen [siehe Seite 148]  
Wählbare Sprachen [siehe Seite 7]  
Wertebelegung [siehe Seite 197]  
Widerspruchsfreiheit [siehe Seite 197]  
Zurück zu den Buttons [siehe Seite 129]

## **14.25 -X-**

(kein Stichwort)  
Zurück zu den Buttons [siehe Seite 129]

## **14.26 -Y-**

(kein Stichwort)  
Zurück zu den Buttons [siehe Seite 129]

## **14.27 -Z-**

Zeit- und Zählglieder [siehe Seite 197]  
Zustand [siehe Seite 197]  
Zustands-Merker [siehe Seite 197]  
Zyklus, Operationenzyklus [siehe Seite 176]  
Zurück zu den Buttons [siehe Seite 129]



## **III Anhang**



# Anhang Erläuterungen

## A.1

- Für Typen-PRAP werden normalerweise keine Adressen angegeben.
- Die Adressenangabe [siehe Seite 123] für eine Variable ist wichtig für Instanzen.
- Haben Sie einen PRAP als Unikat entworfen, dann tragen Sie hier die betreffende Adresse ein.
- Es gibt Fälle, in denen durch die Beschränkung des Betriebssystems nur eine bestimmte Anzahl Formalparameter zulässig ist. Dann geben Sie hier die Hardwareadressen ein, die im Typen-PRAP festzulegen sind.

## A.2

Zu jeder Variablen ist ein Datentyp auszuwählen. Auch konstante Werte benötigen einen Datentyp, um sie den Variablen anpassen zu können.

## A.3

Der Anfangswert einer Variablen kann hier angegeben werden. Gewöhnlich stellen die Betriebssysteme der Ressourcen beim Laden des Steuerungsprogramms alle Werte auf Null. Dieser Initial-Wert wird aber häufig NICHT beim Betriebswechsel der Ressource von STOP auf RUN der Variablen übertragen. Nutzen Sie in der Initialoperation durch die Kopierfunktion (enthalten in Function: Calculate) Anfangswerte auf eine Variable übertragen werden können.

Sollte eine Konstante (z.B. fester Vergleichswert, Zeitwert, Zählwert) eingeführt werden, dann hier die Zahl eintragen. Eine Konstante wird nicht überschrieben, also behält sie ihren Wert.

**Zeitwerte** mit Zeitmaßen versehen, also 1,5h (nicht "Stunden"!) oder 28min5sec, auch 28M5S, oder 575ms. Stets "TIME" als Datentyp wählen und "CONSTANT" als Usage. Wollen Sie einen veränderlichen Zeitwert dem Timer übergeben, dann 0 eintragen oder das Feld leer lassen, Usage ist dann "INPUT", "INPUT/OUTPUT" oder "LOCAL STATIC" je nach Speicherplatz.

**Zählwerte** als Ganzzahlen angeben und Datentyp "Integer" wählen und "CONSTANT" als Usage. Variable Werte wie bei Zeitwerten.

## A.4

Hier nur dann Eintragungen vornehmen, wenn es sich um direkte Adressen handelt, also bei Instanzen. Ansonsten können schon Hinweise auf EMSR-Stellen stehen, z.B. LIS für Füllstands-Anzeige und -Steuerung.

Falls Sie Betriebsmittel-Kennzeichen wie z.B. "Y" für Antriebsort verwenden, ist dies ebenso möglich. Kopplungen können hier ebenfalls eingetragen werden, wenn sie bei Typen überhaupt schon angebar sind.

## A.5

Klicken Sie auf den -Button!

Sie erhalten die nötigen Hinweise zu "NAME".

## A.6

## PARAMETER

können formal und actual sein.

Im Typ-PRAP sind im Normalfall alle Variable formal. Es gibt allerdings zwei Ausnahmen:

- Ein Unikat enthält trotz Typ-PRAP-Eigenschaft für alle Variablen direkte Adressen [siehe Seite 123].
- Ein Typ-PRAP im gemischten Fall, also teils formale, teils aktuelle Parameter, tritt auf, wenn die Anzahl der formalen Parameter vom Hersteller so beschränkt wird, daß nicht alle Variablen formal erklärt werden können. Hierzu zählt z.B. die Siemens-Reihe S7-200.

## A.7

Die Art der Verwendung der Variablen - Variable Usage - wird hierher übernommen. Hier ist keine Eingabemöglichkeit. Ändern Sie bitte im zutreffenden Formular.

## A.8 All Variables - Anzeige aller im Projekt deklarierten Variablen

In dieser Tabelle können Sie alle Variablen manipulieren, deshalb ist äußerst sorgfältig zu arbeiten.

## A.9

Ein Ausdruck ist kein Druckerzeugnis eines Druckers sondern ein arithmetisches Gebilde, das im Textformat entworfen wurde, auch mit "strukturierter Text" bezeichnet.

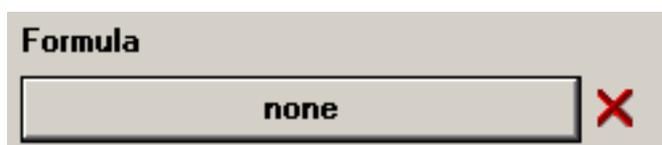
## A.10

### **AusgangsvARIABLE**

sind Größen aller Datentypen, die

- im PRAP meist Formalvariable sind, also ohne direkte Adressenangabe. Aber sie sind bezüglich Nutzung (USAGE) "Output" bzw. "Input/Output" oder "Local Static/Temporary". Es können auch Konstante mit USAGE "Constant" verwendet werden.  
Operations-Variable sind immer vom Datentyp "BOOL", sie können als "Output" oder "Local Static" genutzt werden.
- im PROGRAM (TYP) Aktualvariable sind, also direkte Adressenangaben tragen. Dann ist die Nutzung (USAGE) "Address Output" oder "Address Memory".  
Keine direkte Adressenangabe erfolgt bei im PROGRAM lokal bleibenden Variablen, also USAGE ist dann "Local Static".  
Konstante Werte mit USAGE "Constant" sind ebenfalls möglich.
- im Teil-Prozeß, bzw.in der Instanz werden keine AusgangsvARIABLEN deklariert, aber diejenigen aus dem PROGRAM zugewiesen, dem der TP untergeordnet ist.

## A.11



Mit Klick auf "nothing / none" lesen Sie zur

Bestätigung, ob ein Formelobjekt hergestellt werden soll. Nach einem Klick auf "Yes" erhalten Sie, falls erforderlich, ein Formular zum Konfigurieren der gewählten Funktion. Ansonsten öffnet der Variablenzugriff().

Das rote Kreuz daneben gestattet, die gewählte oder eine zu entfernende Funktion, wenn sie also bereits konfiguriert ist, zu löschen.



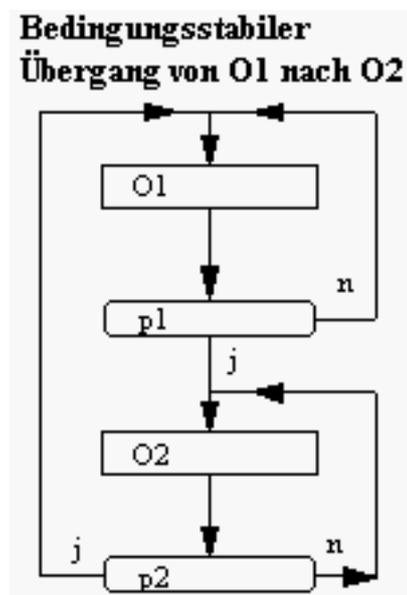
### Warnung

Nach beendeter Bearbeitung des Formelobjektes und seiner Variablen kehrt das Fenster () der Prozeß-Variablen bzw. Operationsvariablen zurück. Nun müssen Sie "OK" anklicken (Enter-Taste), um die Ergebnisse der Konfigurationsarbeit zu speichern. "Cancel" macht alle Arbeit ungeschehen!

Allgemeine Erläuterungen:

Nichtbinäre Funktionen (im Buch Entwurfsprozeß)???

## A.12



Ein Übergang von einer Operation Oj zur Folgeoperation Ok heißt

bedingungsstabil, wenn der Haltezustand der Folgeoperation Ok nicht abhängig von den Bedingungen des Übergangszustandes der Operation Oj zur Operation Ok ist. In der Abbildung ist zu sehen, daß der Haltezustand der Operation O2 durch die Prozeßvariable p2 = nein-Belegung realisiert wird und nicht durch die Prozeßvariable p1.

## A.13

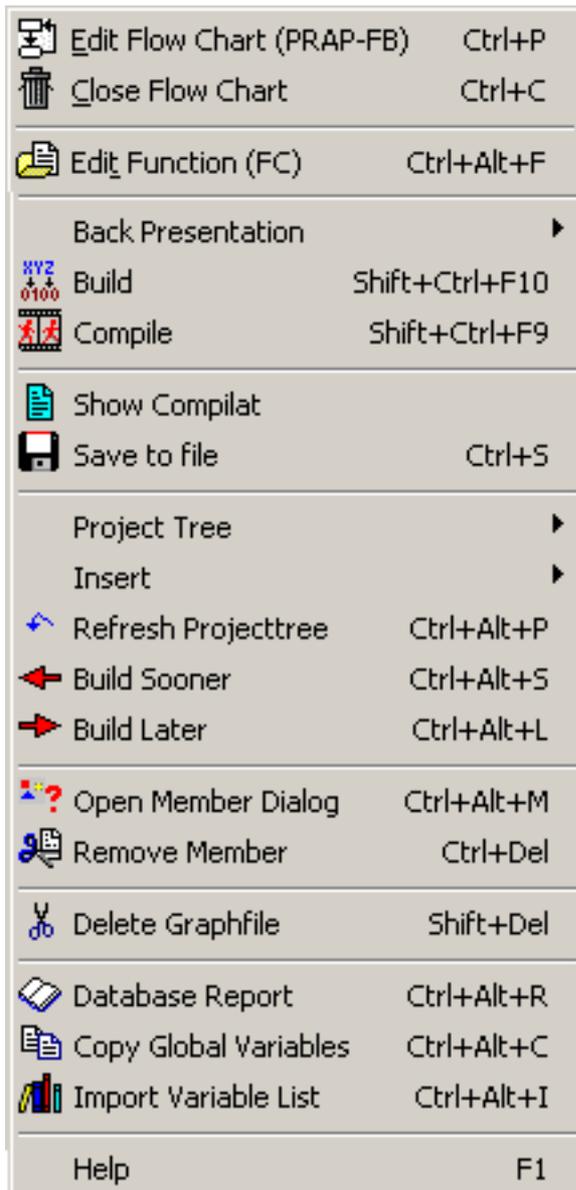
Mit einem Klick auf "Work Book" wird Ihnen mitgeteilt, wieviel "Work Sheets" Ihre Variablen-Tabelle enthält. Im Feld neben "Work Sheet" die Nummer angeben<sup>27</sup>, ab der die Variablen-Tabelle gelesen werden soll.

## A.14

### Neu entwerfen oder bearbeiten eines PRAP:

1. PRAP-Member markieren und
2. rechte Maustaste öffnet das Projekt-Menü:

<sup>27</sup> Geben Sie hier an, welches der insgesamt vorhandenen Blätter übernommen werden soll. Danach Klick auf "Work Sheet", um mitgeteilt zu bekommen, falls die Nummer nicht existiert.



Klick "PRAP load" öffnet das Panel "CHARTS" genau für das markierte Member, mit oder ohne PRAP. Nach dem Import eines PRAP, der mit TeCAD entworfen wurde, müssen Sie hier die Graphik in eine vernünftige Form bringen.

Für alle diese Zwecke gibt es den PRAP-Editor [siehe Seite 61], den Sie erreichen, sobald Sie durch Betätigen der rechten Maustaste ein Menü öffnen, auf dem Sie "Graphic Editor" anklicken müssen.

Klick "PRAP close" schließt den PRAP-Editor. Ob Sie den PRAP speichern wollen oder nicht, werden Sie gefragt, und entscheiden Sie richtig!

## A.15

**Das PopUp-Menue gestattet folgende Auswahl zur Verknüpfung der Booleschen Operations-Variablen mit einer "Nichtbinären Funktion":**

- not Specified  
Die Variable ist ohne Verknüpfung
- Create Timer  
Die Variable schaltet ein Zeitglied (Timer bzw. Timer Function) ein und aus. Außerdem werden in den Programmtext Befehle geschrieben, die für eine Zeitgliedfunktion entscheidend sind.

- Create Counter  
Die Variable schaltet einen Zähler (Counter bzw. Counter Function) ein und aus. Außerdem werden in den Programmtext Befehle geschrieben, die für eine Zählgliedfunktion entscheidend sind.
- Count Down  
Die Variable bewirkt mit jedem Wechsel des Wertes von false auf true (0 auf 1)<sup>28</sup> das Rückwärtszählen eines anzugebenden Zählgliedes.
- Count Up  
Die Variable bewirkt mit jedem Wechsel des Wertes von false auf true (0 auf 1)<sup>29</sup> das Vorwärtszählen eines anzugebenden Zählgliedes.
- Calculate/Copy(norm) und Calculate(single)
  - Calculate(multiple) und Calculate(single):  
(multiple) bedeutet:  
Die Variable bewirkt bei einem Wechsel des Wertes von false auf true (0 auf 1) die Ausführung der Berechnung, die in einem arithmetischen Ausdruck angegeben ist. Mit dem Wechsel von 1 auf 0 schließt die Berechnung. Die Dauer einer Berechnung ist gleich der Dauer der stabilen Situation, so daß das Ergebnis beim letzten Berechnungszyklus gilt. Die Anzahl der Berechnungen, deshalb "multiple", hängt von der Geschwindigkeit der CPU ab.  
(single) bedeutet:  
Mit der 0-1-Flanke der Operationsvariablen erfolgt automatisch eine einmalige Berechnung des arithmetischen Ausdruckes. Die stabile Situation kann beliebig länger dauern, das Ergebnis ändert sich nicht mehr. Sollten während der stabilen Situation Änderungen der Werte bei Operanden erwartet werden, beeinflussen sie natürlich nicht das Ergebnis. Beispiel für "single"<sup>30</sup>  
Die Besonderheit ist, daß auch Schiebe- und Rotier-Operatoren verwendet werden dürfen.  
Der arithmetische Ausdruck wird wie üblich berechnet, d.h. Klammerausdrücke werden zuerst, Schiebe- und Rotieroperatoren vor \* und /, danach + und -.
  - Copy:  
Die Variable bewirkt mit jedem Wechsel des Wertes von false auf true (0 auf 1) das Übertragen, d.h. die Zuweisung (:=), von einer Konstanten auf einen Speicherplatz oder den Inhalt eines Speicherplatzes auf einen anderen. Dabei erfolgt automatisch eine Datentyp-Anpassung, falls das nötig ist. Ebenso können Sie Bitwerte 0 oder 1 auf Boolesche Variablen übertragen.  
Auch bei "Calculate(single)" können Sie eine Kopierfunktion projektieren. Das unterschiedliche Verhalten ist oben beschrieben.

## A.16

**Das PopUp-Menue gestattet die Auswahl zur Verknüpfung der Booleschen Prozeß-Variablen mit einer "Nichtbinären Funktion":(idh\_notbinfc [siehe Seite 25])**

- not Specified  
Die Variable ist ohne Verknüpfung mit einer nichtbinären Funktion, also eine normale, Boolesche Variable.

<sup>28</sup> Ein Wechsel des Wertes ist nur von einer Operation zur ihr folgenden Operation möglich.

Der Wert bleibt von einer zur nächsten Operation erhalten, wenn die Markierung dieselbe ist (Haken oder nicht).

<sup>29</sup> Ein Wechsel des Wertes ist nur von einer Operation zur ihr folgenden Operation möglich.

Der Wert bleibt von einer zur nächsten Operation erhalten, wenn die Markierung dieselbe ist (Haken oder nicht).

<sup>30</sup> Falls Sie z.B. wegen großer Zählwerte keinen Zähler sondern lieber eine Addition mit 1 projektieren wollen, dann beginnen Sie mit E=0 (E=Ergebnis, z.B. mit INT als Datentyp). Formulieren Sie E := E + 1, dann wollen Sie mit jedem Erreichen der Operation, in der die Operationsvariable von 0 auf 1 wechselt, E um den Wert 1 erhöhen. Würden Sie "multiple" gewählt haben, erhöht sich E bei jedem SPS-Zyklus um 1! Also müssen Sie hier "single" wählen.

- Timer Output Q  
Q ist das Kuzzeichen für den Ausgang eines Timers, den Sie zuvor konfiguriert haben müssen. Die Variable(Flag) ändert ihren Wert von 0 auf 1, wenn der anzugebende Timer "abgelaufen" ist, d.h. der ihm zugeordnete Zeitwert ist erreicht.
- Count Down  
Die Variable bewirkt mit jedem Wechsel des Wertes von false auf true (0 auf 1) das Rückwärtszählen eines anzugebenden Zählgliebes.
- Count Up  
Die Variable bewirkt mit jedem Wechsel des Wertes von false auf true (0 auf 1) das Vorwärtszählen eines anzugebenden Zählgliebes.
- Counter Output Q  
Q ist das Kuzzeichen für den Ausgang eines Counters, den Sie zuvor konfiguriert haben müssen. Die Variable(Flag) ändert ihren Wert von 0 auf 1, wenn der anzugebende Counter "abgelaufen" ist, d.h. der ihm zugeordnete Zählwert ist erreicht.
- Comparison  
Ein Vergleich zwischen zwei Werten, also auch Inhalten von Speicherplätzen, soll immer Istwert - Vergleich mit - Sollwert> lauten. Wird "Vergleich mit" erfüllt, dann erhält die "Ergebnis"-Variable(Flag) den Wert 1 (true), ansonsten 0 (false). Istwerte können mit dieser Vorgabe keine Konstanten sein, sind folglich stets Speicherplätze. Die Anpassung der Datentypen erfolgt automatisch. Ist- und Sollwert dürfen arithmetische Ausdrücke sein.
- Raising Edge  
Die "Ergebnis"-Variable(Flag) (Condition) erhält mit jedem Wechsel des Wertes der Eingangs-Variablen (Primary Variable) von false auf true genau einen CPU-Zyklus lang den Wert 1.
- Falling Edge  
Die "Ergebnis"-Variable(Flag) (Condition) erhält mit jedem Wechsel des Wertes der Eingangs-Variablen (Primary Variable) von true auf false genau einen CPU-Zyklus lang den Wert 1.
- Raising and Falling Edge  
Die "Ergebnis"-Variable(Flag) (Condition) erhält mit jedem Wechsel des Wertes der Eingangs-Variablen (Primary Variable) von true auf false und umgekehrt genau einen CPU-Zyklus lang den Wert 1.

## A.17

Nach dem Compilieren können Sie das Ergebnis, das sog. Compilat hier ansehen.

Sie können Members wie Ressourcen, PRAP-Typen und Programme compilieren. Dazu Member



Dann wie folgt vorgehen:

1. Click Build or Compile

Bemerkungen zum Übersetzungslauf im unteren Anzeigefeld bzw. bei "Compiler Notes".

2. Click "Show Compilat" zeigt bei CODE den Quellcode.

3. If required: Click Save to File

Es werden alle erforderlichen Files erzeugt, die zum Generieren des Maschinen-Codes nötig sind. Außerdem wird immer ein File \*.onl bereitgestellt, wodurch die Darstellung des "lebendigen PRAP" (online debugging) möglich ist.

Das Compilat ist nicht veränderbar, editierbar. Jede Änderung wird ausschließlich in den PRAP bzw. in der Datenbank, im Memberdialog oder im Projektbaum durchgeführt.

## A.18

Es sind alle druckbaren Zeichen erlaubt. Die Anzahl ist mit 64 begrenzt.

Ein Zeilenumbruch mit "Enter" ist nicht zugelassen. Zu lange Zeilen werden automatisch bei einem Leerzeichen umgebrochen. Wollen Sie eine Text-Trennung vorher erreichen, dann mit "-" oder auch mehr Leerzeichen.

Eine Text-Trennung kürzerer Texte ist mit "\n " zu notieren; falls ein "-" erwünscht ist, muß dies vorher stehen. Das Leerzeichen nach \n nicht vergessen!

## A.19

In dieses Feld schreiben Sie einen kurzen Text, der aussagt, was technologisch im gesteuerten (Teil-)Prozeß geschehen soll. Mit Klick auf "Return" erfolgt die Übernahme in das Operations-Symbol (Kästchen)

## A.20

In dieses Feld tragen Sie in der Reihenfolge

IstwertVergleichsoperationSollwert

die Vergleichsformel ein. Ist- und Sollwert sind Variable, die Sie aus dem Angebot wählen - anklicken! - können, wofür sie den Tastenschlüssel Strg+Leerzeichen benutzen (Ctrl+Space).(Eingabemöglichkeiten für Variable in Formelausdrücken [siehe Seite 196])

Die Vergleichsoperationen wählen Sie - anklicken! - bitte aus dem Angebot mit dem Tastenschlüssel Strg+J (Ctrl+J) oder nutzen die Zeichen der Tastatur. Sie dürfen auch arithmetische Ausdrücke anstelle eines Istwertes oder Sollwertes eintragen.

**WICHTIG:** FormulaHandling [siehe Seite 153]

## A.21

Praktisch kommen nur drei Möglichkeiten aus dem Angebot in Frage:

- Input / Input/Output  
Falls Ist- bzw. Sollwert auf einer Input- oder Outputadresse liegen.
- Local Static  
Falls sich die Variable innerhalb des Bausteins im RAM befindet. Zum Beispiel kann durch die Kopierfunktion auf einem Speicherplatz ein Wert abgelegt werden.
- Constant  
Falls der Sollwert (rechts vom Vergleichsoperator) eine feste Zahl ist, die in "Initial" eingetragen werden muß nach Klick auf . Beachten Sie im Falle einer konstanten Zeit "T#" voranzustellen und anschließend Zahl und Einheit(en), z.B. T#12s120ms.

## A.22

### COMPILER NOTES

Sowohl die Datenbank-Maschine als auch der Precompiler und das Check-Programm für PRAP schreiben hier ihre Meldungen, die Sie vom Erfolg oder Mißerfolg einer Aktion unterrichten. Mit der rechten Maustaste erhalten Sie ein paar Handlungsmöglichkeiten.

## A.23

Wählen Sie einen technologie- bzw. auf die Anlage bezogenen Text als Bezeichner.

## A.24

Configuration (Konfiguration)

bezeichnet ein Automatisierungssystem mit einer oder mehreren Steuerungseinheiten (einschließlich Regelungen). Hier werden die beteiligten Steuerungseinheiten, wie sie SPS oder andere darstellen, aufgeführt. Jede Steuerungseinheit ist eine Ressource und enthält ihre eigenen Steuerungsaufgaben, die über ein Netzwerk mit den anderen zur Konfiguration gehörenden Ressourcen im Datenaustausch stehen kann.

## A.25

Globale Variable und solche, die der Kommunikation über Busverbindungen dienen (Access Variable), sind derzeit noch nicht eingebbar.

## A.26

Werte müssen auf Adressen ausgegeben werden, wenn Zeitwerte, Zählwerte und Sollwerte für Regler, Vergleichler, sowie Grenzwerte für Alarme und anderem nicht unabhängig vom Prozeßgeschehen festgelegt werden können. Damit ist die Möglichkeit gegeben, von der Prozeßsituation abhängig unterschiedliche Werte vorzugeben. Diese Funktion führt praktisch ein Kopieren von Werten durch. Unter dem Thema "nichtbinäre Funktionen" können Sie mehr erfahren.

## A.27

Einen **Kopierauftrag (Copy)** formulieren Sie ganz einfach durch:

Zielspeicher := Quelle ;

Quelle kann eine Konstante sein aber auch ein Speicherplatz. Der Datentyp soll "kleiner" oder gleich dem des Zielspeichers sein, wobei damit die Bitanzahl bzw. der Zahlenumfang gemeint ist. Die Datentypanpassung erfolgt automatisch. Sie dürfen alle Datentypen verwenden, weil es beim Kopieren nicht auf Vorzeichen ankommt. Aus BYTE, WORD oder DWORD eine Größe mit Vorzeichen "herstellen zu lassen", also z.B. von WORD nach INT wandeln gelingt nur mit ausdrücklichem Befehl WORD\_TO\_INT, weil der Programmierer dies sehr bewußt anzuweisen hat!

Der Zielspeicher darf auch eine Boolesche Variable sein, der Sie entweder 0 oder 1 als Konstante zuweisen können.

Einen **Berechnungsauftrag (Calculate)** tragen Sie nach Ihrem Ermessen ein. Entscheiden Sie, ob das Ergebnis nur im ersten Arbeitszyklus - "single" - berechnet werden darf oder in allen - "multiple" - berechnet werden kann. Mehr hierzu: idh\_Arithmetik [siehe Seite 37]

Das Problem der Datentypen liegt darin, daß Byte, Word, Dword nicht "verrechenbar" sind wegen des fehlenden Vorzeichens. U-Typen, wie z.B. UINT, werden als positive Werte auf dem nächst größeren Bitformat verarbeitet, sofern überhaupt möglich, denn 64-Bit-Maschinen sind noch selten.

Als Berechnungs-Regel gilt, zuerst Klammerausdrücke, dann Shift und Rot und trigonometrische Funktionen, danach Potenzen, Multiplikation und Division, zum Schluß Addition und Subtraktion.

**WICHTIG:** FormulaHandling [siehe Seite 153]



### Tipp

Klick zuerst CHECK danach RETURN!

Sie dürfen in mehrere Zeilen nacheinander eine Formel schreiben, die immer mit ";" endet.

Sie dürfen mehr als eine Formel in das Formula-Fenster schreiben. Die Berechnung erfolgt in der Reihenfolge, wie die Formeln eingetragen sind.

## A.28

- Input  
kann nur für alle Operanden rechts vom Zuweisungszeichen gelten, falls nicht lokale Speicherplätze verwendet werden.
- Input/Output  
kann für alle Operanden beiderseits vom Zuweisungszeichen gelten. Operanden rechts vom Zuweisungszeichen können Ergebnisse der Kopierfunktion oder anderer arithmetischen Ergebnisse sein und nicht auf lokalen Speicherplätzen liegen.

- **Output**  
trifft vorrangig für den Operanden links vom Zuweisungszeichen zu, sofern das Ergebnis nicht lokal abgelegt werden kann.
- **Local Static**  
Falls sich die Variable innerhalb des Bausteins im RAM befindet. Zum Beispiel kann durch die Kopierfunktion auf einem lokalen Speicherplatz ein Wert abgelegt werden. Zeitgliedwerte und Zählerwerte sind ebenfalls lokal.
- **Constant**  
Falls der Operand (rechts der Zuweisung) eine feste Zahl ist, die in "Initial" eingetragen werden muß (nach Klick auf )

## A.29 Zählglied-Ausgang / Counter Output Q

Bitte beachten Sie zwei Möglichkeiten. Sie erwarten die Ausgabe mit true auf der P-Variablen (Condition):

- Es handelt sich um einen im PRAP kreierte Counter, dann suchen Sie "NAME" dieses Counters und übernehmen alle weiteren Eintragungen, Daten.
- Es handelt sich um einen Counter in einem anderen PRAP, dann suchen Sie in diesem MEMBER nach "NAME" des Counters und übernehmen alle weiteren Eintragungen, Daten.

## A.30

- **Name**  
Nach Klick auf  geben Sie eine Kurzbezeichnung zum Statement ein.
- **Statement**  
Tragen Sie hier eine Bezeichnung für den Zählwert ein, z.B. "Anzahl der Pakete".
- **Variable Usage**  
Wählen Sie "Constant" für einen festen Wert, ansonsten "Input".
- **Value**  
Den Wert in Zahlenform am besten bei "Initial Value" eintragen, z.B. 145. Der Datentyp ist sprachabhängig, im allgemeinen INT.

## A.31

### Name

Eine Kurzbezeichnung bitte eingeben, nachdem Sie mit Klick auf  die Datentabelle geöffnet haben.

### Statement

Geben Sie dem Zähler eine Bezeichnung. Aber vermeiden Sie z.B. Z34 zu schreiben, weil das nichtssagend ist. Schreiben Sie dagegen "Anzahl der Mehlsäcke", ist auch nach Jahren noch der Zweck des Zählers klar.

### Variable Usage

im allgemeinen "Local Static" (In Version 1. noch nicht: "Address Memory" für fest vorgegebene Counter, z.B. Z4) oder ist sprachabhängig. [siehe Seite 7]

## A.32 Zähler auf-/abwärts zählen - Count Up/Down

Suchen Sie "Name" des Zählgliebes (Datatype = Counter), das mit jedem Wechsel der Variablen (P- oder O-Variable) von false auf true um den Wert 1 verringert oder erhöht werden soll.

Anschließend markieren und Klick auf .

## A.33

Task-Anforderungen bedingen auch ein Betriebssystem, das derzeit nur begrenzt unterstützt wird. Es gibt nur für die Priorität 0 einen Aufruf im freien Zyklus. Daher brauchen Sie auch keine Auswahl oder Eintragung durchzuführen.

## A.34

### DATA BASE ADMINISTRATION

Die Zugriffe auf die Datenbank erscheinen bei verschiedenen Aktionen, um notwendige Eintragungen bezüglich der verwendeten Variablen vorzunehmen oder Änderungen durchzuführen.

Falls Sie in der oberen Leiste "VIEW" anklicken, erhalten Sie mehrere Angebote, sich Auszüge aus der Datenbank anzusehen. Nutzen Sie bitte dort die Hilfe-Texte.

## A.35

Die Eigenschaft "Datentyp" einer Variablen bestimmt die Codierung der zur Verfügung stehenden Bit, zur Darstellung bzw. Speicherung der Variablen.

Der Typ **BOOL** nutzt nur ein Bit, wobei die Adressierung von Hersteller zu Hersteller der SPS recht unterschiedlich ist. In SPAS wird nach dem Byte.Bit-Prinzip adressiert. Die Kennung E, A, M oder I, O, M am einfachsten nicht davorschreiben, da sie automatisch ergänzt wird.

Byte- und Word-Typen (**BYTE**, **WORD**, **DWORD**, **LWORD**) ebenso wie Unsigned-Typen (**USINT**, **UINT**, **UDINT**, **ULINT**) schöpfen den Wertebereich der 8, 16, 32 oder 64 Bit voll aus. Unsigned-Typen sind bei konventionellen SPS wie S5 oder S7 derzeit NICHT verwendbar!

Integer-Typen (**SINT**, **INT**, **DINT**, **LINT**) nutzen das höchste Bit (niedrigstes Bit ist Bit 0), also Bit 7, 15, 31 oder 63 als Vorzeichen-Kennung, wodurch sich der Wertebereich etwa halbiert.

**TIME** ist ein Datentyp identisch mit **DINT**, jedoch vordefiniert mit der Einheit Millisekunde. Damit sind folglich  $2,1474 \cdot (10 \text{ hoch } 9)\text{ms}$  als Höchstwert erreichbar, das sind  $2,1474 \cdot (10 \text{ hoch } 6)\text{s}$  oder  $35791,33\text{min}$  oder  $596,522\text{h}$ .

Real-Typen (**REAL**), auch Gleitkomma-Zahlen genannt, haben 32 Bit gemäß der Internationalen Norm IEEE 754 (aus dem Jahr 1985), wobei das Bit 31 als Vorzeichen genutzt wird, die Bit 30 bis 23 dienen zur Darstellung des Exponenten e und die Bit 22 bis 0 der Mantisse. Damit sind die Werte

$-3.402823\text{E}+38 \dots -1.175495\text{E}-38, 0, +1.175495\text{E}-38 \dots +3.402823\text{E}+38$  darstellbar.

Neben diesen Typen gibt es noch Funktions-Baustein-Typen:

**TON**, **TIME** - Zeitbausteine /TIMER mit Zeitwert

**CTUD** - Zählbausteine mit Zähler-Werten i.a. vom Typ **INT**

**R\_TRIG**, **F\_TRIG** und **RF\_TRIG** - Flanken-Erkennung

und andere. Hierzu gehören PRAP-Typen, die mit ihrer IDENTIFICATION im Projekt eingebunden sind.

Die BCD-Typen (**BCD**) sind in der IEC 1131-3 nicht vorgesehen, 16 und 32 Bit groß und stellen nur Ziffern im Bereich 0 bis 9 auf Tetraten (4 Bit) dar. Sie werden wie Word-Typen behandelt und sollten nur für Ausgaben z.B. an Ziffernanzeigen benutzt werden, wobei **INT\_TO\_BCD** bzw. **DINT\_TO\_BCD** zur Verfügung gestellt werden.

Weiterhin können abgeleitete, strukturierte Datentypen eingesetzt werden: Strukturen [siehe Seite 189]

## A.36 Datenaufzeichnung und Rückverfolgung

Zunächst müssen Sie alle Handlungen durchführen, die unter Einführung [siehe Seite 167] und Inbetriebnahme [siehe Seite 166] stehen.

tragen Sie nun in "Cycletime" einen Wert ein. Dabei sind immer sieben Stellen vorgegeben, achten Sie also darauf, daß der eingetragene Wert in Millisekunden auch Ihren Vorstellungen entspricht. Nun mit Mausklick auf die Glühlampe 

und Sie können die Aufzeichnungsschritte mit beobachten, d.h. nur deren Nummer. Alle Daten werden an die Datei [Dateiname].onl angehängt. Mit dem Explorer können Sie das verfolgen.

Durch wiederholten Klick auf die Glühlampe beenden Sie die Aufzeichnung. Klicken Sie danach auf die Kamera, und das Verfolgen der aufgezeichneten Daten und Schritte kann beginnen. Dazu müssen Sie nur auf die Schritt-Tasten



klicken.

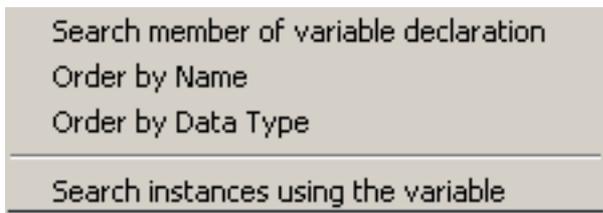
## A.37

### Informations-Bereich zum gefilterten Datenzugriff und zu Daten der Variablen

Bei geöffneter Access Table<sup>31</sup> können Sie über jede Variable folgende Daten erhalten:

- Zugehörigkeit zum Member. Bei mehreren Zugehörigkeiten alle nacheinander.
- USAGE zu jedem Member
- Formula falls eine nichtbinäre Funktion mit der Variablen verbunden ist.

Sie gelangen zu diesen Informationen, indem Sie den Cursor auf die betreffende Variable in der Tabelle führen und die rechte Maustaste klicken. Es öffnet sich:



Order by Name bzw. Order by Data Type beziehen sich auf die Anordnung in der Tabelle. Ein Klick auf "Search Variable Usage" liefert Ihnen im Informationsfeld die Daten.

#### HINWEIS:

Die Größe des Feldes ist sehr unterschiedlich, je nach der Variablen-tabelle darüber und dem zugehörigen Variablen-Dialogfeld. Sie können die Trennlinie nach oben schieben.

## A.38

### Datentyp-Wandel, Type conversion functions

Datentypen sind geordnet nach ihrer Bit-Anzahl und ihrer Kodierung, d.h. Bedeutung der Bit oder Bitgruppen. In diesem Sinne ist der kleinste Datentyp "BOOL" und der größte "REAL" bzw. "LREAL". Die "L"-Datentypen nutzen 64 Bit, derzeit gibt es keine Verwendung.

Es gibt für spezielle Zwecke auf 16-Bit-Typen eine BCD-Zahlendarstellung, die je Tetrade (4bit) nur 0 bis 9 zulassen (sprachabhängig! [siehe Seite 7]). Die Norm IEC 1131-3 erfaßt BCD nicht als Datentyp.

Die sogenannte Aufwärts-Konvertierung von SINT zu REAL wird für arithmetische Ausdrücke, Kopierfunktion, auch Vergleicherausdrücke, automatisch vollzogen. Das bedeutet, Sie dürfen in solchen Ausdrücken, auch in reinen Zuweisungen, also in Kopierfunktionen, die Variablen ohne Datentypwandel schreiben.

Jeder Typenwandel "abwärts" ist durch eine Konvertierungsfunktion auszudrücken, wobei es KEINE KONTROLLE gibt, ob der Inhalt unzulässig verstümmelt wird. Hierzu gehören auch Übergänge zwischen Word- und Integertypen.

Zusammenstellung zu Datentyp-Konvertierungen einer Variablen:

<sup>31</sup> Der Inhalt ist dem Beispielprojekt entnommen.

- **AUTOMATISCHE KONVERTIERUNG:**  
 SINT-->INT-->DINT-->REAL  
 REAL-->DINT mit Rundungsfunktion ROUND  
 TIME<-->DINT  
 INT<-->WORD  
 BCD-->INT / WORD  
 BCD-->DINT / DWORD
- **KONVERTIERUNGSFUNKTIONEN** einer Variablen var:  
 DINT\_TO\_REAL(var)  
 REAL\_TO\_DINT(var) mit Rundungs-Funktion ROUND  
 TRUNC(var) mit Abschneiden der Nachkommazahlen  
 REAL\_TO\_INT(var) ohne Wertekontrolle  
 DINT\_TO\_INT(var) ohne Wertekontrolle  
 DINT\_TO\_DWORD(var)  
 INT\_TO\_SINT(var) ohne Wertekontrolle  
 INT\_TO\_BCD(var) drei Stellen mit Vorzeichen  
 INT\_TO\_WORD(var)  
 DINT\_TO\_BCD(var) sieben Stellen mit Vorzeichen  
 DWORD\_TO\_DINT(var)  
 WORD\_TO\_INT(var)

Diese Liste wird laufend aktualisiert.



### Anmerkung

Bitte Konvertierungsfunktionen NICHT schachteln.  
 Beispiel: REAL\_TO\_DINT(DINT\_TO\_INT(var))

## A.39

STEP 7 (S7-300 und S7-400) wird mit Funktionsbausteinen erzeugt. Alle Aktualparameter [siehe Seite 124] werden in einem Instanz-Datenbaustein übergeben.

Globalvariablen in der Ressource sind in einem Datenbaustein, dessen Nummer einzugeben ist, enthalten. Wenn keine Globalvariablen in der Ressource deklariert sind, sollte keine Nummer eingegeben werden.

Bemerkungen zu Instanz-Datenbausteinen:

- Ein Multi-Instanz-Datenbaustein enthält alle Aktualparameter, deshalb tragen Sie NUR IN "PROGRAM INSTANCE" eine DB-Nr. ein. Alle FB-Instanzen ohne Eintrag bei DB Setting.
- Einzel-Instanz-Datenbausteine enthalten zu jeder FB-Instanz die Aktualparameter. Geben Sie hier nur die DB-Nummer ein!  
 Trotzdem MUSS die PROGRAM INSTANCE einen DB erhalten!



### Anmerkung

Einwandfreies Nachladen geänderter Instanzen bei online laufender CPU (Schalterstellung auf "RUN-P") ist nur bei Einzelinstanzen möglich!

**Tip**

Der Multi-Instanz-DB ist zu bevorzugen. Falls er über 64K Daten enthält, lösen Sie das Problem entweder durch Einzelinstanzen oder fügen ein weiteres PROGRAM ein, in das Sie Aufrufe aus dem "überfüllten" PROGRAM eintragen.

**A.40**

Den Datenzugriff ("Data Access") erhalten Sie auch direkt, wenn Sie zwei Mal hintereinander die linke Maustaste klicken (Doppelklick).

**A.41**

Ein Durchlauf (= Run Through) tritt auf, wenn zu einer Operation der Halteprozeßzustand nicht eintritt, bzw. nicht durch die Belegung der Prozeßvariablen erfüllt wird.

**A.42**

Eine Prozeßvariable heißt dynamisch, wenn sie nur für die Dauer und zum Zweck eines Übergangs ihre Aussage wechselt. Daher sind z.B. bei Flankenerkennungen die nur einen SPS-Zyklus mit dem Wert 1 belegten "Ergebnis"-Variablen dynamischer Natur.

**A.43**

Im Projektbaum sind Ebenen nicht von oben nach unten sondern von links nach rechts definiert. Die erste Ebene ist dem Projekt vorbehalten, die zweite Ebene Konfigurationen, die dritte Ressourcen. Mehr darüber im Thema Projekt-Editor???

**A.44**

- Name  
ist ein Kurzzeichen zu Statement und nur in der Variablen-tabelle (Data Access) editierbar. Es ist diejenige Variable, die nur einen Maschinenzklus lang den logischen Wert 1 hat.
- Statement  
ist die Benennung der "Eingangs- oder Primary Variable". Schreiben Sie hier einen Text, der zeigt, wofür sie nötig ist.
- Variable Usage  
kann Input, Output oder Local Static sein.

**A.45**

Mit Klick auf diesen Button öffnet sich das Fenster zur Bearbeitung von Variablen (ganz allgemein) für die ausgewählte Variable.

Wollen Sie eine Variable löschen, dann markieren Sie diese Variable, und mit rechtem Mausklick wird das Löschen angeboten.

**A.46****EingangsvARIABLE**

sind Größen aller Datentypen, die

- im PRAP meist Formalvariable sind, also ohne direkte Adressenangabe. Aber sie sind bezüglich Nutzung (USAGE) "Input" bzw. "Input/Output" oder "Local Static/Temporary". Es können auch Konstante mit USAGE "Constant" verwendet werden.  
Prozeß-Variable sind immer vom Datentyp "BOOL", sie können als "Input" oder "Local Static" genutzt werden.
- im PROGRAM (TYP) Aktualvariable sind, also direkte Adressangaben tragen. Dann ist die Nutzung (USAGE) "Address Input" oder "Address Memory".  
Keine direkte Adressenangabe erfolgt bei im PROGRAM lokal bleibenden Variablen, also USAGE ist dann "Local Static".  
Konstante Werte mit USAGE "Constant" sind ebenfalls möglich.
- im Teil-Prozeß, bzw.in der Instanz werden keine Eingangsvariablen deklariert, aber diejenigen aus dem PROGRAM zugewiesen, dem der TP untergeordnet ist.

## A.47

Wählen Sie für jedes Feld den Spaltenbuchstaben gemäß Ihrer Tabelle!

Nicht enthaltene Spalten müssen nicht in der Liste angelegt werden, geben Sie einfach einen Buchstaben außerhalb der Liste an.

Als Hilfe betrachten Sie ein Beispiel: <sup>32</sup>

32

|    | A                | B                                 | C      | D     | E        | F              | G                    | H                   | I         | J              | K            | L                                    | M                   |            |
|----|------------------|-----------------------------------|--------|-------|----------|----------------|----------------------|---------------------|-----------|----------------|--------------|--------------------------------------|---------------------|------------|
|    | Act. Name        | Actual Statement                  | Prefix | Addr. | Qt. Type | USAGE          | Init. Val.           | Point of Automation | TP Name   | PRAP Name      | Form. Param. | USAGE                                | Formal Comment      |            |
| 1  | Halltor_groefln  | Hallentor geöffnet                | E      | I 10  | BOOL     | Address Input  | GS 0308              | Innenbel            | Lampe     | Lx_einschalten | Input        | Schließbedingung erfüllt             |                     |            |
| 2  | Halle_beleuchtet | Halle beleuchtet                  | A      | I 10  | BOOL     | Address Output | EU 9302              | Innenbel            | Lampe     | Lx_EIN         | Output       | Lampe einschalten                    |                     |            |
| 3  |                  |                                   |        |       |          |                |                      |                     |           |                |              |                                      |                     |            |
| 4  | Halltor_geschl   | Hallentor geschlossen             | E      | O 0   | BOOL     | Address Input  | GS 0608              | Aussenbel           | Lampe     | Lx_einschalten | Input        | Schließbedingung erfüllt             |                     |            |
| 5  | Hallenruf_bel    | Hallenrufkahn beleuchten          | A      | O 0   | BOOL     | Address Output | EU 9602              | Aussenbel           | Lampe     | Lx_EIN         | Output       | Lampe einschalten                    |                     |            |
| 6  |                  |                                   |        |       |          |                |                      |                     |           |                |              |                                      |                     |            |
| 7  |                  |                                   |        |       |          |                |                      |                     |           |                |              |                                      |                     |            |
| 8  | Tor_oefln        | Tor öffnen                        | M      | K 0   | BOOL     | Address Memory | KDP von Koordination | Tor_AUF             | St_Dr_Anl | Mot_ein        | Input        | Motor einschalten                    |                     |            |
| 9  | ferngest_AUF     | Öffnen ferngesteuert              | M      | I 0   | BOOL     | Address Memory | KDP von Koordination | Tor_AUF             | St_Dr_Anl | FgS            | Input        | Ferngesteuert schalten               |                     |            |
| 10 | YHd_oefln        | Öffnen von Hand                   | E      | I 12  | BOOL     | Address Input  | HS 0206              | Tor_AUF             | St_Dr_Anl | YHd_Mot_ein    | Input        | Motor von Hand einschalten           |                     |            |
| 11 | NOT_Tast         | Öffnen NOT-AUS                    | E      | I 30  | BOOL     | Address Input  | HZ 0202              | Tor_AUF             | St_Dr_Anl | NOTAUS         | Input        | NOT-Aus                              |                     |            |
| 12 | (I_AUF)          | Schaltwert zum Öffnen erreicht    | M      | O 0   | BOOL     | Address Memory | KDP von L_MDT_AUF    | Tor_AUF             | St_Dr_Anl | I_Schalt       | Input        | Strom hat Weicherschaltwert erreicht |                     |            |
| 13 | Oefln_Haupt      | Hauptschütz Öffnen EIN            | A      | I 11  | BOOL     | Address Output | EU 0203              | Tor_AUF             | St_Dr_Anl | S1             | Output       | Hauptschütz einschalten              |                     |            |
| 14 | Oefln_Stern      | Sternschütz Öffnen EIN            | A      | I 12  | BOOL     | Address Output | EU 0203              | Tor_AUF             | St_Dr_Anl | S2             | Output       | Sternschütz einschalten              |                     |            |
| 15 | Oefln_Drei       | Dreieckschütz Öffnen EIN          | A      | I 13  | BOOL     | Address Output | EU 0203              | Tor_AUF             | St_Dr_Anl | S3             | Output       | Dreieckschütz einschalten            |                     |            |
| 16 | Oefln_AUS        | Öffnen ausgeschaltet              |        |       |          | BOOL           | Local Static         | KDP nach L_MDT_AUF  | Tor_AUF   | St_Dr_Anl      | Mot_aus      | Output                               | Motor ausgeschaltet |            |
| 17 | T_Anl_AUF        | AnlaufTIMER Öffnen                |        |       |          |                | TON                  |                     | Tor_AUF   | St_Dr_Anl      | T_Anl        | Input                                | Anlauf-TIMER        |            |
| 18 |                  |                                   |        |       |          |                |                      |                     |           |                |              |                                      |                     |            |
| 19 | Tor_Schli        | Tor schließen                     | M      | K 1   | BOOL     | Address Memory | KDP von Koordination | Tor_ZU              | St_Dr_Anl | Mot_ein        | Input        | Motor einschalten                    |                     |            |
| 20 | ferngest_ZU      | Schließen ferngesteuert           | M      | I 11  | BOOL     | Address Memory | KDP von Koordination | Tor_ZU              | St_Dr_Anl | FgS            | Input        | Ferngesteuert schalten               |                     |            |
| 21 | YHd_Schli        | Schließen von Hand                | E      | O 2   | BOOL     | Address Input  | HS 0508              | Tor_ZU              | St_Dr_Anl | YHd_Mot_ein    | Input        | Motor von Hand einschalten           |                     |            |
| 22 | (NOT_Tast)       | Schließen NOT-AUS                 | E      | I 30  | BOOL     | Address Input  | HZ 0502              | Tor_ZU              | St_Dr_Anl | NOTAUS         | Input        | NOT-Aus                              |                     |            |
| 23 | (I_ZU)           | Schaltwert zum Schließen erreicht | M      | O 1   | BOOL     | Address Memory | KDP von L_MDT_ZU     | Tor_ZU              | St_Dr_Anl | I_Schalt       | Input        | Strom hat Weicherschaltwert erreicht |                     |            |
| 24 | Schli_Haupt      | Hauptschütz Schließen EIN         | A      | O 1   | BOOL     | Address Output | EU 0501              | Tor_ZU              | St_Dr_Anl | S1             | Output       | Hauptschütz einschalten              |                     |            |
| 25 | Schli_Stern      | Sternschütz Schließen EIN         | A      | O 2   | BOOL     | Address Output | EU 0501              | Tor_ZU              | St_Dr_Anl | S2             | Output       | Sternschütz einschalten              |                     |            |
| 26 | Schli_Drei       | Dreieckschütz Schließen EIN       | A      | O 3   | BOOL     | Address Output | EU 0501              | Tor_ZU              | St_Dr_Anl | S3             | Output       | Dreieckschütz einschalten            |                     |            |
| 27 | Schli_AUS        | Schließen ausgeschaltet           |        |       |          | BOOL           | Local Static         | KDP nach L_MDT_ZU   | Tor_ZU    | St_Dr_Anl      | Mot_aus      | Output                               | Motor ausgeschaltet |            |
| 28 | T_Anl_ZU         | AnlaufTIMER Schließen             |        |       |          |                | TON                  |                     | Tor_ZU    | St_Dr_Anl      | T_Anl        | Input                                | Anlauf-TIMER        |            |
| 29 |                  |                                   |        |       |          |                |                      |                     |           |                |              |                                      |                     |            |
| 30 | I_AUF            | Schaltwert zum Öffnen erreicht    | M      | O 0   | BOOL     | Address Memory | KDP nach Tor_AUF     | L_MDT_AUF           | L_MDT     | I_Schalt       | Output       | Weicherschaltwert erreicht           |                     |            |
| 31 | (Oefln_AUS)      | Öffnen ausgeschaltet              |        |       |          | BOOL           | Local Static         | KDP von Tor_AUF     | L_MDT_AUF | L_MDT          | Mot_aus      | Input                                | Motor ausgeschaltet |            |
| 32 | I_Soll_AUF       | Soll-Stromwert beim Öffnen        |        |       |          | INT            | Constant             | 12345               | L_MDT_AUF | L_MDT          | I_Soll       | Input                                | Weicherschaltwert   |            |
| 33 | I_LIST_AUF       | Ist-Stromwert beim Öffnen         |        |       |          | PELV252        | INT                  | Address Input       | EF 0101   | L_MDT_AUF      | L_MDT        | I_LIST                               | Input               | Motorstrom |
| 34 |                  |                                   |        |       |          |                |                      |                     |           |                |              |                                      |                     |            |
| 35 | I_ZU             | Schaltwert zum Schließen erreicht | M      | O 1   | BOOL     | Address Memory | KDP nach Tor_ZU      | L_MDT_ZU            | L_MDT     | I_Schalt       | Output       | Weicherschaltwert erreicht           |                     |            |
| 36 | (Schli_AUS)      | Schließen ausgeschaltet           |        |       |          | BOOL           | Local Static         | KDP von Tor_ZU      | L_MDT_ZU  | L_MDT          | Mot_aus      | Input                                | Motor ausgeschaltet |            |
| 37 | I_Soll_ZU        | Soll-Stromwert beim Schließen     |        |       |          | INT            | Constant             | 12333               | L_MDT_ZU  | L_MDT          | I_Soll       | Input                                | Weicherschaltwert   |            |
| 38 | I_LIST_ZU        | Ist-Stromwert beim Schließen      |        |       |          | PELV250        | INT                  | Address Input       | EF 0201   | L_MDT_ZU       | L_MDT        | I_LIST                               | Input               | Motorstrom |
| 39 |                  |                                   |        |       |          |                |                      |                     |           |                |              |                                      |                     |            |

## A.48

Tragen Sie hier eine technologisch zutreffende Bezeichnung ein, die auch mehrere Worte haben darf, um deutlich zu beschreiben, was die Instanz im Projekt steuern soll.

Empfehlung:

Verwenden Sie den Text aus dem Prozeßzerlegungsgraphen.

## A.49

Teilen Sie der Instanz ein technologisch sinnvolles Kürzel zu, das natürlich innerhalb der Ressource nicht wiederholt werden darf.

Empfehlung:

Verwenden Sie die Bezeichnung aus dem Prozeßzerlegungsgraphen.

## A.50

Die Tabelle enthält alle zum Typ gehörigen "FORMAL"-Parameter, denen Sie "ACTUAL"-Parameter zuweisen müssen, die zur Instanz gehören. Die Zuweisung fehlt bei allen mit "none" bezeichneten Aktualparametern.

Klicken Sie auf  und erhalten alle in PROGRAM deklarierten Variablen und der Aktual-Parameter [siehe Seite 212] mit ihren direkten Adressen angezeigt:

Die Anzeige erfolgt in der Form PROGRAM.Variable.

Es fehlen alle lokal deklarierten (Local Static, Local Temporary), da diese automatisch Variable aus dem RAM zugewiesen erhalten.

## A.51

Die Beziehung einer Instanz zu einem PRAP-TYP ist zwingend. Aus dem Angebot ist der zugehörige Typ auszuwählen.

Bei IEC 1131-3 vorgesehen: Neben der Task zur Programminstanz kann auch für eine FB-Instanz eine extra Aufruforderung erforderlich sein. Dann ist eine bei der Ressource deklarierte Task auszuwählen.

## A.52

STEP 7 (S7-300 und S7-400) wird mit Funktionsbausteinen erzeugt. Geben Sie hier nur die FB-Nummer ein!

## A.53

Tragen Sie hier eine technologisch zutreffende Bezeichnung ein, die auch mehrere Worte haben darf, um deutlich zu beschreiben, was die Funktion bewirken soll.

## A.54

Der Name der aufzurufenden Funktion ist der gleiche wie die IDENTIFICATION der deklarierten Funktion in der Ressource. Er ist nicht esitierbar, wählen Sie die Funktion bei "RELaTIONS" aus!

## A.55

Tragen Sie hier eine technologisch zutreffende Bezeichnung ein, die auch mehrere Worte haben darf, um deutlich zu beschreiben, was die Funktion allgemein bewirken soll.

## A.56

STEP 7 (S7-300 und S7-400) wird mit Funktionen erzeugt. Geben Sie hier nur die FC-Nummer ein!

## A.57

Hier werden alle zur Funktion zugeordneten Variablen angezeigt, sofern sie in einer funktionellen Verarbeitung ("Edit Function (FC)") bekannt sind.

Bearbeiten können Sie Variable entweder einzeln nach Klick auf  oder nach Öffnen des "Edit Function (FC)".

## A.58

## A.59

Eine Funktion

- erhält ihren Namen durch die Deklaration, ein neuer wird nicht zugeordnet, wohl aber hier aus dem "Vorrat" der FUNCTIONS gewählt.
- ist mit keiner Task verbindbar, daher hier keine Editiermöglichkeit.

## A.60

Die Filter-Auswahl ist ein- und ausschaltbar mit dem "Filter-Button". Die Filterbedingungen sind unter "Filter Options" wählbar und mit "Filter Values" zu spezifizieren. Wird z.B. "Member" markiert, dann muß aus den angegebenen Members bei Values das Zutreffende gewählt werden. Bleiben die anderen Options unmarkiert und der Filterbutton wird angeklickt, dann werden alle Variablen des Members angezeigt.

"Parameter" ist die Art der Variablen, ob es formale oder aktuelle mit fester Adresse sind. Diese Auswahl ist sinnvoll bei Typen-PRAP mit teilweise direkten Adressen.

Die Markierung "Data Type" gestattet Ihnen eine besondere Auswahl aus allen Variablen.

Die Markierungen "Parameter" und "Data Type" spezifiziert aus allen Variablen des Projektes.

Sie können für ein Member sinnvoll kombinieren:

- "Member" und "Parameter"
- "Member" und "Data Type"
- "Member" und "Parameter" und "Data Type"

## A.61

Der Begriff "Formelobjekt" soll aussagen, daß zum Erfassen aller Daten für die gewählte Funktionalität aus dem Angebot der "Nichtbinären Funktionen" ein entsprechender Datenbank-Eintrag erfolgt.

## A.62

- Nach einer Neu-Einführung einer Variablen erfolgt mit Klick auf  (ACCEPT) sowohl die Übernahme in die Formel bzw. Funktion als auch in die Datenbank und wird damit auch dem Member zugeordnet.

- Nach Änderungen an Daten zu einer Variablen - Ausgenommen "NAME" - in der Datenbanktabelle mit Klick auf  diese Änderungen in die Datenbank eintragen.



### Warnung

Bei Klick auf ACCEPT wird diese Variable nochmal an der Stelle des Cursors eingetragen. Veränderte Eintragungen zur Variablen NICHT!

- Nach jeder Auswahl ein Leerzeichen einfügen als Trennzeichen zwischen Operanden und Operationszeichen.
- Solange Sie den Cursor innerhalb einer Variablen stehen haben, werden deren Eigenschaften Name, Statement und Variable Usage oben angezeigt.
- Am Ende der Formel ein ";" anfügen. Damit wird deutlich, daß der Ausdruck hier abschließt.  
Denn:
  - Sie dürfen die Formel, den Ausdruck, auf mehrere Zeilen verteilen.
  - Sie dürfen auch mehrere arithmetische Ausdrücke - ausgenommen sind folglich Vergleiche! - eintragen. Die Abarbeitung der Formeln erfolgt in dieser Reihenfolge.
  - Sie dürfen in diesem Feld wie mit jedem einfachen Texteditor arbeiten, müssen also nicht mit den angegebenen Tastenschlüsseln Ctrl+Space bzw. Ctrl+J operieren, aber es erleichtert das Eintragen.

## A.63

Eine Funktion ist gemäß IEC 1131-3 als eine Programm-Organisationseinheit definiert, die genau ein Datenelement liefert bzw. hervorbringt. Datenelemente können neben elementaren Typen auch Strukturen sein. Zur Deklaration sind nur Input und Local zugelassen.

SPAS verwendet in der Version 1.4 die Funktion mit freiem Ergebnis bzw. offener Rückgabe (Function xxxx : VOID), also sind Variablendeklarationen Input und Output zugelassen und natürlich Local.

## A.64

### Globale Variable

sind Variable mit einem Geltungsbereich innerhalb einer Konfiguration, einer Ressource, einem Programm. Diese Variablen sind in den jeweils untergeordneten POE als externe Variable nutzbar. Insbesondere dienen sie der Instanzbildung der Typen-PRAP in "PROGRAM".

## A.65

Unter einem zusammenhängenden, gerichteten Graph versteht man eine Abbildung, die aus Knoten (z.B. Kreisen) und gerichteten Kanten (Pfeile zwischen den Knoten) besteht.

## A.66

Als Stabilitäts-Prozeßzustand = Self Transition (kurz: Stabilitätszustand) bzw. Halteprozeßzustand (kurz Haltezustand) werden alle Pfade bezeichnet, die eine betrachtete Operation  $O_i$  mit sich selbst verbinden. Aus Sicht der Graphentheorie bilden  $O_i$  und der Stabilitätszustand  $P_{ii}$  einen Zyklus. Ein Haltezustand ist Voraussetzung zur Ausführung der zugehörigen Operation. Der Halteprozeßzustand ist folglich nur eine spezielle Form eines Übergangsprozeßzustandes mit gleicher Start- und Zieloperation.

Ergibt sich durch die Belegung der Prozeßvariablen zu einer erreichten Operation kein Pfad des Haltezustandes, so wird sie durchlaufen.

Werden mehrere Operationen durchlaufen, so daß die Operation  $O_i$  wieder erreicht wird, spricht man von einem instabilen Zyklus. Eine Operation zusammen mit ihrem Halteprozeßzustand bzw. Stabilitätsprozeßzustand, nennt man Situation, die solange erhalten (stabil) bleibt, wie der Haltezustand erfüllt wird.

Jede Situation wird im Compilat als (Automaten)Zustand behandelt.

## A.67

Es ist zu erwarten, daß Sie so manchen Begriff nicht verstehen, dann versuchen Sie bitte im Buch "Begriffe in SPAS" danach zu suchen. Im Wörterbuch sind viele Begriffe alphabetisch geordnet enthalten.

Die sogenannte kontextsensitive Hilfe erhalten Sie, wenn Sie in einem Formblatt, bzw. einem Fenster, den Cursor in ein veränderbares Eintragsfeld (Editierfeld) plazieren und die Taste "F1" drücken, falls kein  vorhanden. Falls aber ein  oben rechts eingetragen, können Sie direkt z.B. in Editierfeldern nach Hilfe suchen (Achtung funktioneller Fehler<sup>33</sup>). Benutzen Sie auch den "Hilfe"-Button, falls vorhanden, um eine Hilfe-Übersicht zu erhalten, von der aus Sie auch die "Fragezeichen-Themen" erreichen können. Bei Menüs werden entweder mit "Hilfe"-Button oder mit "F1" Erläuterungen zu den Menü-Zeilen angezeigt.

Bitte auch am unteren Rand des SPAS-Manager-Bildes die Hinweis-Texte beachten!

Bevor Sie verzweifeln, weil Sie einen Hilfetext nicht verstehen: hier klicken<sup>34</sup>

## A.68 Deklaration der TASK, VAR\_GLOBAL und VAR\_EXTERNAL in RESOURCE



### Anmerkung

Vorbemerkung zu SPAS Version 1:

Derzeit sind VAR\_EXTERNAL nicht deklarierbar!

VAR\_GLOBAL sind Deklarationen aller innerhalb der Ressource erforderlichen Aktual-Parameter, auch mit den zugehörigen direkten Adressen. Hierzu gehören alle Variablen, die der Kommunikation der Programme innerhalb der Ressource, z.B. Koppelungen, dienen und nicht schon in der Konfiguration (in Version 1 nichts in Konfiguration deklarieren) stehen!



Die Bedeutung der Datenbank-Symbole entnehmen Sie bitte den Hinweisen auf der unteren Zeile des Fensters.

Tragen Sie in NAME [siehe Seite 172] den Variablenbezeichner ein, der hier als Symbol für die anzugebende Hardwareadresse gilt. Im Feld USAGE. [siehe Seite 195] beschränkt sich die Auswahl auf Constant, Address Input, Address Output und Address Memory (E., A., M. bzw. %I., %O., %M.). In das Feld ADDRESS geben Sie nur noch die Ziffernfolge für die Adresse ein! Falls erforderlich, die Byte-Bit-Trennung nicht vergessen. Bei Klick auf  erfolgt die Übernahme der bei "NAME" eingetragenen Zeichenfolge einschließlich Datentyp, Adresse und "USAGE".

Zum Schluß nicht vergessen auf "OK" zu klicken!!

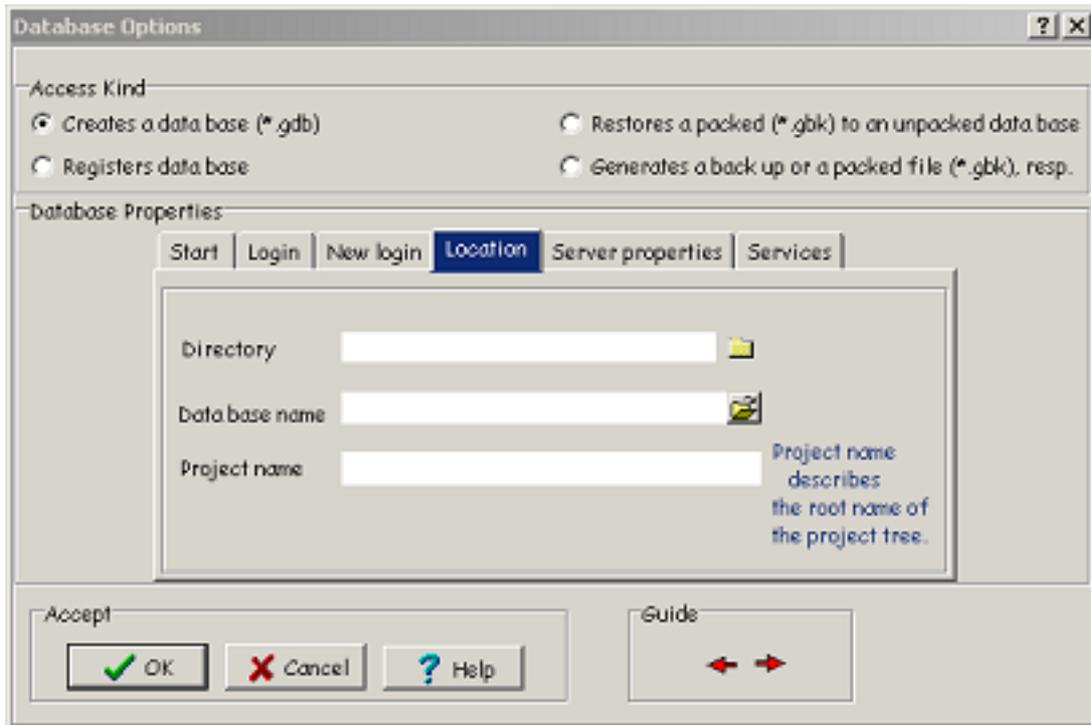
<sup>33</sup> Erscheint in einem Hilfetext, aufgerufen mit , ein weiterer grüner, gestrichelt markierter Text, so erhalten Sie eine Meldung, es sei nicht genug

Speicherplatz vorhanden usw. Das ist noch eine Fehlfunktion des Programms Time2Help. Verwenden Sie dann den Help-Button!!

<sup>34</sup> Rufen Sie an: 0341/30761183 und verlangen Herrn Dr. Alder oder

Schreiben Sie eine e-mail: calder@mail.htwk-leipzig.de

## A.69



- **Directory**

Falls für das neue Projekt noch kein Verzeichnis (Folder) vorhanden ist, legen Sie es bitte z.B. mit dem Explorer an.

Das Verzeichnis darf auch auf einem Server liegen, es beginnt dann mit \\[ServerName] usw.

Wenn ein Verzeichnis für das neue Projekt angelegt ist:

Suchen Sie nach Klick auf das Folder-Symbol  das Verzeichnis, in dem die Datenbank Ihres **neuen Projektes** gespeichert werden soll.

- **Database Name**

Der hier einzutragende Name für ein **neues Projekt** erhält in Ihrem Directory die Erweiterung ".gdb" (Groton Data Base). In dieser Datei sind sämtliche Projektdaten enthalten! Überlegen Sie sich diesen Namen gut, da er nicht änderbar ist. Verwenden Sie keine Umlaute oder 'ß'.

Das Registrieren eines **vorhandenen Projektes** beginnt mit Klick auf .

- **Project Name** = Project Alias<sup>35</sup>

Der Eintrag steht im Projektbaum hinter dem Symbol .

Diese Bezeichnung ist später nicht mehr änderbar! Deshalb wählen Sie einen technologisch sinnvollen Bezeichner, nicht etwa "Projekt 1", sondern z.B. "Futtermischanlage" oder "Ersatzteil-Lager".

Alles zu Data Base Options [siehe Seite 206]

## A.70

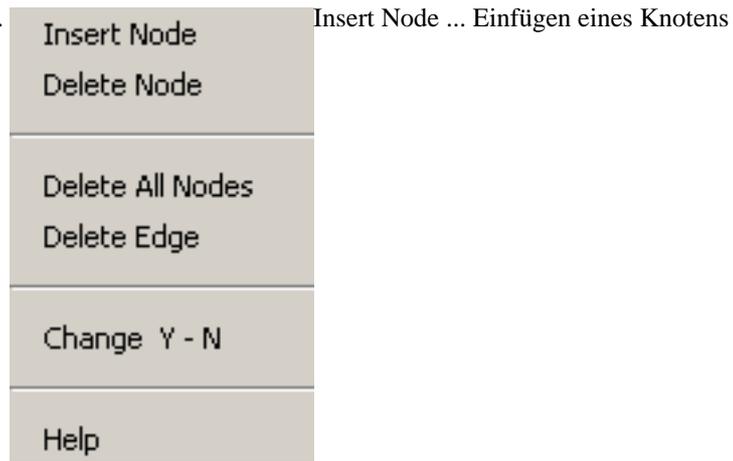
Mit diesem Menue bearbeiten Sie eine ausgewählte gerichtete Verbindung (Edge), auch Kante genannt, zwischen zwei PRAP-Objekten.

Ein Knoten dient dazu, die Kante dort "anzupacken" (linke Maustaste festhalten) und einen Knick zu erzeugen.

Jede der zwei Edges einer Condition muß eine Bewertung/Aussage Yes oder No tragen. Änderungen der

<sup>35</sup> Die Zeichenfolge, die den Bezeichner des Projektes darstellt, muß einmalig sein, bezogen auf das "Database Login". Der Projektname dient außerdem der Registrierung beim Betriebssystem des PC.

Aussage werden hier ermöglicht.



Delete Node ... Löschen eines Knotens

Delete All Nodes ... alle Knoten der Kante werden gelöscht

Delete Edge ... die markierte Kante wird gelöscht

Change Y - N ... Wechsel der Kanten-Bewertung von Ja auf nein oder umgekehrt

## A.71



Save Graph: Speichern des PRAP und aller Variablen in der

Datenbank. Voraussetzung zur Compilierung( Build und Compile) nach Änderungen im PRAP

Check Graph: Vollständige Überprüfung der Graphik

Reload Graph: Neue Darstellung des PRAP aus dem Datenbank-Speicher

Graph Editor: Der Editor für eine PRAP-Entwicklung erscheint als Menü

Print: Ausdruck der Graphik über einen Drucker, mit Formatierungen

Copy to Clipboard: Die Graphik wird in die Zwischenablage als Bitmap kopiert

Zoom: Vergrößerung / Verkleinerung der graphischen Darstellung

Descriptions: Öffnet ein Fenster für die sog. Auftragsdaten und die Beschreibung oder Erläuterung der PRAP-Funktion.

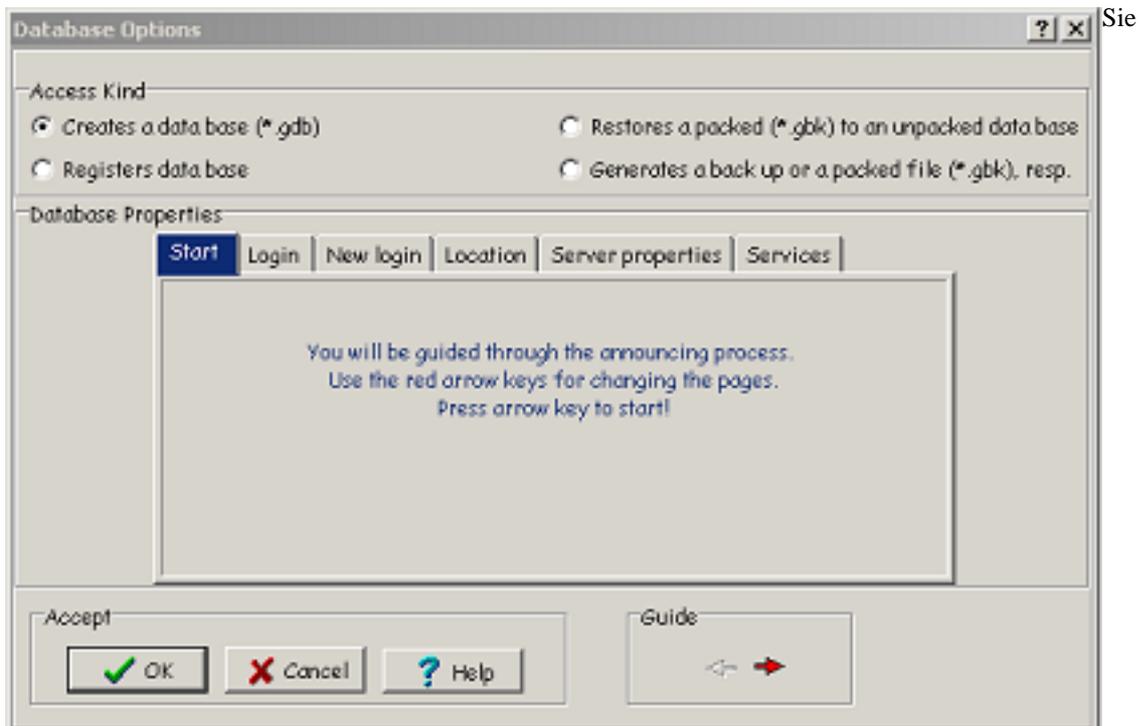
Clear Graphic Panel: Das Entfernen der graphischen Darstellung bedeutet nicht deren Speicherung!



### Tipp

**Die vollständige Prüfung eines PRAP** geschieht während des Compilierens. "Build" für jeden PRAP ist daher sehr zu empfehlen!

## A.72

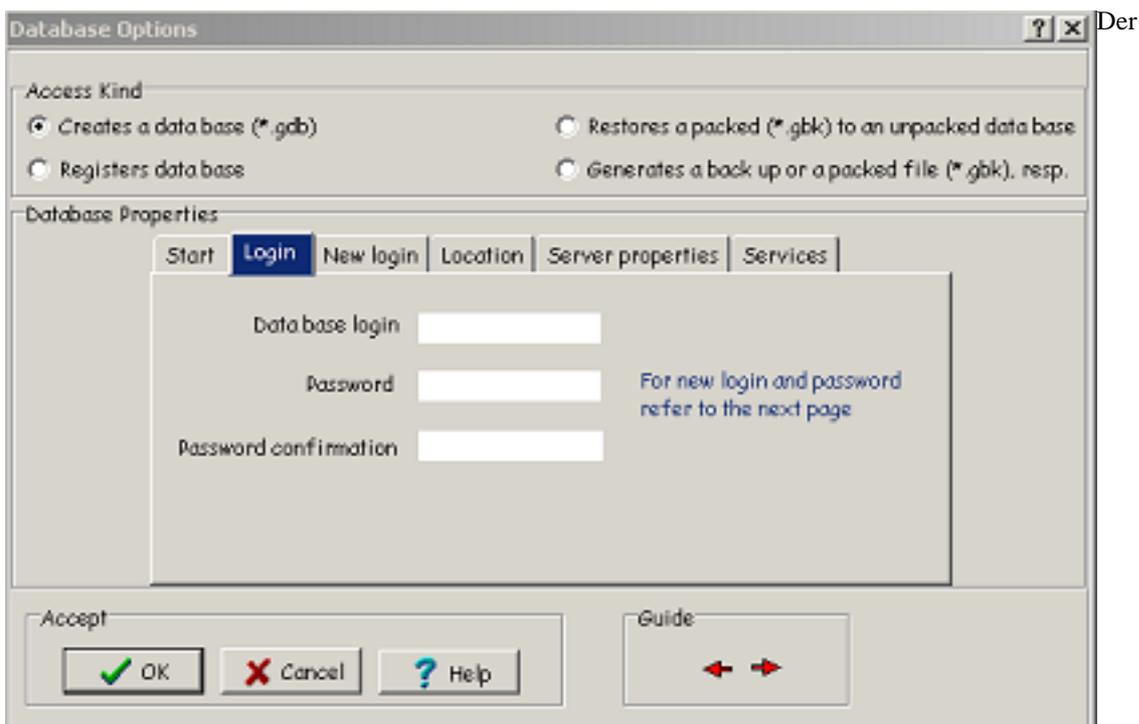


sehen im Bild das Start-Blatt markiert und dazu rechts unten im Bild - im Feld "Guide" - den Pfeil . Sie müssen diesen Pfeil benutzen, um die nötigen Datenbank-Informationen eingeben zu können. Nur mit Hilfe beider Pfeile navigieren Sie erfolgreich.

Sie können zwar alle Blätter durch Klick zur Ansicht bekommen, aber nicht zum editieren.

Alles zu Data Base Options [siehe Seite 206]

## A.73



Bezeichner Database Login ist Ihre persönliche Festlegung beim Anlegen eines neuen Projektes. Sie können auch ein vorhandenes Login verwenden. Dazu Öffnen Sie das Blatt "Server Properties". Ein

Klick auf  zeigt Ihnen alle bekannten Login-Bezeichner. Markieren und Kopieren<sup>36</sup> nach "Database Login" (im Blatt "Login") erledigt die Übernahme. Natürlich müssen Sie sich das "Password" vom Administrator der Datenbank geben lassen! Für Login "STANDARD" gilt "spas" als Passwort, für "Test" gilt "test".

Registrieren eines vorhandenen Projektes setzt die Kenntnis von Login und Passwort voraus.

Alles zu Data Base Options [siehe Seite 206]

## A.74



Undo.....Zuletzt ausgeführte Text-Aktion wird rückgängig gemacht

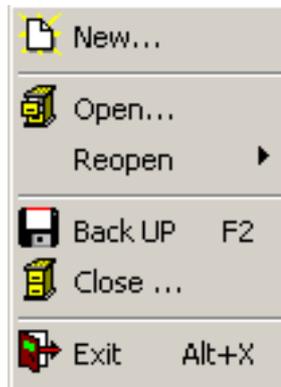
Cut.....markierten Text ausschneiden und zwischenspeichern

Copy.....markierten Text zwischenspeichern

Paste.....Einfügen eines Textes aus dem Zwischenspeicher

Delete.....markierten Text löschen

## A.75



New.....Anlegen eines neuen Projektes, einer neuen Projekt-Datei

Open.....Öffnen eines vorhandenen Projektes aus der Alias-Tabelle

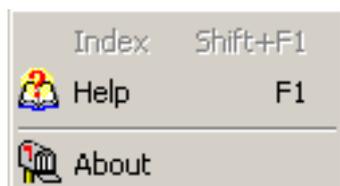
Reopen.....Liste der bisher bearbeiteten Projekte

Project Back Up...F2...Komprimierte Projektdatei als Rücklage mit .GBK-Erweiterung

Close.....Das Projekt wird geschlossen

Exit.....SPAS wird geschlossen

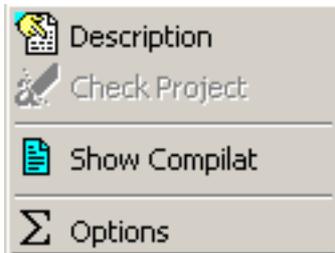
## A.76



Sprung zum Anfang von SPAS-Hilfe [siehe Seite 3]

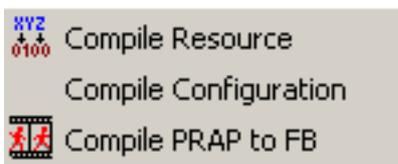
<sup>36</sup> Nach dem Markieren z.B. mit Strg+C in die Zwischenablage bringen. Danach das Blatt "Login" durch Klick öffnen, Cursor in "Database Login" setzen und z.B. mit Str+V das Login eintragen.

## A.77



Description  
Check Project

## A.78



Compile Resource.....Die AWL / Das Compilat für die markierte

Ressource wird hergestellt

Compile Configuration.....Das Compilat für alle zur Konfiguration gehörenden Ressourcen wird hergestellt

Compile PRAP to FB....Der Function Block zum markierten PRAP wird compiliert

## A.79

Zeigt alle in der Datenbank eingetragenen Informationen zu:



Configuration.....alle Konfigurationen des Projektes

Resources.....alle Ressourcen

Members.....alle Bausteine

Variables.....(mehr zur internen Information)

All Variables.....Nach Klick auf  sind alle Variablen des Projektes angezeigt und editierbar.

## A.80 Bemerkungen zum Handbuch zu SPAS

Die sogenannte kontextsensitive Hilfe erhalten Sie, wenn Sie in einem Formblatt, bzw. einem Fenster, den Cursor in ein veränderbares Eintragsfeld (Editierfeld) plazieren und die Taste "F1" drücken, falls kein  vorhanden. Benutzen Sie auch den "Hilfe"-Button, um eine Hilfe-Übersicht zu erhalten. Von dort aus erreichen Sie auch die kontextsensitiven Texte. Genaueres finden Sie hier [siehe Seite 155].

Am unteren Rand des SPAS-Manager-Bildes die Hinweis-Texte beachten.

## A.81

Falls Sie ein neues Login verwenden, müssen Sie das Feld "New Login" markieren. Anderenfalls übergehen Sie einfach dieses Blatt. Im Zweifelsfall gehen Sie vor zu "Server properties", um zu erfahren, ob das login neu ist!

Legen Sie aber ein neues Login an, dann empfiehlt es sich, den eigenen Namen (Vorname, Familienname) anzugeben, weil dann der Datenbank-Administrator erkennen kann, wer für das Login verantwortlich ist. Das Passwort vereinbaren Sie am besten mit ihm.

Das Erscheinen des "ADMINISTRATOR-REQUEST" beantworten Sie mit Klick auf OK.

Alles zu Data Base Options [siehe Seite 206]

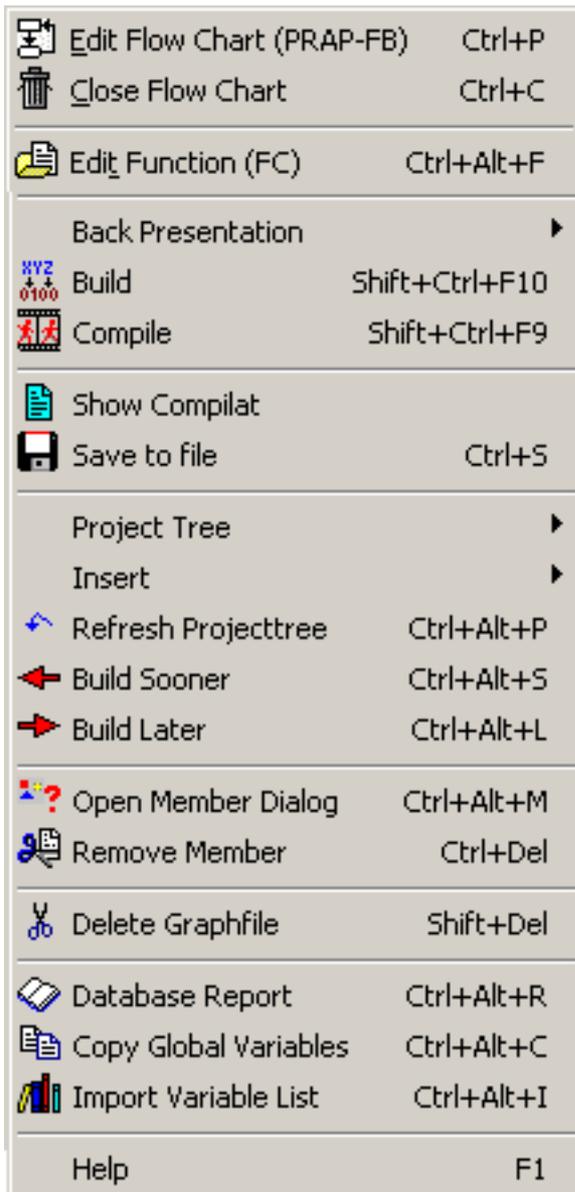
## A.82

Das **Projekt-Menue** wird bei einem markierten Member des Projektbaumes und Betätigen der rechten Maustaste dargestellt. Dabei sind nur die Texte aktiv, die zum Member passen.



### Tipp

Ziehen Sie das Help-Fenster so weit auseinander, bis die Texte rechts vom Bild auch zum Bild passen.



(Edit)-Öffnen des PRAP-Editors in "Charts".

PRAP-Member mit

(Close)-Schließen des PRAP-Editors mit Frage nach PRAP speichern

(Edit Function)-Öffnen des Funktions-Editors

(Back Presentation)-Alle PRAP einer Ressource online (menue siehe unten!) darstellen

(Build)-Compiliert den markierten Knoten vollständig

(Compile)-Compiliert nur geänderte Teile des markierten Knotens

(Show)-Anzeige des Compilates/Codes im Blatt "Code"

(Save)-Speichern des Compilates im "Repository"

(Project Tree)-Projektbaum-Menue siehe unten!

(Insert)-Einfügen-Menue öffnen (Hilfe zu jedem mit "F1")

(Refresh)-Neuaufbau des Projektbaumes aus der Datenbank

(sooner)-Reihenfolge des Members eher im Projektbaum (auch mit drag and drop)

(later)-Reihenfolge des Members später im Projektbaum (auch mit drag and drop)

(Open)-Öffnen des Member-Dialoges zu dem markierten Member

(Remove)-Member wird aus Datenbank entfernt (nicht rückgängig!)

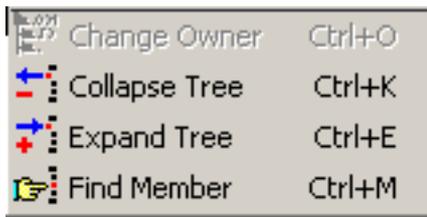
(Delete)-Graphikfile wird aus Datenbank entfernt (nicht rückgängig!)

(Data base)-Der Datenbank-Inhalt aller Variablen des markierten Members (druckfähig). Hilfe siehe DatabaseReport [siehe Seite 43]

(Copy)-Globalvariablen kopieren: Fragen Sie bei AMIKO GMBH!!

(Import)-Import einer Excel-Variablen-Tabelle

### Projektbaum-Menue



Änderung des +Projektnamens(noch nicht verfügbar)

+Projektbaum minimieren

+Projektbaum vollständig öffnen

+Suchen eines Members - Namen eingeben Back Presentation mit dem gilt für Build und Compile

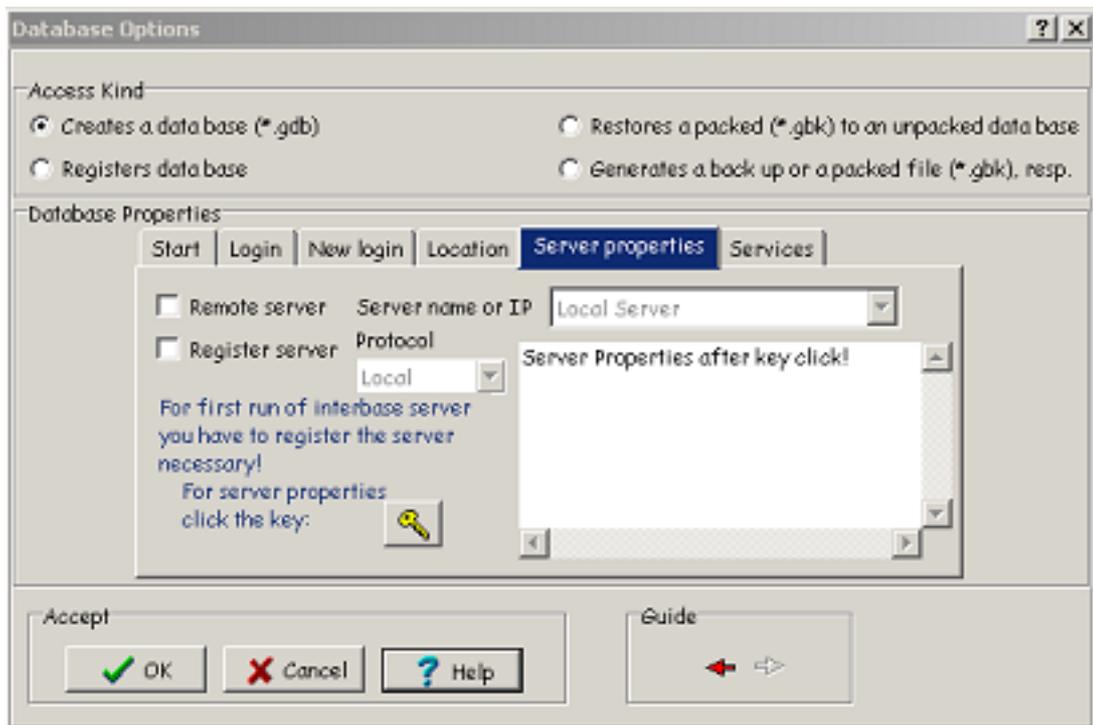


+(Prevent)-Vehindert die Erzeugung des Codes für die Darstellung der

"lebendigen PRAP"

+(Including)-Schließt die Code-Erzeugung ein

## A.83

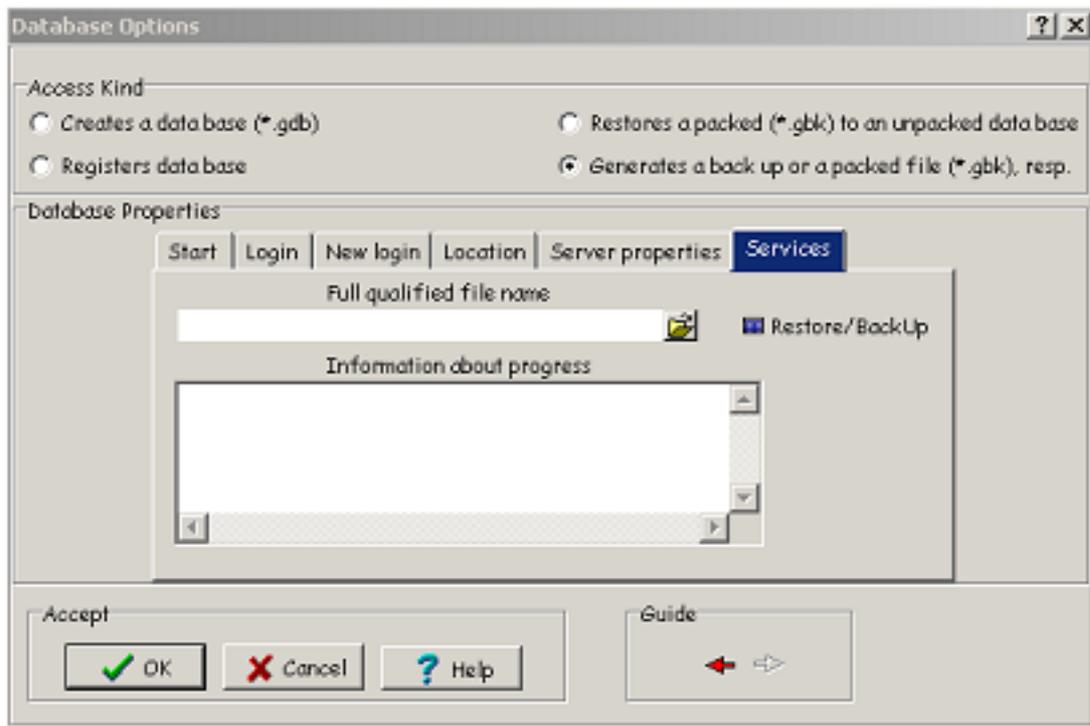


Markieren Sie "Remote Server", müssen Sie noch den Protokolltyp wählen und den Servernamen. "Local Server" ist zugleich Schlüsselwort und Aussage, daß der lokale Server eingeschaltet sein soll. Sollte nichts Passendes bei "Server Name or IP" eingetragen sein, dann markieren Sie bei "Register Server" (Anmeldung beim Betriebssystem) und tragen dann das Schlüsselwort/Kennwort ein.

Ein Klick auf  zeigt Ihnen alle bekannten Login-Bezeichner.

Alles zu Data Base Options [siehe Seite 206]

## A.84



Sie können von der Datenbank-Datei (\*.gdb) eine "Rücklagedatei" (\*.gbk) anfertigen lassen, ein back up, wodurch Ihnen eine gewisse Sicherheit gegeben wird. Diese Datei erhält denselben Namen wie die \*.gdb. Umgekehrt können Sie aus einem back up wieder eine ordentliche Projektdatei \*.gdb herstellen.

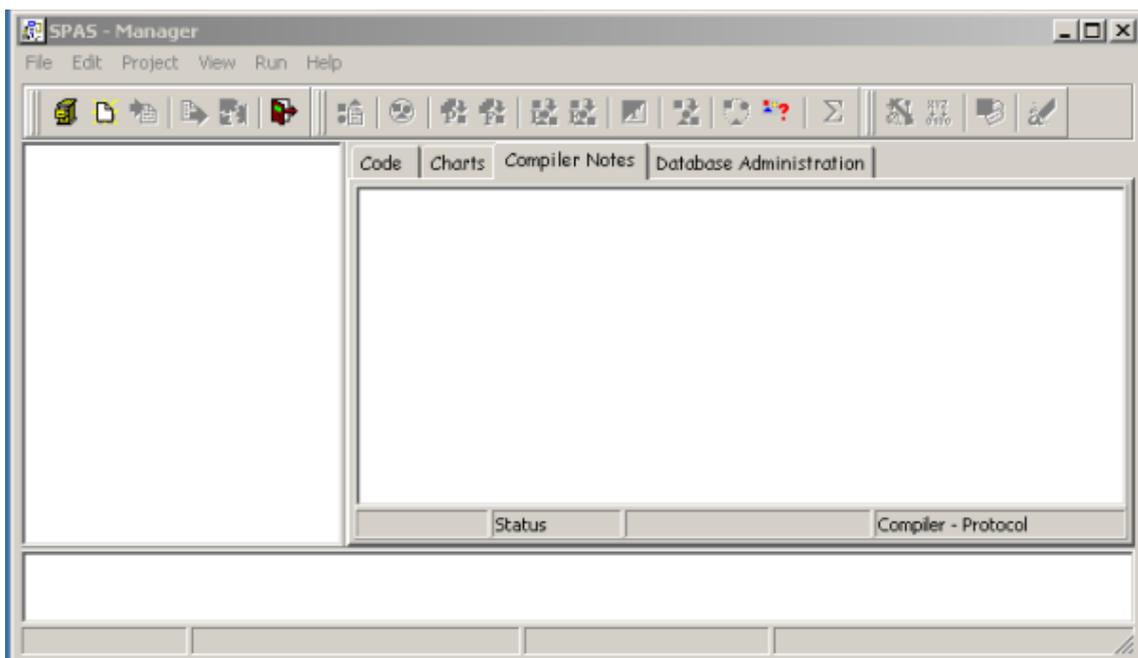
Der Start zur Ausführung der markierten Aktion erfolgt mit Klick auf  Restore/BackUp.

Das Feld "Information" enthält Eintragungen des Ablaufes der gewählten Aktion durch Interbase.

Der Abschluß mit Klick auf "OK" schließt das Formblatt.

Alles zu Data Base Options [siehe Seite 206]

## A.85 SPAS Manager



Das Bild zeigt das Formblatt unmittelbar nach dem Start von SPAS. Direkthilfe erhalten Sie durch Setzen des Cursors auf das betreffende Feld und Betätigen von "F1".

In der oberen Reihe stehen, beginnend mit File, [siehe Seite 159], danach Edit [siehe Seite 159] usw. PopUp Menüs zur Verfügung, die entsprechende Aktionen bewirken.

In der zweiten Reihe bieten Buttons verschiedene Aktionen, wobei Sie deren Bedeutung als Hinweise in der unteren Leiste lesen können. Auch Direkthilfe wird dort angeboten!.

Unter den Buttons befinden sich die Arbeitsfelder, das linke zum Aufbau und Bearbeiten des Projektbaumes. Es dient als Arbeitsfeld für den Projekt-Editor [siehe Seite 51]. Das rechte ist in vier Nutzungsbereiche eingeteilt:

- **CODE:**  
Nach dem Compilieren mit dem Precompiler können Sie das Ergebnis, das sog. Compilat ansehen. CODE [siehe Seite 142]
- **CHARTS:**  
Jeder PRAP wird auf diesem Panel entworfen oder korrigiert. Für diesen Zweck gibt es den PRAP-Editor. [siehe Seite 61]  
CHARTS [siehe Seite 139]
- **COMPILER NOTES:**  
Sowohl die Datenbank-Maschine als auch der Precompiler schreiben hier ihre Meldungen, die Sie vom Erfolg oder Mißerfolg einer Aktion unterrichten. Mit der rechten Maustaste erhalten Sie ein paar Handlungsmöglichkeiten.
- **DATABASE ADMINISTRATION:**  
Die Zugriffe auf die Datenbank erscheinen bei verschiedenen Aktionen, um notwendige Eintragungen bezüglich der verwendeten Variablen vorzunehmen oder Änderungen durchzuführen.  
Falls Sie in der oberen Leiste "VIEW" anklicken, erhalten Sie mehrere Angebote, sich Auszüge aus der Datenbank anzusehen. Nutzen Sie auch die Hilfeangebote.

Die beiden unteren Felder enthalten ein Informationsfeld, in dem die "Titel" der COMPILER NOTES eingetragen sind. Danach sehen Sie in der letzten Zeile vier Abteilungen. Von links nach rechts enthalten sie

- The Database is connected or disconnected - Datenbank geöffnet oder nicht
- The Hints of the buttons etc. - Bedeutung der Buttons usw.
- The Database Name - Bezeichnung der Datenbank
- The Login Name / The Database Login

## A.86 Deklaration VAR\_GLOBAL und VAR\_ACCESS in CONFIGURATION



### Anmerkung

Vorbemerkung zu SPAS Version 1:

Derzeit sind ACCESS\_VAR, also Größen zur Kommunikation über Bus nicht deklarierbar!

...später mehr...

## A.87

Die Tabelle der ungefilterten oder gefilterten Variablen mit Daten, die in die Datenbank eingetragen sind. Umordnen in alphabetischer Reihenfolge oder nach Datentyp nach RECHTE MAUSTASTE. Ebenso Informationen [siehe Seite 148] zur markierten Variablen.

## A.88

Nach Klick auf "Check Import" können Sie hier das Ergebnis der Importaktion sehen. Die Daten sind mit Komma getrennt. Gelesene Leerzeilen enthalten nur Kommata.

## A.89

Mit Klick auf , bitte Ihre Excel-Tabelle (name).xls auswählen!

## A.90 Inbetriebnahme des "Lebendigen PRAP"

Zunächst die Einführung genau befolgen!

Öffnen Sie einen Teilprozess durch Doppelklick auf den gewünschten im Projektbaum. Danach ein Klick auf die "Kamera", und Ihnen wird in Abständen, die wesentlich durch das Prodave-System und die Bedienung der MPI-Schnittstelle durch die SPS bestimmt sind, ein Datensatz übertragen. Damit ändern sich die Conditions immer mit der "Wirklichkeit". Der Fortschritt im Steuerungsablauf wird durch Farbumschlag der betreffenden Operation sichtbar.

Die aktuellen Werte der Variablen können Sie zu jeder Instanz ansehen, indem Sie mit der Maus auf die betreffende Instanz gehen und RECHTS klicken. Alle diese Werte sind ohne Adresse, weil eigentlich der PRAP - und das ist doch ein TYP! - angezeigt wird, nur eben mit den aktuellen Werten der Instanz.

Wollen Sie die Werte der adressierten Variablen betrachten, dann mit rechtem Mausklick auf das Programm!

## A.91

### ANZEIGEFELD

Sie erhalten hier vor allem das Ergebnis der Ermittlung der Variablenzugehörigkeit, die Sie auslösen können, wenn Sie oben in der Tabelle eine Variable markieren, die rechte Maustaste betätigen und "Search Variable Usage" anklicken.

Sie erhalten drei Informationen: Zugehörigkeit zum Member, Usage (Verwendung der Variablen) und die Formel, in der die Variable eingebunden ist.

Bitte führen Sie den Cursor an die obere Feldbegrenzung, bis Sie das Erfassungszeichen (zwei parallele Linien) sehen, linke Maustaste drücken und Feld nach oben vergrößern, um alle Mitteilungen lesen zu können.

Sie können alle Anzeigen löschen, indem Sie im Feld mit der rechten Maustaste den Löschvorgang einleiten und mit Klick ausführen.

## A.92

Initial-Operation oder Start-Operation (= Launch Operation) nennt man diejenige Operation in einem PRAP, die mit dem Einschalten/Starten (Stop => Run) der Steuer-Einrichtung als erste ausgeführt werden muß. Das gilt auch dann, wenn mit dem Start schon ein Übergang zu einer Folgeoperation erfüllt sein sollte, der erst im nächsten Verarbeitungs-Zyklus zur Ausführung gelangt. In der Initialoperation können daher z.B. Speicherbereiche mit Anfangswerten belegt werden (siehe Kopierfunktion).

Zur Initial-Situation gehört neben der Initialoperation noch der Stabilitäts-Prozeßzustand.

## A.93

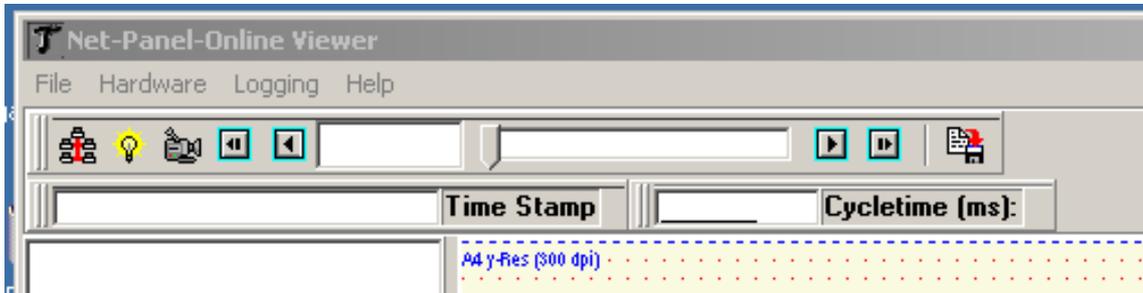
Mit Klick auf diesen Button und anschließendem Doppelklick auf die neue Eintragung öffnet sich das Fenster zum Eintragen einer neuen Operationsvariablen. Weitere Hilfe siehe dort.

Wollen Sie eine Variable löschen, dann markieren Sie diese Variable und mit rechtem Mausklick wird das Löschen angeboten.

## A.94

Gibt es in einem Operationen-Zyklus keine Möglichkeit, daß eine Operation stabil auftritt, dann liegt ein instabiler Zyklus vor. Es sind in diesem Fall im PRAP alle Übergangszustände dieses Zyklus' zugleich erfüllt.

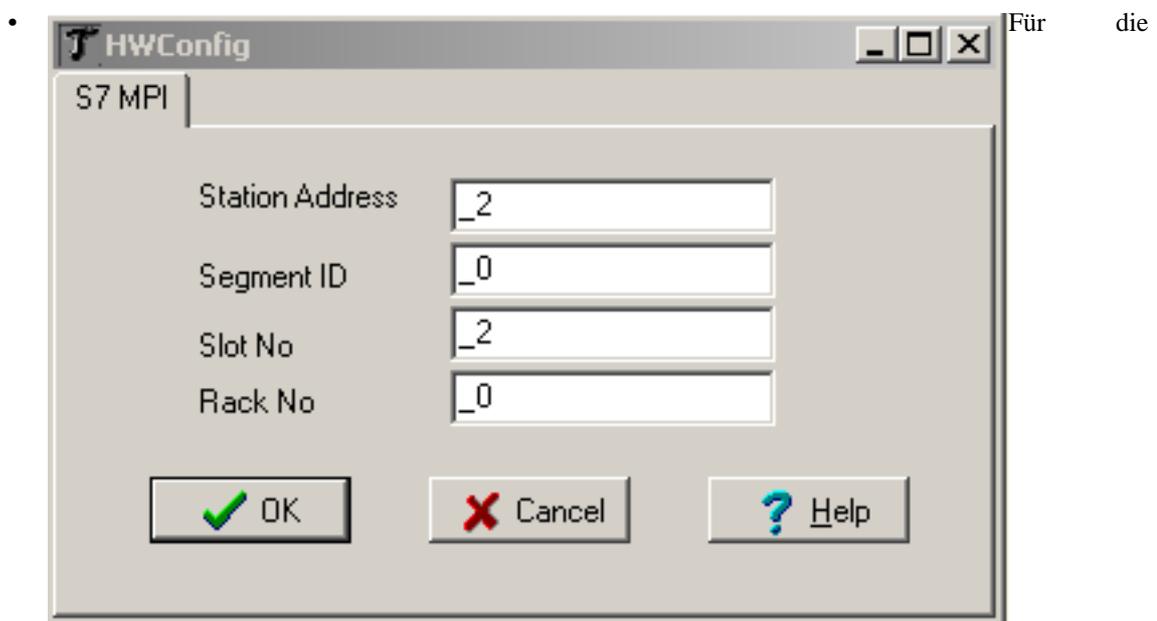
## A.95 Einführung - Introduction



Mit dem Öffnen der Datei "Online.exe" ergibt sich obiges Bild, die Taskleiste. Hier können Sie die drei Anzeigeleisten frei positionieren (linke Maustaste an der linken Seite anfassen und verschieben).

Gehen Sie nun wie folgt vor:

- Ein Klick auf "File" bietet das Öffnen (open) der für Ihr Projekt zuständigen Datei [Dateiname].onl an. Diese Datei erzeugt SPAS dann, wenn Sie nach dem Compilieren - "Build" - "Save to File" angeklickt haben. Neben der Quellcode-Datei der Zielsprache wie z.B. Step 7 finden Sie [Dateiname].onl und [Dateiname].sdf, die Symboldatei .  
Der erste Schritt genügt für einen statischen Test. [siehe Seite 190]



Datenübertragung von der Steuereinheit(Ressource, z.B. SPS) klicken Sie nun auf "Hardware", um die Netzdaten eingeben zu können. Im Beispiel handelt es sich um eine S7-315 mit ihren Daten [siehe Seite 173].

Der lebendige PRAP [siehe Seite 166] steht Ihnen zur Verfügung.

- Ein Klick auf "Logging" bietet Ihnen das Entfernen von abgespeicherten Schritten an, die Sie per Hand (Offline, statisch [siehe Seite 190]) oder dynamisch (Online [siehe Seite 166]) haben aufzeichnen lassen. Den ersten und den letzten Index finden Sie in der Taskleiste, rechts:



Die im Beispiel zu sehende Zahl "4" ist der letzte Index der Aufzeichnung. Die vorhergehenden

finden Sie durch Klick auf den linken Pfeil-Button.

## A.96

Eine kombinatorische Struktur entsteht, wenn der Übergangszustand  $P_{jk}$  der Operation  $O_j$  gleich dem Haltezustand  $P_{kk}$  der Operation  $O_k$  ist und umgekehrt  $P_{ki} = P_{ii}$  gilt. Erfüllt der Übergangszustand  $P_{if}$  zu einer gemeinsamen Folgeoperation  $O_f$  die Gleichheitsbedingung nicht, dann gehört  $O_f$  nicht zur kombinatorischen Struktur, aber es muß  $P_{if} = P_{kf}$  für alle Operationen der kombinatorischen Struktur gelten.

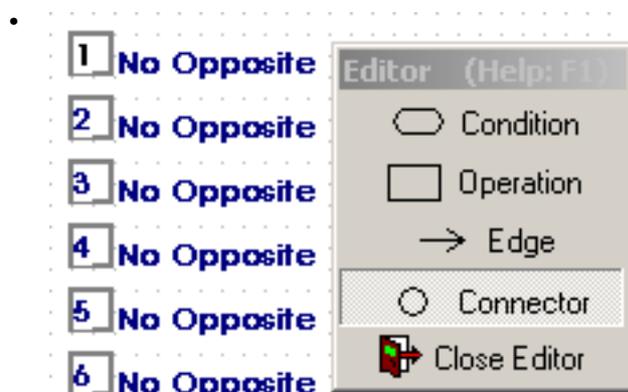
## A.97

### Konnektoren

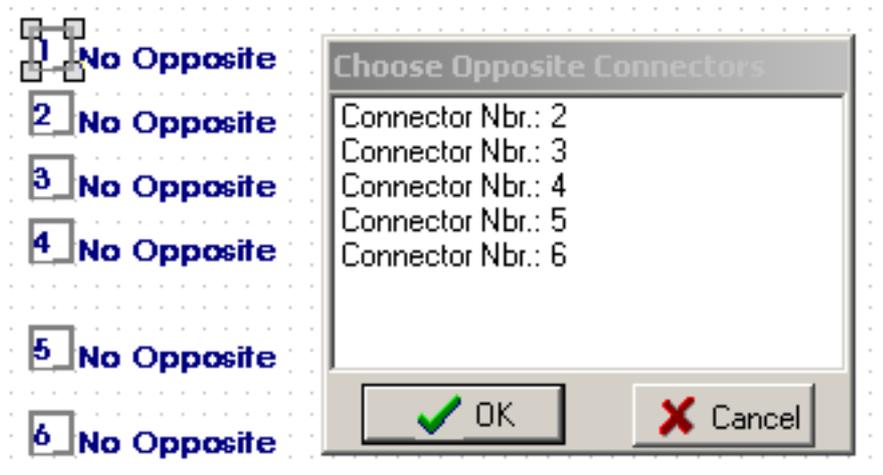
Konnektoren dienen zum Verbinden von PRAP-Elementen an Stelle von gerichteten Linien und helfen, den PRAP von Linien-Netzen zu befreien.

- Platzieren Sie mindestens zwei Konnektoren.
- Einer von beiden wird genau eine Kante erhalten, die zu einem Objekt (Operation oder Condition) hinführt, also vom Konnektor weggeführt. Das ist der "wegführende Konnektor".
- Der andere Konnektor erhält wenigstens eine Kante, die von einem Objekt weggeführt, also auf den Konnektor hinführt. Das ist der "hinführende Konnektor". Jedes weitere Objekt führt "seine Kante" hin zum Konnektor. Es dürfen beliebig viele Kanten zum Konnektor führen.
- Der wegführende Konnektor kann mit vielen hinführenden verbunden werden, nicht umgekehrt! Man kann auch die Vorstellung von Quelle und Ziel anwenden. Der wegführende ist das Ziel von wenigstens einer Quelle.
- Konnektor und Kommentar können durch Markieren des Konnektors und festhalten der Maustaste verschoben werden. Der Kommentar ist allein verschiebbar.

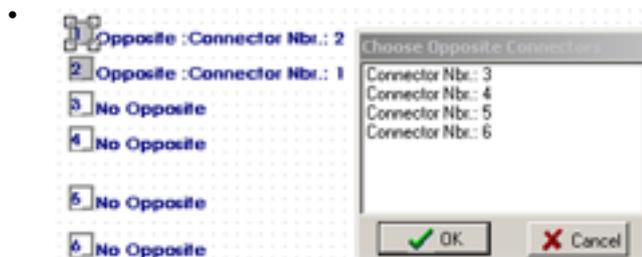
Beispiel:



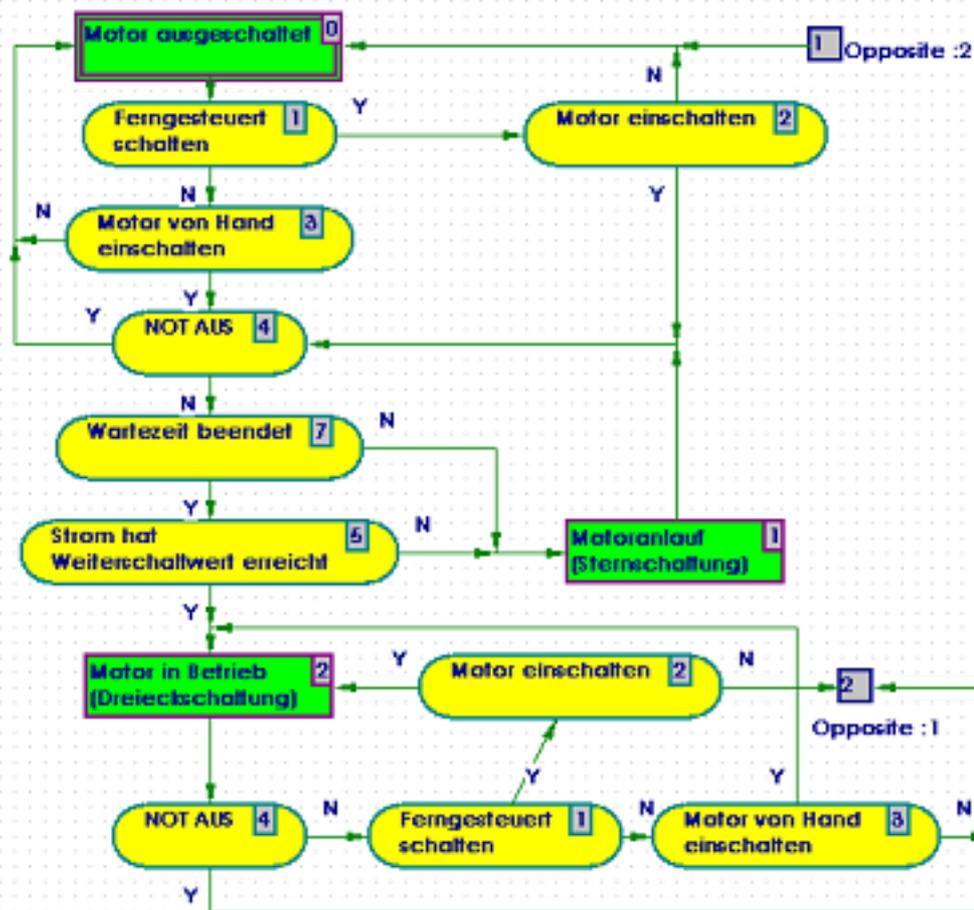
Im Bild sehen Sie sechs Konnektoren, die nicht miteinander verbunden sind, weshalb sie "No Opposit" (kein Gegenüber, keinen Partner) als Kommentar haben.



Der Konnektor 1 soll ein wegführender sein: Markieren und rechte Maustaste betätigen, danach: Klick auf "Connect To Connector" öffnet dieses Auswahl-Angebot. Markieren des Partners und OK zeigt die gewählte Verbindung an. Soll Konnektor 1 mit mehr Konnektoren verbunden werden, dann diese Prozedur wiederholen.



Das Bild zeigt das Ergebnis einer Verbindung von Konnektor 1 mit Konnektor 2, und Sie können zu Konnektor 1 weitere Verbindungen herstellen, falls er ein hinführender ist. Deshalb müssen Sie den hinführenden Konnektor zuerst mit einer Kante mit der PRAP-Graphik verbinden.



Aus dem Lehrbeispiel sehen Sie den Stern-Dreieck-Motor-Anlauf<sup>37</sup> mit einem Konnektorpaar. Auf den hinführenden Konnektor 2 gehen drei Kanten. Sie kommen von den Conditions 2, 3 und 4.



### Tipp

Es ist oft einfacher oder leichter "lesbar", noch einen wegführenden Konnektor zum gleichen Ziel, z.B. zu einer Operation, dann einzuführen, wenn im PRAP "weiter entfernt liegende" Conditions ein Anbinden an nur einen Konnektor zur Unübersichtlichkeit beiträgt. Dies ersetzt das Verbinden mehrerer Konnektoren miteinander.

## A.98

Koppelgrößen oder Koppelvariable sind natürliche oder erzeugte(=eingeführte) Kopplungen zwischen den Teilprozessen.

- Natürliche Koppelungen ergeben sich durch die Nutzung einer Prozeßvariablen/Prozeß-Zustandsvariablen in mehreren Teilprozessen. Diese P-Variable muß bezüglich EMSR- Stelle bzw. Betriebsmittelkennzeichen demjenigen TP zugeordnet werden, wo sie durch eine Operationsvariable über den Wirkungskreis des zu steuernden Prozesses beeinflußt wird. Beispiel: "Licht einschalten" ergibt Aussage über "Helligkeit erreicht?"  
Das Prozeß-Ablauf-Netz (PAN) liefert von den Transitionen, an denen Verzweigungen stattfinden, natürliche Koppelungen an die aus dem PAN abgeleitete PRAP.
- Erzeugte, eingeführte Koppelungen ergeben sich, wenn keine natürliche Koppelung möglich ist. Dann ist als Operationsvariable eine Koppelvariable zu "erzeugen, einzuführen", die für alle TP, die diese



Wollen Sie eine Variable löschen, dann markieren Sie diese Variable und mit rechtem Mausklick wird das Löschen angeboten.

## A.100

Zugelassen sind maximal 8 Zeichen aus A .. Z und 0 .. 9, das erste muß ein Buchstabe sein.

## A.101

### Mensch-Anlagen-Kommunikation (MAK) oder Mensch-Prozeß-Kommunikation

bringt zum Ausdruck, daß der Bediener einer Automatisierungseinrichtung (AE) durch vereinbarte (in der Bedienungsanleitung!), vom Projektanten der AE vorgesehene Signale zur Tätigkeit entsprechend der Bedeutung des Signals aufgefordert wird. Besteht diese Tätigkeit in einer Bedienhandlung, so müssen Bedieneinrichtungen vorhanden sein, die eben zu dieser Zweckerfüllung geeignet sind.

Das Informationsflußmodell zeigt die Einbettung in das Modell des zu steuernden Prozesses.

## A.102

Alle Knoten im Projektbaum haben die Eigenschaft eines "Mitgliedes" des Projektes (= Member) zu sein. Sie werden von oben nach unten durchnummeriert, was den Nutzer von SPAS eigentlich nicht weiter interessiert.

Jedes Member erhält eine Identifikation, also eine Zeichenfolge zur "Erkennung" im Tool SPAS, und - nicht alle - eine Beschreibung (Description), ein kurzer Text, also keine Erläuterung, Erklärung, sondern ein technologisch relevantes Wort oder eine Wortgruppe wie "Vorderer Haupteingang" ist damit gemeint, das kann eine Tür oder ein Tor sein. Dies sind z.B. Typenbezeichnungen für die tatsächliche Einrichtung (vorderer Haupteingang), das sogenannte Exemplar (eines Tores), die sogenannte Instanz des Typs.

Zu jedem PRAP-Knoten gehört eine Identifikation, ein sogenannter Bezeichner, der eine Kurzform der Description sein sollte.

## A.103

### **Alle Mitteilungen der Editoren, der Datenbank und des Precompilers**

Beachten Sie vor allem Warnungen und Fehlermeldungen, die eine Fertigstellung eines Berechnungsschrittes nicht erlauben.

Zum Beispiel: Solange der PRAP-Editor noch Fehler meldet, startet der Precompiler ebenfalls mit Fehleranzeige.

Klick auf rechte Maustaste ermöglicht Löschen aller Mitteilungen oder eine Erläuterung zur Fehlermeldung zu erhalten.

## A.104

Als Zeichen sind nur "A ... Z", "a ... z", "0 ... 9" und "\_" erlaubt. Die Anzahl ist mit 64 begrenzt und kann sprachabhängig kleiner sein. Eine Wort-Trennung (Leerzeichen) ist nicht zugelassen. Das erste Zeichen muß ein Buchstabe oder höchstens ein Unterstrich sein.

Da "NAME" ein Identifikator ist, sind gleiche NAME auch gleich geschrieben, also mit gleicher Zeichenfolge. Daher sind Groß- und Kleinbuchstaben zu beachten!

## A.105

Technologisch begründeter Text, dem "Name" zugeordnet ist. Hier änderbar bzw. bei neuen Variablen einzutragen.

## A.106

Hier wird die Variable (deren "Name") angezeigt, auf der der Cursor im Feld "Formula" steht. Editierbar

nur nach Klick auf .



### Anmerkung

Momentanwerte der Zeit- und Zählglieder sind mit "Name".ET (Datentyp TIME) bzw. "Name".CV (Datentyp INT) **in den Formelausdruck** einzutragen.

## A.107

### Netzdaten bei MPI

- "Station Address" ist die der MPI-Verbindung an der S7.
- "Segment ID" kann 0 bleiben, wird erst mit Profibus interessant.
- "Slot No" ist die Steckplatz-Nummer der S7-CPU im Einschub.
- "Rack No" ist die Nummer des Einschubes bzw. der Profilschiene.

Falls die Daten in Ordnung sind, gibt es keine Rückmeldung. Anderenfalls zeigt eine Fehlermeldung an, welche Probleme es gibt.

## A.108



Delete Object...Löschen des markierten Graphik-Symbols

Connect to/ Disconnect from Connector...Betrifft die Kante vom Graphik-Symbol zum Konnektor

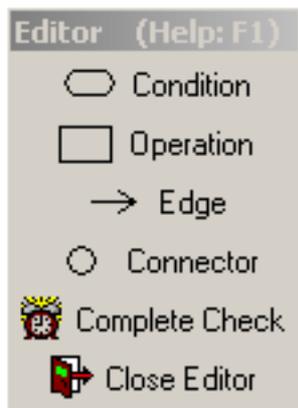
Data Access.....Zugriff auf den Datensatz des markierten Graphik-Symbols

Route Information ... (noch nicht programmiert)

Set Launch Operation....Die markierte Operation wird als Start- bzw. Initialoperation gekennzeichnet (Doppelrahmen)

## A.109

Durch Klick auf eine Menü-Zeile ändert der Cursor seine Form. Es können folgende PRAP-Objekte



Condition = Bedingung, Prozeß-Variable

Operation = Operation, Wertebelegung aller Operationsvariablen

Edge = Kante, gerichtet (Pfeil-Linie), zwischen zwei Objekten

Connector = Konnektor dient zur Unterbrechung von Kanten und enthält Zählnummern mit Angabe des Partners

Complete Check

Vollständige Prüfung aller PRAP-Eigenschaften. Keine Prüfung von Formeln aller Art (eingebettete Funktionen wie TIMER, Vergleiche usw.)



### Warnung

Unterbrechung beim Kanten-Ziehen: Lesen Sie nach Klick bei 'Problemen...' nach [siehe Seite 217]

Ein Klick auf die markierte Menü-Zeile schaltet den Editor ab. Damit können Sie z.B. Kanten bearbeiten.

Löschen eines Objektes: Hinführen des Cursors, dann rechte Maustaste und Klick "Delete Object".

Weitere Hilfetexte:

Wie behandelt man Operationen (am Beispiel) [siehe Seite 106]

Wie behandelt man Prozeßvariable (am Beispiel) [siehe Seite 110]

Wie zeichnet man Kanten [siehe Seite 217]

Wie nutzt man Konnektoren [siehe Seite 168]

## A.110

STEP 7 (S7-300 und S7-400) wird mit Organisationsbausteinen erzeugt. Geben Sie hier nur die OB-Nummer ein oder lassen Sie das Feld offen; denn wenn die Felder PRIORITY und SINGLE/INTERVAL passend ausgefüllt sind, fügt SPAS die richtige Zahl dann ein.

## A.111

Eine Operation enthält alle zu einem Teilprozeß gehörenden, binär bewerteten Operationsvariable (kurz: OVar, auch Steuervariable genannt). Zu jeder Operation gehört eine zu ihr gehörende Wertebelegung. Jede Operation ist daher ein Binärvektor.

Sie sollten auch einen Text in eine Operation schreiben (=Comment), um dem Leser des PRAP eine Art Schrittbezeichnung anzubieten, die Auskunft über den technologisch wesentlichen Sinn der Operation gibt.

## A.112

Mit Klick auf diesen Button übertragen Sie den Inhalt des Formblattes "Operation" in die Datenbank, so daß der Inhalt der Liste der Operationsvariablen der Operation bei "Comment" fest zugeordnet wird.

Wollen Sie eine Variable aus der Liste löschen, dann markieren Sie diese Variable, und mit rechtem Mausklick wird das Löschen angeboten.

## A.113 Operation Variables

In diesem Formular bestimmen Sie alle Eigenschaften, die mit der ausgewählten Operations-Variablen in Zusammenhang stehen. Folgende Parameter stehen zur Verfügung:

- Statement - Variablenbezeichner [siehe Seite 176]
- Point of Automation - EMSR-Stelle [siehe Seite 176]
- Variable Name (Flag) [siehe Seite 176]
- Generated [siehe Seite 175]
- Address [siehe Seite 175]
- Variable Usage [siehe Seite 195]
- Choose a Function [siehe Seite 140]
- Formula [siehe Seite 138]

## A.114 Operation

Jede Operation [siehe Seite 174] gibt die binäre Belegung der Operationsvariablen in der "List of Operation Variables" an, die zum PRAP gehören.

Folgende Hinweise können Sie erhalten:

- Zu Statement [siehe Seite 144]
- Zu Liste der Operationsvariablen [siehe Seite 171]
- Zu Edit Selected Variable [siehe Seite 150]
- Zu Insert New Variable [siehe Seite 166]
- Zu Return [siehe Seite 174]

## A.115

Unter einer Operationenfolge versteht man Operationen, die paarweise über wenigstens einen Pfad (vergleiche auch Übergangsprozeßzustand) verbunden sind, also beginnend bei einer Start-Operation ein Pfad zu deren Folge-Operation führt, und diese als neue Start-Operation eine weitere Folge-Operation besitzt usw.

Ist die letzte Folge-Operation im speziellen Fall zugleich die erste Start-Operation, dann ist die durchlaufene Operationenfolge ein Operationen-Zyklus, oder kurz ein Zyklus.

## A.116

Die Variablen-Adresse ist hier nicht eintragbar, mit Klick auf  öffnet sich das zugehörige Formblatt.

## A.117

Die Eigenschaft "Generated" ist hier nicht eintragbar, mit Klick auf  öffnet sich das zugehörige Formblatt.

## A.118

Tragen Sie - wenn überhaupt - hier nur Funktionen-Hinweise, keine konkreten EMSR-Stellen ein. Z.B. bei einem allgemeinen Antrieb UV oder elektrisch EU, oder HO für optische Signalisierung usw.

Konkrete EMSR-Stellen müssen bei den Aktualparametern bzw. direkt adressierten Variablen der Teilprozesse - Instanzen der PRAP-Typen - eingetragen werden, und bei Unikaten oder in Typen mit beschränkter Anzahl der Formalparameter.

## A.119

Der Variablen-Name ist hier nicht eintragbar, mit Klick auf  öffnet sich das zugehörige Formblatt. (Flag) deutet auf die Eigenschaft der hier einzutragenden Booleschen Variablen hin. Sie kann, sofern sie mit einer Funktion (siehe Choose a Function) verbunden ist, diese einschalten und ausschalten. (siehe "Nichtbinäre Funktionen im PRAP").

## A.120

Der Variablenbezeichner ist hier eintragbar oder änderbar und wird mit Klick auf "OK" in die Liste der Operationsvariablen dann übernommen, wenn auch ein VARIABLE NAME (Flag) vorhanden. Anderenfalls wird der Bezeichner nur ohne Bezug zwischengespeichert. Mit Klick auf "Cancel" bleibt der vorhandene Text.

## A.121

Unter einem Operationen-Zyklus versteht man in einem PRAP eine in sich geschlossene Folge von Operationen, d.h. jede beliebige Anfangsoperation ist auch Zieloperation in einer Operationenfolge.

## A.122 Information und Beschreibung zur Funktion des PRAP

### Allgemeines:

Die Felder sind alle editierbar. Allerdings erfolgt der Eintrag des Instanz-Namens, also des Teilprozeß-Namens, automatisch immer in die Zeile "Project" am Anfang. Die ersten vier Felder sind die sog. Auftragsdaten.

### Bemerkungen zu den Feldern:

Das Feld "Project" beginnt mit dem Namen (Identifikation)des Teilprozesses, der Instanz. Danach sollte der Projektname stehen, den Sie selbst eintragen.

Im Feld "Editor" wird der Name des Entwerfers, des Projektanten, erwartet, der den PRAP entwickelt hat. "Order" bedeutet soviel wie Auftraggeber, schreiben Sie also nach Bedarf etwas Sinnvolles ein, weil der Auftraggeber für die Richtigkeit des PRAP mit haften sollte.

"Last Access" ist ein automatischer Eintrag.

"Description" sollte die wichtigsten Besonderheiten des PRAP enthalten. Dazu gehören Hinweise auf eingebettete Funktionen, Erläuterungen aller Besonderheiten, die einem bestimmten Zweck dienen. Auch die Lösung einer Aufgabenbeschreibung kann für den späteren Leser sehr hilfreich sein.

### Handhabungshinweis:

Die sog. Auftragsdaten müssen Sie wenigstens einmal ausgefüllt haben. Wenn Sie nun einen anderen PRAP beschreiben wollen, dann werden diese Eintragungen genau von dem vorher geöffneten übernommen. Sie können natürlich Veränderungen vornehmen, falls erforderlich.

"OK"-Klick speichert die Eintragungen, "Cancel"-Klick übernimmt keine Veränderungen.

## A.123

Operationsvariable sind binär zu wertende Größen. Sie können sein:

- Stellgrößen für alle prozeßtechnologischen Eingriffe:  
Ein- und Ausschalten. Öffnen und Schließen.  
Einschalten und nicht Einschalten. Ausschalten und nicht Ausschalten.  
Öffnen und nicht Öffnen. Schließen und nicht Schließen.
- Informationen an den Operateur im Rahmen der MAK
- Ansteuersignale für Zeit- und Zählglieder, Programmgeber, Schieberegister, usw.
- Erzeugte Koppelvariable zur funktionellen Beeinflussung anderer Teilprozesse.

## A.124

Hier tragen Sie den Namen der formalen O-Variablen ein, eine Kurzfassung zum "Statement".  
Beachten Sie dabei den Hinweis  
sowie den Text zu "ADDRESS".

## A.125

Bitte wählen Sie aus dem Angebot die zutreffende Eigenschaft aus.

## A.126

Paarweise Steuerrelationen sind Koppelbeziehungen zwischen je zwei Teilprozessen, die durch Koppelvariable hergestellt werden.

## A.127

Parameter sind alle Größen, die der Verarbeitung in einer Steuerungs- oder Regelungsfunktion zugeführt werden. Unterschieden wird zwischen Formalparametern und Aktualparametern, Eingangs- und Ausgangsparametern.

Formalparameter stehen in einem Typ-Teilprozeß (also auch in einem Typen-PRAP), Aktualparameter einer Instanz eines Typen, werden den Formalparametern des Typen zugewiesen.

## A.128

Tragen Sie dieselben Bezeichner (Zeichenketten) ein, wie sie beim Anlegen des Projektes verwendet wurden.

## A.129

Zugelassen sind mehr als 8 Zeichen aus A .. Z, a .. z und 0 ..9.

Die ersten acht Zeichen sind einschließlich der Groß- und Kleinschreibung signifikant.

## A.130

Unter einem Pfad = path versteht man die Verbindung zwischen einer Operation Oj und einer Folgeoperation Ok durch eine Prozeßvariable allein oder mehrere aufeinanderfolgende, jede sollte nur einmal in einem Pfad auftreten.

## A.131

Programm-Organisations-Einheit (Program Organization Unit)

Abgekürzt POE ist laut DIN IEC 61131-3 eine Funktion (FUNCTION), ein Funktionsbaustein

(FUNCTION BLOCK) oder ein Programm (PROGRAM) sowohl als Type als auch als Instanz.

## A.132

Bitte Speicherort für exportierten PRAP-Typ angeben.

## A.133

PreCOM ist die Abkürzung von Precompiler, Vorübersetzer. Damit wird aus einem Steuerungsprojekt ein Steuerungsprogramm erzeugt, allerdings im Quellcode, der mittels zugehöriger Software zur Zielmaschine (Ressource wie SPS, Microcontroller usw.) in den Maschinencode zu übersetzen und zu implementieren ist.

## A.134

### Primary Variable | Name

Die "Eingangsvariable", deren Flanke(n) ausgewertet werden soll(en), muß vom Datentyp BOOL sein. Ihre Verwendung (USAGE) kann ein Input, Output oder Input/Output sein, d.h. es handelt sich um eine EMSR-Stelle. Sie kann aber auch interner Merker sein, also mit Local Static oder Local Temporary deklarierbar.

Sie können eine vorhandene Variable auswählen oder eine neue eintragen. Beachten Sie die richtige Bedienung der Symbole  und .

## A.135 Process Variables

In diesem Formular bestimmen Sie alle Eigenschaften, die mit der ausgewählten Prozeß-Variablen in Zusammenhang stehen. Folgende Parameter stehen zur Verfügung:

- Statement - Variablenbezeichner [siehe Seite 185]
- Point of Automation - EMSR-Stelle [siehe Seite 184]
- Variable Name (Flag) [siehe Seite 184]
- Generated [siehe Seite 184]
- Address [siehe Seite 184]
- Variable Usage [siehe Seite 195]
- Choose a Function [siehe Seite 141]
- Formula [siehe Seite 138]

## A.136

### Description

Die Beschreibung darf ausführlicher sein, muß jedoch im Rahmen der Beschränkungen bleiben.

## A.137

### Identification

Tragen Sie hier eine Kurzbezeichnung ein, die bestimmten Bedingungen genügen soll.

## A.138

## RELATIONS

Es ist auf jeden Fall der Programm-Typ zu wählen. Danach erfolgt die Zuweisung einer Task, die bei der "zuständigen" Ressource deklariert wurde.

### A.139

Bezeichnen Sie hier den technologischen Abschnitt, der durch die Instanzen gesteuert wird. Beachten Sie dabei

### A.140

Setzen Sie hier eine Kurzbezeichnung ein, die in der Ressource einmalig ist.

### A.141

Die globalen Variablen (PARAMETER) sind lokal in PROGRAM deklariert und dienen der Instanzierung der formalparameter und der Kommunikation zwischen den Bausteinen, die zu PROGRAM "gehören". Externe Variable liefern die Ressource und die Konfiguration, also dort deklarierte globale Variable.

Falls Sie noch keine Ihrem Programm zugeordnet haben oder Veränderungen wünschen, dann klicken Sie auf den HELP-Button. Dort ist unter PARAMETER das Vorgehen beschrieben.

## A.142 Project Member Actual Function, Call Function

### Grundsätzliches:

Der Funktionsaufruf ist ein eigenständiger Vorgang insofern, als er in keinem Prozeß-Zerlegungsgraphen, in keiner Automatisierungs-Projekt-Dokumentation enthalten ist. Die Funktion mit ihren aktuellen Variablen ist ein Hilfsmittel des Projektanten zur Gestaltung des Automatisierungs-Programms.

### Die Ausführung:

- Markieren Sie das "PROGRAM", in das Sie einen Funktionsaufruf eingliedern wollen.
- Klicken Sie auf 
- Sie können keinen neuen Namen eingeben, weil es sich um einen Aufruf der deklarierten Funktion handelt. Daher bei "FUNCTIONS" die zutreffende auswählen.
- Sie sehen rot unterlegt alle Formalparameter, denen Sie Aktualparameter zuweisen können, aber jetzt noch nicht dürfen. Geben Sie aber noch eine DESCRIPTION ein und ZUM SCHLUSS OK klicken.

Damit ist der Funktionsaufruf im PROGRAM angelegt.

Nun folgt mit neuerlichem Öffnen des Memberdialoges die Zuordnung der Formalparameter zu den Aktualparametern:

Sie erhalten Zugriff durch Klick auf eine rot markierte Zeile oder auf  und wählen aus dem im Programm [siehe Seite 213] oder der Ressource [siehe Seite 181] deklarierten Aktualparametern.



### Anmerkung

Achten Sie auf die Zugehörigkeit der Variablen (Anzeige im Feld "Member"), weil in verschiedenen Members gleiche Namen zugelassen sind. Die Zugehörigkeit muß immer das gleiche PROGRAM bzw. die RESSOURCE sein.

## A.143 Project Member - Configuration

- Allgemeines  
Das einzige Feld, in das ein Eintrag erfolgen kann ist "Identification".  
Globale Variable und solche, die der Kommunikation über Busverbindungen dienen (Access Variable), sind derzeit noch nicht eingebbar.
- Identification  
Wählen Sie einen technologie- bzw. auf die Anlage bezogenen Text als Bezeichner [siehe Seite 172].

## A.144 Project Member PRAP-Instance

- Allgemeines  
Instanz bedeutet das Bestimmen einer Identifikation und das konkrete Zuweisen aller im zugehörigen Typen deklarierten Variablen durch direkte Adressen (ADDRESS) und zugehörige Namen (NAME).  
Im Falle eines Unikates [siehe Seite 193] besteht die Instanz lediglich aus der FB-Nummer.
- IDENTIFICATION  
Teilen Sie der Instanz ein technologisch sinnvolles Kürzel [siehe Seite 172] zu, das natürlich innerhalb der Ressource nicht wiederholt werden darf.  
Empfehlung:  
Verwenden Sie die Bezeichnung aus dem Prozeßerlegungsgraphen.
- DESCRIPTION  
Tragen Sie hier eine technologisch zutreffende Bezeichnung [siehe Seite 143] ein, die auch mehrere Worte haben darf, um deutlich zu beschreiben, was die Instanz im Projekt steuern soll.  
Empfehlung:  
Verwenden Sie den Text aus dem Prozeßerlegungsgraphen.
- RELATIONS: TYPES  
Die Beziehung einer Instanz zu einem PRAP-TYP ist zwingend. Aus dem Angebot ist der zugehörige Typ auszuwählen.
- RELATIONS: TASKS  
Bei IEC 1131-3 vorgesehen: Neben der Task zur Programminstanz kann auch für eine FB-Instanz eine extra Aufruffolge erforderlich sein. Dann ist eine bei der Ressource deklarierte Task auszuwählen.
- RELATIONS: PARAMETER  
Die Tabelle enthält alle zum Typ gehörigen "FORMAL"-Parameter, denen Sie "ACTUAL"-Parameter zuweisen müssen, die zur Instanz gehören. Jede Instanz ist einem "PROGRAM" untergeordnet, dort sind die Aktualparameter eingetragen oder auch in der Ressource. Die Zuweisung fehlt bei allen mit "none" bezeichneten Aktualparametern.  
Klicken Sie auf  und erhalten alle im PROGRAM deklarierten Variablen und der Aktual-Parameter [siehe Seite 212] mit ihren direkten Adressen angezeigt.



### Anmerkung

Es fehlen alle lokal deklarierten (Constant, Local Static, Local Temporary), da diese automatisch Speicherplätze im RAM zugewiesen erhalten.

## A.145 Project Member PRAP-Type

- Allgemeines  
Es sind im Formblatt nur zwei Angaben einzutragen, die IDENTIFICATION, also ein Kurzbezeichner für den Code, den der Precompiler als Function\_Block aufbaut und einen ausführlicheren Text, DESCRIPTION.
- Identification  
Prägen Sie hierfür eine wenige Zeichen [siehe Seite 172] erforderliche Bezeichnung.
- Description  
Schreiben Sie hier einen ausführlicheren Text [siehe Seite 143], aber keine lange Beschreibung sondern einen technologisch relevanten Bezeichner.
- PARAMETER  
Die Variablen werden nur angezeigt. Zutreffend für den Typen-PRAP sind i.a. nur die FORMAL-Parameter. Einführen und bearbeiten können Sie diese nur in Zusammenhang mit dem PRAP, den Sie importieren, laden oder neu aufbauen müssen. Siehe PRAP-Editor. [siehe Seite 61]
- Sprachabhängig [siehe Seite 7] sind zusätzliche Angaben nötig. Im Fall "FB Settings" müssen Sie eine FB-Nummer eintragen.

## A.146 Project Member - Ressource

- Allgemeines  
siehe Ressource einfügen, bearbeiten [siehe Seite 55]  
Die Ressource wird praktisch durch eine Steuereinrichtung (z.B. SPS) bereitgestellt (ressource [siehe Seite 187]), vorgegeben sind damit die Kapazitäten wie z.B. die Anzahl der zulässigen Programmzeilen oder der Timer usw.  
Die globalen Variablen [siehe Seite 154] (PARAMETER) dienen der Kommunikation [siehe Seite 170] zwischen den Bausteinen, die zur Ressource "gehören". Außerdem können Sie direkt adressierte Variablen deklarieren
- Access Paths, Kommunikationswege zwischen Ressourcen und/oder Konfigurationen sind noch nicht vorgesehen!
- Identification  
Die Kurz-Bezeichnung [siehe Seite 172], die für die Station bzw. für die Steuereinrichtung wichtig ist, nicht den Hersteller und ähnliches, sondern z.B. St\_Eintr\_1, eintragen.
- Description  
Die herstellerbezogene Bezeichnung [siehe Seite 143] , aus der die Kapazitäten, die Leistungsfähigkeit der CPU bzw. der Steuereinheit hervorgehen, eintragen; z.B. für ein Siemens-Produkt S7-315. Dabei kann noch ein technologischer Bezug stehen wie z.B. "Reinigungs-Abteilung".
- PARAMETER - Globale Variable  
Ein Klick auf  öffnet das Eingabefeld für Globale Variable.  
Globale Variable sind innerhalb der Ressource für die Instanzen zu deklarieren. [siehe Seite 155]  
Diese sind bei der Instanziierung [siehe Seite 212] den Formal-Variablen der Typen-PRAP zuzuordnen.  
Wählen Sie zur Darstellung der dieser Ressource zugeordneten Variablen in 'Filter Options' "Member" und suchen Sie in 'Filter Values' die Ressource aus. Danach Klick auf "Filter Off".  
Fügen Sie mit  eine neue Variable hinzu. Nach dem Ausfüllen der Eigenschaften jeder Variablen

**müssen Sie** auf  klicken, um die Speicherung dieser Variablen, zugeordnet zur Ressource, in der Datenbank zu veranlassen. Ein Klick auf  ergibt zwar den Eintrag in die Datenbank, aber nicht die Zuordnung!



### Warnung

Vergessen Sie nicht, am Ende den OK-Button zu betätigen, sonst ist alle Mühe umsonst gewesen!

- Externe Variable  
sind derzeit nicht deklarierbar. Es handelt sich um solche, die in der Konfiguration global deklariert wurden.



### Tipp

Sie können auch die Variablen aus einer Excel-Liste importieren [siehe Seite 198].

## A.147 Project Member - Task

- Allgemeines  
siehe TASK bearbeiten, einfügen [siehe Seite 55]
- Identification  
Kurze Bezeichnung [siehe Seite 172], z.B.: Zyklisch, oder 300ms
- Description  
Hinweis auf Programm-Instanzen, in denen Task verwendet wird(Hinweis dazu [siehe Seite 143])

Weitere TASK-Themen:

SINGLE / INTERVAL und PRIORITY [siehe Seite 190]

Festlegungen für STEP 7 [siehe Seite 187]

Festlegungen für C und C++ [siehe Seite 147]

## A.148

Der Projektbaum ist eine graphische Darstellung aller zu einem Automatisierungssystem gehörigen Bestandteile - Mitglieder (=Members) - und ihrer funktionellen Zuordnung. Die Benennung der Mitglieder richtet sich nach der Internationalen Norm IEC 1131-3 wie Configuration, Resource, Task, Function Block, Program usw. Der Entwurf des Projektbaumes erfolgt mit dem Projekt-Editor und der eines Function Block mit dem PRAP-Editor.

## A.149

Ein binärer Prozeßzustand verbindet eine Quell- (Anfangs-, Start-) Operation Oj mit einer Ziel- (Folge-) Operation Ok. Sind beide Operationen gleich (j=k) heißt dieser Prozeßzustand Halteprozeßzustand, ansonsten Übergangprozeßzustand. Der binäre Prozeßzustand kann aus einem Pfad allein oder mehreren (alternativen) Pfaden gebildet werden.

Binäre Prozeßzustände entsprechen Transitionen im Petrinetz, wenn die Operationen den Stellen im Netz gleichgestellt werden.

## A.150

Schreiben Sie hier einen ausführlicheren Text, aber keine lange Beschreibung sondern einen technologisch relevanten Bezeichner.

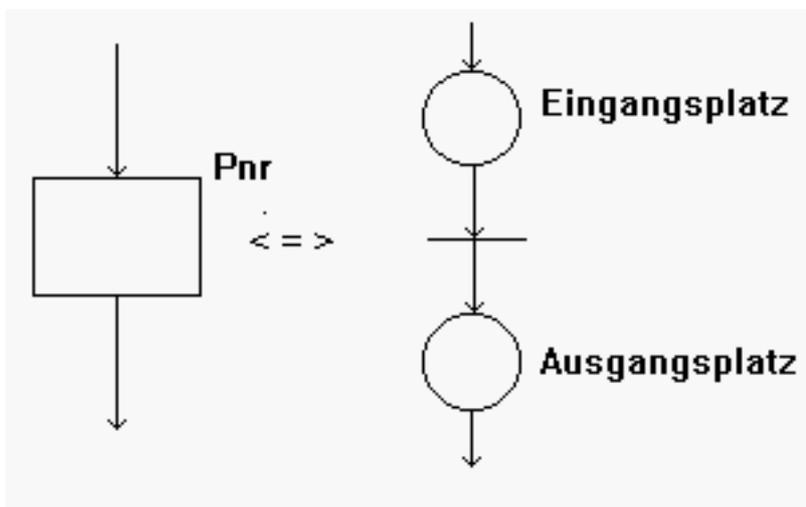
## A.151

Prägen Sie hierfür eine wenige Zeichen erforderliche Bezeichnung.

## A.152

Hier werden alle zum PRAP-TYP zugeordneten Variablen angezeigt.  
Bearbeiten können Sie Variable nur nach Öffnen des PRAP.

## A.153



Auf der Basis des

Bedingungs-Ereignis-Netzes betrachtet ist ein Prozessschritt P<nr> eine Sequenz von Bedingung - Ereignis - Bedingung, hier dargestellt als

--> Eingangsplatz --> Transition --> Ausgangsplatz -->

## A.154

### Prozeßsituation

Für die Dauer der Ausführung einer Operation, z.B. O7, ändert sich deren Halte- oder Stabilitäts-Prozeßzustand, P77, nicht. Enthält er im PRAP mehrere Pfade, so kann es sein, daß zwischen ihnen die Belegung der P-Variablen während der Situationsdauer wechselt.

Die beiden zusammengehörigen Größen Operation und Halteprozeßzustand stellen eine Prozeßsituation Sit7 = (O7, P77) dar, auch kurz mit Situation bezeichnet.

Übergangssituationen gibt es nicht, nur Übergangs-Prozeßzustände, weil die Übergangsdauer zwischen Situationen im PRAP null ist. Die Steuereinrichtung benötigt dazu genau einen Arbeitszyklus.

Eine Situation kann auch als WHILE P77 DO O7 gelesen werden, oder REPEAT O7 UNTIL(P77 = false)

## A.155

Prozeßzustandsvariable (kurz Prozeßvariable oder p-Var) stellen die mit ja/nein bzw. Yes / No (binär) bewerteten Ausgangsvariablen eines (Teil-)Prozesses, auch als Prozeßgrößen bezeichnet, dar. Es handelt sich um

- natürliche, technologisch notwendig zu erfassende Zustandsveränderungen von Produkten, Energien, Informationen usw.

- Informationen vom den Prozeß bedienenden Menschen (Operateur).
- Reaktionen von Zeit- und Zählgliedern auf Grund ihrer Ansteuerung.
- Ergebnisaussage einer Vergleichsfunktion
- natürliche und erzeugte Koppelvariable aus anderen Teilprozessen, den "Empfang" von Steuerinformationen im Rahmen der Koppelbeziehungen.

## A.156

Hier tragen Sie den Namen der formalen P-Variablen ein, eine Kurzfassung zum "Statement".

Beachten Sie dabei den Hinweis

Lesen Sie auch den Text zu "ADDRESS".

Neu aber wiederholt verwendete P-Variable für Kopplungszwecke können Sie durch Nutzung des Filters ( Member = PRAP-Name der P-Variablen ) finden. Dann mit Klick auf  eine neue Variable anlegen und die Daten der ausgewählten kopieren.



### Anmerkung

Die unmittelbare Übernahme einer P-Variablen ist gesperrt, da sie dem anderen Member - hier Typen-PRAP - zugeordnet ist. Eine Kopie ordnet den gleichen Namen dem aktuellen Typen-PRAP zu.

## A.157

Die Variablen-Adresse ist hier nicht eintragbar, mit Klick auf  öffnet sich das zugehörige Formblatt.

## A.158

Die Eigenschaft "Generated" ist hier nicht eintragbar, mit Klick auf  öffnet sich das zugehörige Formblatt.

## A.159

Bitte wählen Sie aus dem Angebot die zutreffende Eigenschaft aus. Für formale Parameter kommen nur Input, Output oder Input/Output in Frage.

## A.160

Tragen Sie hier nur Funktionen-Hinweise, keine konkreten EMSR-Stellen ein. Z.B. bei einem allgemeinen Positionserfassung GS oder Füllstand LS, oder HS für Bedienerinfluß usw.

Konkrete EMSR-Stellen müssen bei den Aktualparametern bzw. direkt adressierten Variablen der Teilprozesse - Instanzen der PRAP-Typen - eingetragen werden. Und bei Unikaten oder in Typen mit beschränkter Anzahl der Formalparameter.

Sollten Sie Betriebsmittelkennzeichen statt EMSR-Stellen nutzen, gilt das Gesagte sinngemäß.

## A.161

Die Variable ist hier nicht eintragbar, mit Klick auf  öffnet sich das zugehörige Formblatt (weitere Hilfe dort!).



### Anmerkung

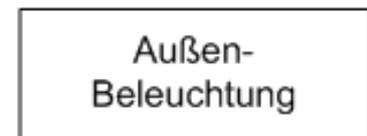
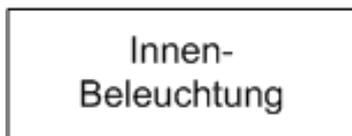
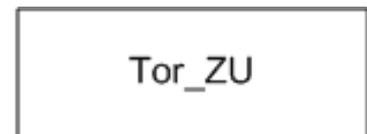
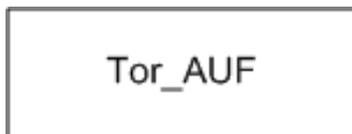
Sie können eine Prozeßvariable nur dann wiederholt verwenden, wenn Variable Name (Flag) ausgefüllt ist!

(Flag) deutet auf die Eigenschaft der hier einzutragenden Booleschen Variablen hin. Sie zeigt an, sofern sie mit einer Funktion (siehe Choose a Function) verbunden ist, ob das Ergebnis true oder false ist. Deshalb wird sie auch Ergebnis-Variable genannt.

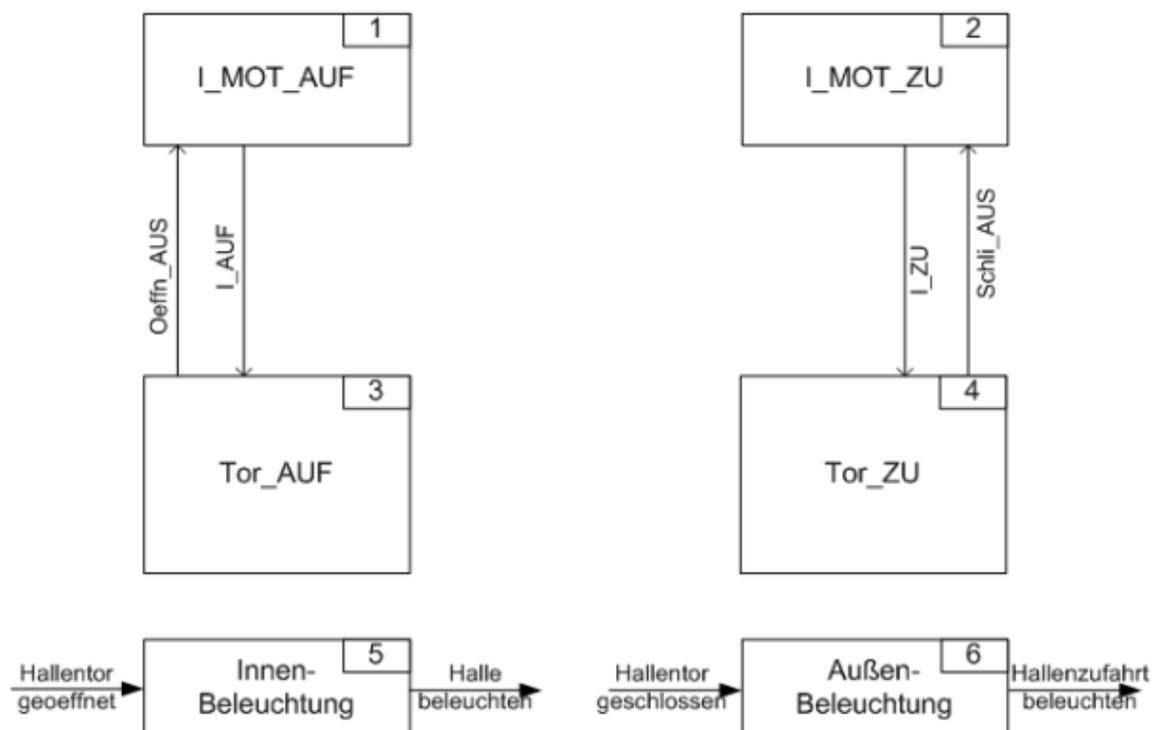
## A.162

Der Variablenbezeichner ist hier eintragbar oder änderbar und wird mit Klick auf "OK" in die Liste der Prozeßzustandsvariablen NUR DANN übernommen, WENN auch ein VARIABLE NAME (Flag) vorhanden. Anderenfalls wird der Bezeichner nur ohne Bezug zwischengespeichert und die Eintragung in die "Condition" erfolgt. Mit Klick auf "Cancel" erfolgt kein Eintrag in die Variablenliste, also bleibt der alte Text.

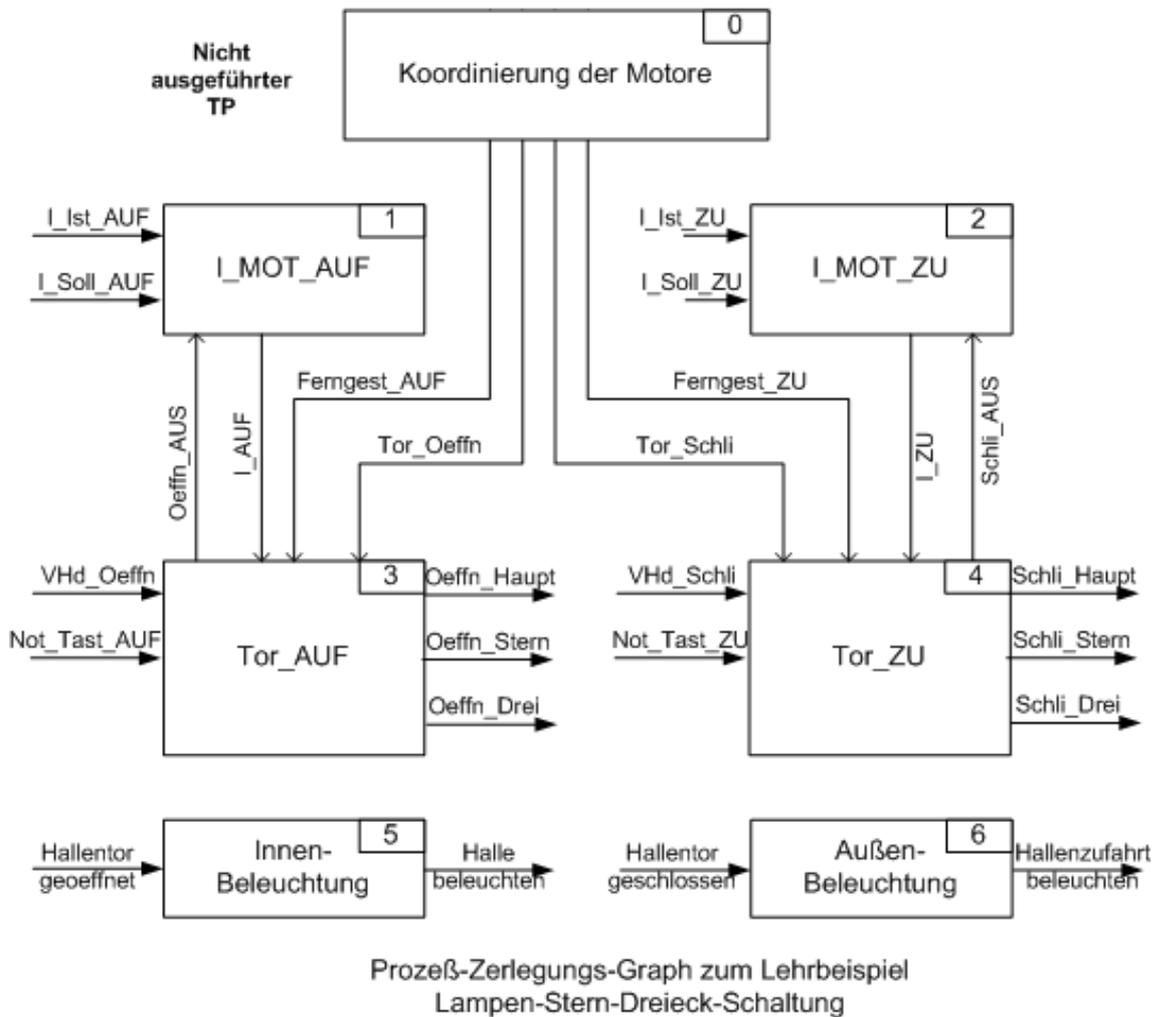
## A.163



## A.164



## A.165



## A.166

Für Variable mit fester Adresse bleibt nur "active" zu wählen

## A.167

Eine herstellerbezogene Bezeichnung, aus der die Kapazitäten, die Leistungsfähigkeit der CPU bzw. der Steuereinheit hervorgehen, muß hier stehen; z.B. für ein Siemens-Produkt S7-315. Zusätzlich ein technologischer Bezug wie "Anfahrüberwachung". Insgesamt damit S7-315 / Anfahrüberwachung.

## A.168

Ein Kürzel, das die Automatisierungsstation kennzeichnet, z.B. STAT\_1, muß hier eingetragen sein.

## A.169

Diejenige EMSR-Stelle eingetragen, die der Variablen zugeordnet ist und wie sie im Technologie-Schema, z.B. R-I-Fließbild, steht. Bei verfahrenstechnischen Prozessen ist die DIN 19227 verbindlich.

Bei anderen Prozessen das Betriebsmittelkennzeichen wie Y für Antrieb, B für Melder.

Koppelungen sind hier ebenfalls eintragbar. Vorläufig nur als Information für den Leser dieser Tabelle.

## A.170

Angezeigt werden alle in der Ressource global deklarierten Variablen. Falls Sie noch keine hier eingeführt haben, dann finden Sie eine Hilfe mit dem HELP-Button unter "PARAMETER".

## A.171

Hier wird angezeigt, welchem Member die Variable zugeordnet ist, bei Mehrfachzuordnungen natürlich nicht, sondern "Multiple Choise" ist dann zu lesen.

## A.172

### Resource (Ressource)

ist ein Begriff für eine Steuerungseinheit. Sie kann eine SPS, ein Prozeßrechner, eine Prozeßstation, ein Mikrokontroller, eine Soft-SPS usw. sein.

Zur Zeit ist nur eine Ressource zugelassen!

Zur Identifikation erhält sie eine Bezeichnung wie z.B. Station\_A, Pumpenhaus, Nebengebäude\_7 usw. Damit sind Bereiche des Automatisierungssystems gemeint.

Zur Hardware-Bezogenheit dient die Beschreibung (=Description), eine Form der Instanz für die Ressource. Hier ist also einzutragen, wie Steuereinrichtung "amtlich" benannt wird, also z.B. für Siemens-SPS S5-135U, S7-223, S7-415 oder für Schneider-Electric- SPS Compact, Quantum, Momentum usw.

Vergessen Sie nicht, Project|Options bzw.  anzuklicken und das Formular auszufüllen.

## A.173

Im Feld USAGE. beschränkt sich die Auswahl auf Constant, Address Input, Address Output und Address Memory

## A.174

In das Feld ADDRESS geben Sie nur noch die Ziffernfolge für die Adresse ein! Falls erforderlich, die Byte-Bit-Trennung nicht vergessen.

## A.175

Tragen Sie in NAME den Variablenbezeichner ein, der hier als Symbol für die anzugebende Hardwareadresse gilt.

## A.176

Stellen Sie die erste und letzte zu übernehmende Zeile aus Ihrer Tabelle ein. Sie können die Zeilen-Nummer auch direkt in das Feld schreiben. Gleiche Zeilennummer bedeutet, nur diese eine Zeile zu importieren.

## A.177

In der nachstehenden Tabelle sind einige OB aufgeführt, die auch in SPAS programmiert werden dürfen.

| PRIORITY  | 0 | 1 | 2               | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10   | 11   | 12   | 13   | 14   | 15   |
|-----------|---|---|-----------------|----|----|----|----|----|----|----|------|------|------|------|------|------|
| OB-Nummer |   | 1 | 10<br>bis<br>17 | 20 | 21 | 22 | 23 | 30 | 31 | 32 | 33   | 34   | 35   | 36   | 37   | 38   |
| INTERVAL  |   |   |                 |    |    |    |    | 5s | 2s | 1s | 0,5s | 0,2s | 0,1s | 50ms | 20ms | 10ms |

---

| PRIORITY  | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25               | 26 | 27                  | 28 | 29 | 30 | 31 |
|-----------|----|----|----|----|----|----|----|----|----|------------------|----|---------------------|----|----|----|----|
| OB-Nummer | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |    | 60,<br>70,<br>80 |    | 100,<br>101,<br>102 | 72 | 90 |    |    |

#### Priorität, OB-Nummer und Intervall-Zeiten bei STEP 7

Die OB 121 und 122 sind nicht aufgenommen, weil solche Fehler eigentlich nicht auftreten sollten. Für weitergehende Fragen müssen Sie sich bitte in der Hilfe von STEP 7 Antwort suchen.

## A.178

### MEMBER

Es wird Ihnen angezeigt, welchen PRAP-Knoten (dessen Identification) Sie im Projektbaum markiert haben, weil automatisch das Filter gesetzt wurde.

Falls Sie den Filter-Button anklicken (Filter Off) oder , können Sie die Zuordnung der jeweiligen Variablen hier ablesen.

## A.179 Hinweis zu Umgehung der Peripherieadressen bei STEP 7

"Siemens Prosave" kann die Peripherieadressen PEB .. und PA .. in Instanzen nicht verarbeiten. Daher ist eine Sonderbehandlung wie folgt zu empfehlen.

Schreiben Sie z.B. in WORDPAD eine Funktion:

```
FUNCTION FC[nr]:VOID
```

```
BEGIN
```

```
L PE....
```

```
T MB.... (MW...,MD..)
```

```
END_FUNCTION
```

ändern die Endung .tx in .AWL und importieren in STEP 7. Oder Sie schreiben eine Lade-Transfer-Funktion in STEP 7, in der die betreffende(n) Peripherieadresse(n) auf den RAM abgelegt werden. Diese RAM-Adresse(n) geben Sie im SPAS-Projekt an.

Natürlich muß diese FC übersetzt und dann mit in den OBI eingetragen werden, möglichst vor dem eigentlichen Programm.

Eine zweite Möglichkeit besteht darin, im PRAP-Typen für die Peripherieadressen eine "harte Adressierung" einzuführen. Dann gibt es in der Instanz keine Adresszuweisung. Der OnlineViewer überspringt dann diese Adresse, wodurch Sie deren wahren Wert nicht dargestellt bekommen.

## A.180

Eine Prozeßvariable heißt statisch, wenn sie eine neue Aussage mindestens solange beibehält, wie eine oder mehrere davon abhängige stabile Operationen auf den Prozeß einwirken.

## A.181

Hier sind Binärsteuerungen = binary control gemeint.

Es braucht nicht in Verknüpfungs- und Folgesteuern unterschieden zu werden, weil das Entwurfsverfahren mit Hilfe des PRAP das erübrigt, also PreCOM dies automatisch erkennt und im

Compilat verwirklicht.

## A.182

### Steuerungs-System

Mit diesem Begriff ist gemeint, wenn im Ergebnis der Prozeßzerlegung und der Binären Prozeßanalyse eine Anzahl gekoppelter Teilprozesse vorliegt.

Teilsysteme können z.B. dadurch entstehen, daß ein (Gesamt-)Steuerungs-System zerlegt wird, um auf mehreren Steuer-Geräten (Ressourcen) verteilt zu werden. Das Zerlegen "durchtrennt" Koppelungen, die im Gesamtsystem vorgesehen waren.

Jedes Teilsystem wird auf einer Ressource implementiert und kommuniziert mit Nachbarsystemen so, daß die Koppelungen als Access Paths ausgewiesen die Gesamtfunktion garantieren.

## A.183

Der Begriff Struktur sagt aus, daß es eine Menge Elemente gibt, die untereinander zumeist paarweise eine Relation besitzen.

Die Elemente sind hier Variable, jede mit dem ihr zugeordnetem Datentyp. Die Relation der Variablen ist die Zugehörigkeit zu einem Bezeichner, Name der Struktur, einer Baum-Struktur.

Die Norm IEC 1131-3 nennt unter dem Abschnitt "Derived data types" auch die "STRUCT declaration" und meint damit structured data types.

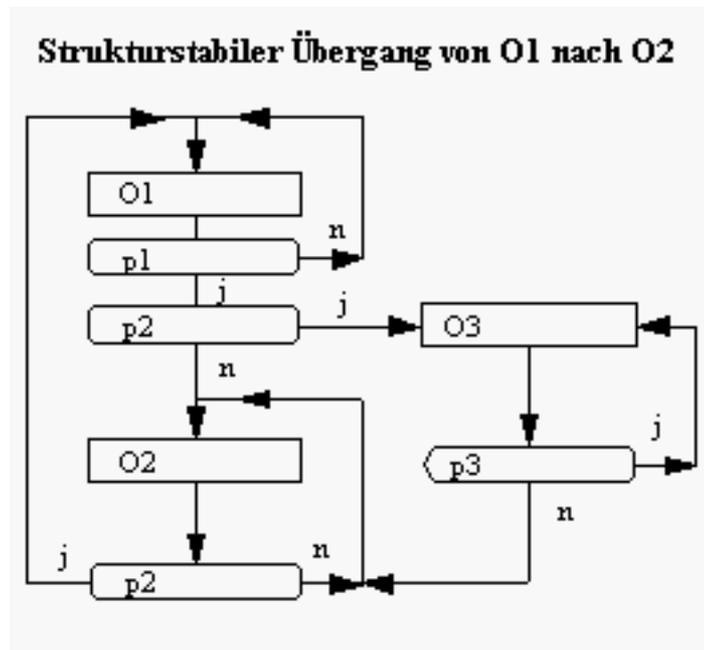
Zu den Elementen gehören die elementaren Datentypen wie BOOL bis REAL, ferner die Standard-Funktionstypen TON, CDUD, TIME, DATE usw., und es werden alle PRAP-Typen als funktionelle Strukturen eingeordnet sowie die (baum-)strukturierten Datentypen.

Der Projektant eines Automatisierungssystems kann in SPaS sogenannte User Defined Types (UDT, angelehnt an SIEMENS) definieren.

Siehe auch: Einfügen von UDT in Ressourcen [siehe Seite 39]

## A.184

Strukturstabilität:



Ein Übergang von

einer Operation  $O_i$  zur Folgeoperation  $O_k$  heißt strukturstabil, wenn der Haltezustand zu  $O_k$  abhängig von Bedingungen des Übergangszustandes erfüllt wird.

In der Abbildung ist zu sehen, daß im Übergang von Operation 1 zu Operation 2 schon eine Bedingung, die Prozeßvariable  $p_2 = \text{nein-Belegung}$ , für die Stabilität der Operation 2 (sprich für den Haltezustand der Operation 2) enthalten ist.

Der Übergang von Operation 1 zu Operation 3 ist bedingungsstabil.

## A.185

Ein Graph ist stark zusammenhängend, wenn von einem Knoten  $K_j$  jeder andere Knoten  $K_k$  und auch  $K_j$  wieder selbst über einen Weg erreicht werden kann.

Stark Zusammenhängend (= strong connected) ist ein PRAP, wenn ausgehend von einer Operation  $O_j$  jede andere Operation  $O_k$  und sie selbst erreicht werden kann.

Ein Weg in einem Graph setzt sich aus mehreren Pfaden zusammen. Dabei ist im PRAP diese Zusammensetzung auch mit einem Wertewechsel der P-Variablen in Pfaden zugelassen.

## A.186

### SINGLE/INTERVAL

Entweder Sie benötigen eine zeitzyklische Aufruffolge oder eine ereignisgesteuerte (Hardware Interrupt). Tragen Sie also einen Zeitwert oder einen Variablen-Namen ein. Diese Variable muss natürlich vom Typ BOOL sein!

Dieses Feld muss unbeschrieben bleiben, wenn es sich um rein prioritäten-gesteuerte Aufrufe handelt.

### PRIORITY

Die Prioritäten der Tasks sind von 0 bis 31 verfügbar. Wählen Sie die für Ihren Zweck richtige aus [siehe Seite 7].

## A.187

Eine Teil-AWL ist die Anweisungsliste zu einem PRAP, also einem Teilprozeß, wenn es sich um einen Typen-PRAP oder TP-PRAP (Unikat-PRAP) handelt. Oder eine Teil-AWL entspricht einer Instanz, einem Instanzbaustein.

## A.188 Funktions-Test eines PRAP - Offline Test

Voraussetzung zum Offline-Test ist die Öffnung der [Dateiname].onl, wodurch Sie links im Bild den Baum der Instanzen, der Teilprozesse, sehen. Nach dem Markieren eines Teilprozesses können Sie die aktuellen Daten der Variablen durch Klick auf die rechte Maustaste sehen.

Öffnen Sie diesen durch Doppelklick. Im rechten Feld sehen Sie den PRAP des Typen, aber mit den aktuellen Parametern der Instanz, also des Teilprozesses.

Wenn Sie den Cursor auf ein Condition-Element führen und die rechte Maustaste betätigen, haben Sie die Wahl, den Wert zu ändern, d.h. von No in Yes zu wechseln oder umgekehrt. Der Farbumschlag zeigt Ihnen den neuen aktuellen Pfad, und falls dieser zu einer Folgeoperation führt, dann auch diese in roter Farbe. Wenn Sie nun auf die neu erreichte Operation den Cursor stellen, dann können Sie mit einem rechten Mausklick und "Toggle Value" die Vorgängeroperation zum Farbumschlag bringen, d.h. die neue Situation ist eingenommen.



ermöglicht Ihnen, die Schritte Ihrer Testreihe abzuspeichern. Nach jedem vollzogenen Schritt

müssen Sie diesen Button klicken oder nicht, falls ein Schritt nicht registriert werden soll. Die Buttons links und rechts des Indexanzeigers gestatten das wiederholte Beobachten der Schritte:



## A.189 Zeitglied-Ausgang Q / Timer Output Q

Bitte beachten Sie zwei Möglichkeiten. Sie erwarten die Ausgabe mit true auf der P-Variablen (Condition):

- Es handelt sich um einen im PRAP kreierte Timer, dann suchen Sie "NAME" dieses Timers und übernehmen alle weiteren Eintragungen, Daten.

- Es handelt sich um einen Timer in einem anderen PRAP, dann suchen Sie in diesem MEMBER nach "NAME" des Timers und übernehmen alle weiteren Eintragungen, Daten, durch Klick auf .

## A.190

- Name  
ist ein Kurzzeichen zu Statement und nur in der Variablen-tabelle (Klick ) editierbar. Beachten Sie die Hinweise bei Wählbare Sprachen???
- Statement  
ist die Benennung des Timers wie z.B. Schließzeit, Anlaufdauer usw. Schreiben Sie hier einen Text, der zeigt, wofür der Timer nötig ist, also niemals T121!
- Variable Usage  
im allgemeinen "Local Static" (In Version 1. nicht: "Address Memory" für fest vorgegebene Timer, z.B. T4) oder ist sprachabhängig. [siehe Seite 7]

## A.191

- "Name" ist ein Kurzzeichen zu Statement und nur in der Variablen-tabelle (Klick ) editierbar. "Name" erhält entweder eine feste Zeit als Konstante zugewiesen oder (idh\_WaehlbareSprachen [siehe Seite 7]) ist ein Formalparameter, dem die Instanz des PRAP-Typen einen Zeitwert übergibt oder es ist ein Speicherplatz, dem anderweitig der Zeitwert übertragen wird.
- "Statement" bezeichnet die Zeitgröße z.B. als Wartezeit, Sicherheitswert, Kontrolldauer usw.
- "Variable Usage" charakterisiert mit
  - "Constant" einen festen Zeitwert,
  - "Local Static" bei Übergabe von verschiedenen Zeitwerten über einen Speicherplatz im RAM,
  - "Input" oder "Input/Output" bei Übergabe von verschiedenen Zeitwerten über einen Speicherplatz, der mit einer direkten Adresse instanziiert wird.
- "Timer Value" die vorgegebene Maske ist nicht auszufüllen, wenn "USAGE = Constant" gewählt wurde. Sie müssen den Wert als INITIAL VALUE(AccessInitialValue [siehe Seite 137]) in das Formular nach dessen Öffnen eingeben.

## A.192

Teilprozeß, kurz TP, heißt jede Einheit aus dem gesamten Steuerungsbereich (im Prozeßerlegungsgraphen zu erkennen) eines Prozesses.

Ist ein Teilprozeß durch einen PRAP beschrieben, so stellt der PRAP eine Teilprozeßbeschreibung dar. Dabei sind die Variablen formaler Natur, sind also nicht mit direkten Adressen verbunden. Der PRAP ist daher der Typ des TP, dessen Variable jedoch adressiert bzw. Konstante sind. Sie heißen aktuelle Parameter, weil sie für den TP speziell gelten. Jeder TP ist demnach eine Instanz des PRAP.

Auch eine Stabilisierungsaufgabe für eine oder mehrere Prozeßgrößen, also eine Regelungsaufgabe, ist ein Teilprozeß, der z.B. durch ein PID-Modell beschrieben wird.

Jeder PRAP muß einen einmaligen Namen pro Ressource haben. Jeder Teilprozeß als Instanz eines Typen-PRAP muß einen einmaligen Namen in "seinem" PROGRAM haben. Er muß im Prozeßerlegungsgraphen - PZG - dieser Ressource enthalten sein.

#### **Wichtige Eigenschaft eines TP:**

Jeder Teilprozeß aus dem PZG ist als Instanz [siehe Seite 126] oder als Unikat zu verstehen. Zu jedem Teilprozeß gehört eine Anzahl Eingangsvariable??? und Ausgangsvariable [siehe Seite 138]. In **keinem anderen Teilprozeß** des Teil-Projektes, also innerhalb einer Ressource, darf es dieselben Ausgangsvariablen, im Blick auf die Instanz - dieselben Ausgangsvariablen mit direkten Booleschen Adressen - nicht noch einmal geben! Mengentheoretisch bedeutet dies, daß die Schnittmenge der Mengen der Ausgangsvariablen zweier TP leer ist.

Die durch arithmetische oder Kopier-Ausdrücke sich ergebenden Ausgangsvariablen dürfen in mehreren Instanzen dann gleich sein, wenn sie nicht innerhalb eines Arbeitszyklus' der Steuereinrichtung "geschrieben" werden.

## **A.193**

Transition ist ein Begriff aus der Theorie der Petrinetze. Er wird original in diesem Sinne Im Porzeßablaufnetz verwendet. Wird dieser Begriff mit dem PRAP in Zusammenhang gebracht, so meist im Sinne des Übergangs-Prozeßzustandes.

Das englische Wort transition kann mit Übergang übersetzt werden, also auch im obigen Sinne gebraucht werden.

## **A.194 Typen-PRAP: Eingabe und Bearbeitung der Variablen**

Das Datenbankhandling erfolgt mit den Arbeitssymbolen:



Die Symbole bedeuten:

- Neuen Datensatz einer Variablen hinzu
- Datensatz zur markierten Variablen löschen
- Datenbank-Inhalt editieren
- alle Änderungen in Datenbank sichern
- Editiervorgang ungeschehen beenden
- Datenbankinhalt erneut sichern
- Daten übernehmen, zugleich in Datenbank sichern
- alle Variablen-Records anzeigen

In dieser Tabelle haben Sie Zugriff zu allen Variablen, die zur Ressource gehören. Dabei haben Sie die Wahl, die Variablen ohne Filter-Auswahl [siehe Seite 153] oder unter bestimmten Bedingungen dargestellt zu bekommen. Im Einzelnen stehen folgende Felder zur Verfügung:

- MEMBER [siehe Seite 188]
- NAME [siehe Seite 177]: Im Fall einer Operationsvariablen.  
NAME [siehe Seite 184]: Im Fall einer Prozeßvariablen.
- DATATYPE [siehe Seite 137]
- Address [siehe Seite 137]
- POINT OF AUTOMATION [siehe Seite 137]  
Hier wird entweder übernommen, was unterhalb "Statement" eingetragen wurde, oder Sie schreiben hier den Text ein.
- INITIAL VALUE [siehe Seite 137]

- PARAMETER [siehe Seite 137]
- USAGE [siehe Seite 138]
- Anzeigefeld [siehe Seite 166]



### Tipp

Mit Klick auf  können Sie bei zu vielen Variablen einen Suchvorgang nutzen.

## A.195

### Typenprozeßablaufpläne (TTP), auch Typen-PRAP,

sind Funktionstypen, deren Variablen - zum Teil oder vollständig - keine direkten Adressen haben. Sie werden hauptsächlich bei sich strukturell wiederholenden Teilprozessen eingesetzt, wie z.B.

Antriebsteilprozesse,

Alarmteilprozesse,

Signalisierungsteilprozesse.

Die internationale Norm IEC 1131-3 und C/C++ gestatten keine direkten Adressen in Typen-Funktionsbausteinen. Das bedeutet, auch bei Unikat-TP (TP-PRAP) muß ein Typ entworfen werden, dem genau eine Instanz zugeordnet wird.

Zwei Typen-TP dürfen die gleichen Variablenbezeichner (Statements) haben, weil die Variablen an den TTP gebunden sind (Kapselung).

## A.196 Strukturierte Datentypen

Hier können Sie in Anlehnung an die Datenbausteine, wie sie von der Fa. Siemens bekannt sind, selbst solche entwerfen. Deshalb wird auch der Begriff UDT [siehe Seite 39] (USER DEFINED TYPE) zuweilen verwendet. Grundsätzlich ist nach IEC 1131-3 "structured data type" unter "derived data types" zu finden. Aber in dieser Form ist in der SPAS-Version 1 noch keine Deklaration möglich.

IDENTIFICATION bedeutet den Namen [siehe Seite 172], der kurz sein sollte.

DESCRIPTION bedeutet eine Beschreibung [siehe Seite 143], die z.B. den Einsatz-Zweck benennt.

Einfügen von UDT in das Projekt: UDT\_einfuegen [siehe Seite 39]

Siehe auch: UDTInstance [siehe Seite 209]

## A.197

Wie bei Datenbausteinen werden hier einfach laufende Nummern eingesetzt.

## A.198

Ein Pfad allein oder mehrere Pfade bilden einen Übergangsprozeßzustand =Transition (kurz Übergangszustand). Er stellt den Übergang zwischen der Operation Oj und deren Folgeoperation Ok dar. Jeder Übergangszustand enthält die für den Übergang signifikanten Prozeßvariablen in geeigneter Bewertung und Verkettung, so daß die sich ergebende Aussage für jeden Pfad des Übergangszustandes eindeutig begründet und zu jedem anderen widerspruchsfrei ist.

## A.199

Ein PRAP, der keine Variablendeklaration benötigt, enthält ausschließlich direkte Adressen [siehe Seite 123]. Das bedeutet aber, es gibt ihn nur in dieser Form einmalig. Beispiele hierfür sind Koordinierungs-Teilprozesse, die speziell für eine Technologie gelten. Auch Teilprozesse der Mensch-Maschine-Kommunikation tragen häufig diesen einmaligen Charakter.

Die Norm IEC 1131-3 gestattet solche Unikate nicht, weil innerhalb eines FUNCTION BLOCK keine Hardware-Adressen zulässig sind! Das gilt ebenso für C/C++. Folglich muß ein Typ entworfen werden, dem eine einzige Instanz zuzuordnen ist. Dann ist diese Instanz ein Unikat.

## A.200

Die Art der Verwendung der Variablen - Variable Usage - wird hierher übernommen. Sie können aber auch ändern!

## A.201

### Variable

sind veränderliche Größen, Parameter, die funktionell in PRAP und dort eingebetteten Funktionen verarbeitet werden. Jede Variable sollte eine technologisch sinnvolle Bezeichnung (**STATEMENT**) und **MUSS** eine Identifikation, ein Kurzzeichen (**IDENTIFICATION, NAME**) haben.

Zu jeder Variablen gehört zwingend ein Datentyp, wodurch die binäre Darstellung vorgegeben ist.

Die Verwendung der Variablen (USAGE) ist durch den Projektanten zu überprüfen. Insbesondere sind lokale Variable im PRAP zu deklarierbar. Diese nur für die funktionellen Eigenschaften im PRAP zuständig und nutzbar.

Globale Variable sind dagegen in der Konfiguration, der Ressource und im Programm zu deklarieren.

## A.202 Variable der Vergleich-, Kopier- und Arithmetik-Funktionen

Jede Variable muß zuerst mit einem "NAME", der Variablen-Identifikation [siehe Seite 172], eingeführt werden, bevor sie in Formelausdrücken verwendet wird. Statement, also Erklärung in Kurzfassung [siehe Seite 143] und Usage, die Verwendung jeder Variablen werden im oberen Bereich festgelegt. Im unteren Bereich sind - je nach Filter - alle Variablen angezeigt.

Mit Taste "F1" oder  sind auch Einzelinformationen zu den Feldern zu erhalten.

Im unteren Anzeigefeld [siehe Seite 148] ist es mit RECHTER Maustaste möglich, weitere Informationen zu einer Variablen zu erhalten. Die Anzeige erfolgt unterhalb der Tabelle, die Anzeigefläche ist nach oben hin vergrößerbar.<sup>39</sup>

- Symbole  und  für "POST" und "ACCEPT" richtig bedienen: WICHTIG [siehe Seite 153]
- NAME  
Falls nicht schon vorhanden, auf  klicken, um eine neue Variable einzuführen. Die Kurzbezeichnung zum Statement (siehe oben!) eintragen, dabei bitte beachten: NAME [siehe Seite 172]  
Falls schon vorhanden, dann Zuordnung im Fenster "Member" sichtbar.
- DATATYPE  
BOOL ist nur sinnvoll bei der Zuweisung von 0 (false) oder 1 (true), also in einer Kopierfunktion.  
BYTE, WORD, DWORD, LWORD nicht in arithmetischen und Vergleich-Ausdrücken einsetzen, weil nur mit Vorzeichen behaftete Größen dort zugelassen sind.  
BCD-Typen nur einsetzen, wenn bekannt ist, daß sie in der Zielsprache akzeptiert werden.  
TON, CTUD An Timer und Counter können mit COPY Werte zugewiesen werden, auch der Vergleich mit einem Sollwert ist sinnvoll: Für TIMER mit Datentyp TIME, für Counter mit Datentyp INT.  
CONSTANT sind feste Werte, die in der Instanz auch keine Änderung erfahren. Eintrag des Wertes bei Initial Value (siehe dort).
- ADDRESS  
Hier nur Hardware-Adressen eintragen, wenn der PRAP ein Unikat [siehe Seite 193] ist oder alle

<sup>39</sup> Cursor auf obere Begrenzung, bis Cursorbild wechselt, linke Maustaste festhalten und Rand nach oben schieben.

Instanzen die gleiche Adresse haben. Für diese Adressen gilt auch PARAMETER "actual".

- **POINT OF AUTOMATION**  
Falls USAGE mit Input, Output oder Input/Output erklärt, ist eine EMSR-Stelle zu erwarten. Tragen Sie bei PARAMETER "formal" hier höchstens die Buchstabenkombination des Meß- oder Stellortes ein. Nur bei PARAMETER "actual" hat die komplette Eintragung einen Sinn.
- **INITIAL VALUE**  
Hier tragen Sie vor allem auch Konstant-Werte ein, nicht nur Anfangswerte. Zeitwerte: T#.H..M..S...MS. Es müssen nicht alle Zeit-Maße angegeben werden, z.B. T#580ms.

## A.203 Variables - Anzeige geordnet nach P-, O-, sonstige Variable

Mit den Agaben in den Tabellen haben Sie umfassenden Zugriff

### A.204

Variable Usage - Verwendung der Variablen bedeutet im einzelnen:

- **NotSpecified**  
Die Variable hat keine Festlegung.  
Fehler beim Compilieren sind zu erwarten.
- **Constant**  
Diese Verwendung kann sowohl Formal- [siehe Seite 125] als auch Aktualparameter [siehe Seite 124] betreffen.  
Die (Symbol-)Variable ist eine Konstante, also kein Speicherplatz im Eingangs-, Ausgangs- oder Merkerbereich. Der Variablen werden nur feste Werte, gemeint sind Zahlen, zugewiesen.
- **Input**  
Die Variable ist ein Formalparameter und wird ausschließlich als Eingang des Funktionsbausteins verwendet. Sie darf also nicht überschrieben werden, z.B. durch die Funktion Copy.
- **Output**  
Die Variable ist ein Formalparameter [siehe Seite 125] und wird ausschließlich als Ausgang des Funktionsbausteines verwendet. Eine Verwendung als Eingang erfolgt nicht. Output-Variable erhalten im Rahmen der funktionellen Verknüpfung Werte eingetragen. Z.B. bei den Funktionen Copy, Arithmetik, Comparison (true / false) (Vergleich).
- **Input/Output**  
Die damit gekennzeichneten Variablen sind Formalparameter und werden auch "Durchgangvariable" genannt. Z.B. erfolgt die Übertragung einer Größe auf eine Variable (Copy-Funktion) zunächst als Ausgabe (Output) - diese Größe wird auf einen Ausgabespeicherplatz abgelegt -, danach aber als Input bei einem Vergleich als Istwert verwendet.
- **Local Static**  
1. Variable sind Formalparameter [siehe Seite 125], die den Funktionsbaustein, den Typ(!), nicht von außen erreichen und nicht nach außen verlassen. Falls z.B. der soeben beschriebene Vergleich nur innerhalb des Bausteins funktioniert. Speicherwerte bleiben dem Funktionsbaustein ständig erhalten.  
2. Variable sind Aktualparameter [siehe Seite 124], wenn sie in einem Programm oder einer Ressource deklariert werden. Z.B. Koppelvariable können so deklariert werden.
- **Local Temporary**

Zunächst gilt die Eigenschaft "Local", aber mit dem Aufruf eines anderen Funktionsbausteins oder Programms durch das Betriebssystem werden alle hier angelegten Speicherbereiche freigegeben. Geeignet für Vergleichsergebnisse.

- **Address Input, Address Output, Address Memory**

Eigenschaften gelten für Instanzen!!

Aber auch in Typen-PRAP können feste Adressen vorgegeben werden, wenn die Anzahl der formalen Variablen für eine Ziel-SPS beschränkt ist. Unikate erhalten ausschließlich feste Adressen.

Es handelt sich um Aktualparameter [siehe Seite 124]: Fest vorgegebene Hardware-Adressen (direkte Adressen [siehe Seite 123]) als Input oder Output der Steuereinrichtung. Liegt eine Adresse im Merkerbereich, muß sie als Address Memory hinterlegt werden.

## A.205

### Eingabemöglichkeiten für Variable in Formelausdrücken

Die Tasten "Strg" ("Ctrl") + "Leerzeichen" ("Space") zeigen alle Variablen. Zur Nutzung als Operanden wählen Sie mit Klick den jeweils richtigen aus.



#### **Tip**

Nach Öffnen des Fensters blinkt der Cursor darüber. Geben Sie mit der Tastatur die gewünschte Zeichenfolge ein, um den Operanden leichter zu finden.

Die Tasten "Strg" ("Ctrl") + "J" zeigen Ihnen alle Operationen, im Fall der Vergleichs-Ausdrücke auch die Vergleichsoperanden. Sie dürfen anstelle eines Vergleichs-Operanden einen arithmetischen Ausdruck eintragen. Sie dürfen alles von der Tastatur eingeben.

Nicht in der Tabelle vorhandene Variable

- ENTWEDER

vorher, also vor dem Schreiben der Formel, in die ACCESS TABLE einführen nach Klick auf:



Danach im Formelfeld mit Strg+Leerzeichen den Formelausdruck aufbauen.

- ODER

Falls Sie schon die Formel mit nicht eingetragenen Variablen geschrieben haben, dann empfiehlt es sich, die Variablen nacheinander nach Klick auf  in die ACCESS TABLE, also in die Variablen-tabelle wie folgt einzutragen:

- Variable in Formel markieren und mit Strg+X entfernen oder mit Strg+C kopieren.
- Klick auf 
- Standard-Text im Feld "NAME" ersetzen: Zunächst markieren und danach Strg+V betätigen
- Fehlende Daten wie Datentyp jetzt eintragen bzw. wählen.
- Falls mit Strg+X entfernt, dann mit Klick auf  übertragen. Falls mit Strg+C kopiert, einfach  anklicken.

## A.206

### Vergleichsfunktion

Das Vergleichen zweier Größen geschieht in der Reihenfolge

Ist-Größe VGL Soll-Größe.

VGL sind =, <>, >=, <=, >, <

Das Ja-Nein-Ergebnis wird einer Booleschen Variablen übergeben, die als Prozeßvariable (Condition) im PRAP einzutragen ist.

Mehr hierzu in den Nichtbinären Funktionen im Kapitel Entwurfsprozeß.

## A.207

Vollständig (complete) ist ein PRAP, wenn alle zu einer Operation gehörenden Übergangspfade genau irgendeiner anderen Operation zugeordnet werden können, alle Haltepfade führen zu der betrachteten Operation zurück. Das ist die pragmatische Beschreibung.

Oder die mathematische Form:

Wird jeder Pfad, der von einer Operation ausgeht, als konjunktiver Boolescher Ausdruck beschrieben, wobei ja mit 1 und nein mit 0 auszudrücken sind, dann ergibt die disjunktive Verknüpfung aller Konjunktionen den Wert 1. Die Vollständigkeit muß für jede Operation nachgewiesen werden.

## A.208

**Wertebelegung** ist eine Bezeichnung für eine Anzahl verschiedener Variabler und deren Werte. Die Anzahl ausgewählter Variablen erfüllt einen bestimmten Zweck und wird oft als Vektor (Dimension = Anzahl) aufgefaßt.

Im Falle der PRAP-Variablen wie Operations- oder Prozeß-Variable gibt es nur zwei Werte. Daher auch die Bezeichnung **Binärvektor**.

Prozeßvariable haben die Werte ja und nein, sie sind im PRAP sichtbar an den abgehenden Kanten (exits yes and no) angebracht.

Operationsvariable haben Werte wie ein / aus, auf / zu (on / off, open / close), aber auch jede andere Aussage und deren Negation ist denkbar: Ventil öffnen (Open the Valve) und Ventil nicht öffnen (Don't open the Valve), Tor schließen (Shut the Gate) und Tor öffnen (Open the Gate), Motor einschalten (Switch on the Motor) und Motor ausschalten (Switch off the Motor), Beleuchtung einschalten und Beleuchtung nicht einschalten, Beleuchtung ausschalten und Beleuchtung nicht ausschalten. Entscheidend ist das zu steuernde Stellglied, allgemein "der Empfänger des Befehls", wie die Wertebelegung technologisch interpretiert wird.

Sichtbar, welche Wertung in einer Operation gilt, ist die Belegung im Operations-Variablen-Fenster. Dort ist jede Operationsvariable mit ihrer Aussage eingetragen. Gilt diese in der betreffenden Operation, so ist sie markiert; ansonsten gilt die negierte Aussage.

## A.209

Widerspruchsfrei (contradictionless)

ist ein PRAP, wenn sich alle zu einer Operation gehörenden Übergangs- und Haltepfade gegenseitig ausschließen.

## A.210

Das Wort Zustand betrifft hier einen Automatenzustand. Jede Prozeß-Situation wird in einen Zustand überführt, manche Prozeß-Situationen vereinigt unter einen Zustand.

Zustands-Merker und symbolische Merker:

Der Prozeßablaufplan ist ein automatentheoretisch fundierter, zustandsorientierter Graph.

Das Compilat muß z.B. zur Markierung der Zustände Merkerbit zugewiesen bekommen. SPAS generiert automatisch Boolesche Variable, die der Nutzer bei Betrachtung des Compilates betrachten kann.

## A.211

Zeit- und Zählglieder = Timer and Counter

Zur Steuerung von technologischen Prozessen finden auch Hilfsfunktionen Anwendung. Dazu gehören

Zeit- und Zählfunktionen, die häufig Prozeßgrößen insofern simulieren, als sie ein bestimmtes Ereignis nach einer zu erwartenden Zeit bzw. nach einer Anzahl irgendwelcher Hilfsereignisse anzeigen. Unter dem Thema Zeit- und Zählglieder im Kapitel Entwurfsprozeß erfahren Sie mehr hierzu.

## A.212 Project Member Function

- Vorgehensweise
  - Funktion als Member in Ressource einfügen [siehe Seite 154] und im Formblatt IDENTIFICATION und DESCRIPTION ausfüllen
  - Berechnungen editieren:  
Formblatt schließen (OK-Button), Funktion im Projektbaum markieren, RECHTE MAUSTASTE | Edit Function anklicken. Nun Kopieren [siehe Seite 37] und allgemeine Berechnungen [siehe Seite 37] eintragen. Verwenden Sie die Hilfe-angebote!
- Allgemeines

Eine Funktion ist in SPAS ein Hilfsmittel, bestimmte Zusammenhänge mit arithmetischen Ausdrücken [siehe Seite 138] herstellen zu können, die keinen PRAP benötigen, weil jeder der Ausdrücke keine sequenzielle Struktur hat sondern ein Textformat (strukturierter Text nach IEC 1131-3) ohne Eigenschaften eines Ablauf-/Zustands-Verhaltens wie beim PRAP. Es stehen die gleichen Möglichkeiten zur Verfügung wie im PRAP mit der Funktion "Calculate". Es sind im **Formblatt** nur zwei Angaben einzutragen, die IDENTIFICATION, also ein Kurzbezeichner für den Code, den der Precompiler als Function\_Block aufbaut und einen ausführlicheren Text, DESCRIPTION.
- IDENTIFICATION

Prägen Sie hierfür eine wenige Zeichen [siehe Seite 172] erforderliche Bezeichnung, den Namen.
- DESCRIPTION

Schreiben Sie hier einen ausführlicheren Text [siehe Seite 143], aber keine lange Beschreibung sondern einen technologisch relevanten Bezeichner.
- PARAMETER

Die Variablen werden nur angezeigt. Zutreffend für die Funktion sind nur die FORMAL-Parameter, die beim ersten Einfügen noch nicht existieren. Sie müssen anschließend die Funktion als Text im Editor eingeben.

## A.213 Import einer Excel-Tabelle mit Teilprozeß-Variablen



### Anmerkung

Der Import einer Variablenliste ist grundsätzlich sowohl in eine Ressource als auch in ein Programm(Typ) möglich.

Es gibt KEINEN Export in eine Excelliste!

Legen Sie für die Ressource und jedes Programm eine Variablenliste an, ansonsten haben Sie nur Zusatzarbeit mit dem Variablenhandling. Die Variablenlisten können in ein und derselben Datei, aber sollten in unterschiedlichen Blättern enthalten sein oder wenigstens in unterschiedlichen Zeilenbereichen.



**Tip**

Der Import mehrfach auftretender Bezeichner unter "NAME" erfolgt ohne weitere Kennzeichnung. Vermeiden Sie den Mehrfachimport durch Einklammern des Namens. Beispiel: "Motor ein" bei erstem Auftreten und "(Motor ein)" bei Wiederholungen. Praktisch ist es auch, bei Koppelvariablen den "Output" zu importieren und alle "Input" einzuklammern. Änderungen an Variablen gegenüber dem Listeneintrag am besten im betreffenden Member, also PROGRAM oder RESOURCE, erledigen. Änderungen am Statement nach Klick auf "View|Variables"

**Hinweise zum Vorgehen:** Im Laufe der Prozeßzerlegung erhalten Sie Teilprozesse [siehe Seite 191] und deren zugehörige Eingangs- und Ausgangs-Variablen als Namen bzw. Bezeichner [siehe Seite 172] oft sogar auch deren Anschlußadressen [siehe Seite 123] durch das Verdrahtungsprojekt, auch Busadressen sind damit gemeint. Als Beispiel diene die Liste zum Lehrbeispiel Lampen- und Stern-Dreieckschaltung: Liste nach PZG<sup>40</sup>

Wenn Sie sich eine Exceltabelle anlegen können, dann brauchen Sie viele Handgriffe nur einmal zu tun und außerdem haben Sie eine "Variablensammlung" immer als Dokument dabei.

Die Zusammenstellung der Variablen für ein Programm darf jede Variable, d.h. jede Adresse nur einmal enthalten. Zwei Listenformen werden empfohlen,

- zum einen die nach Teilprozessen orientierte Liste [siehe Seite 201], in der die Variablen wie im PZG (siehe oben) sind,
- zum anderen die nach Adressen orientierte Liste [siehe Seite 202], in der die Variablen in einer z.B. "aufsteigenden" Adressen-Folge geordnet sind. In dieser Form könnte ein Verdrahtungsprogramm die Tabelle liefern.



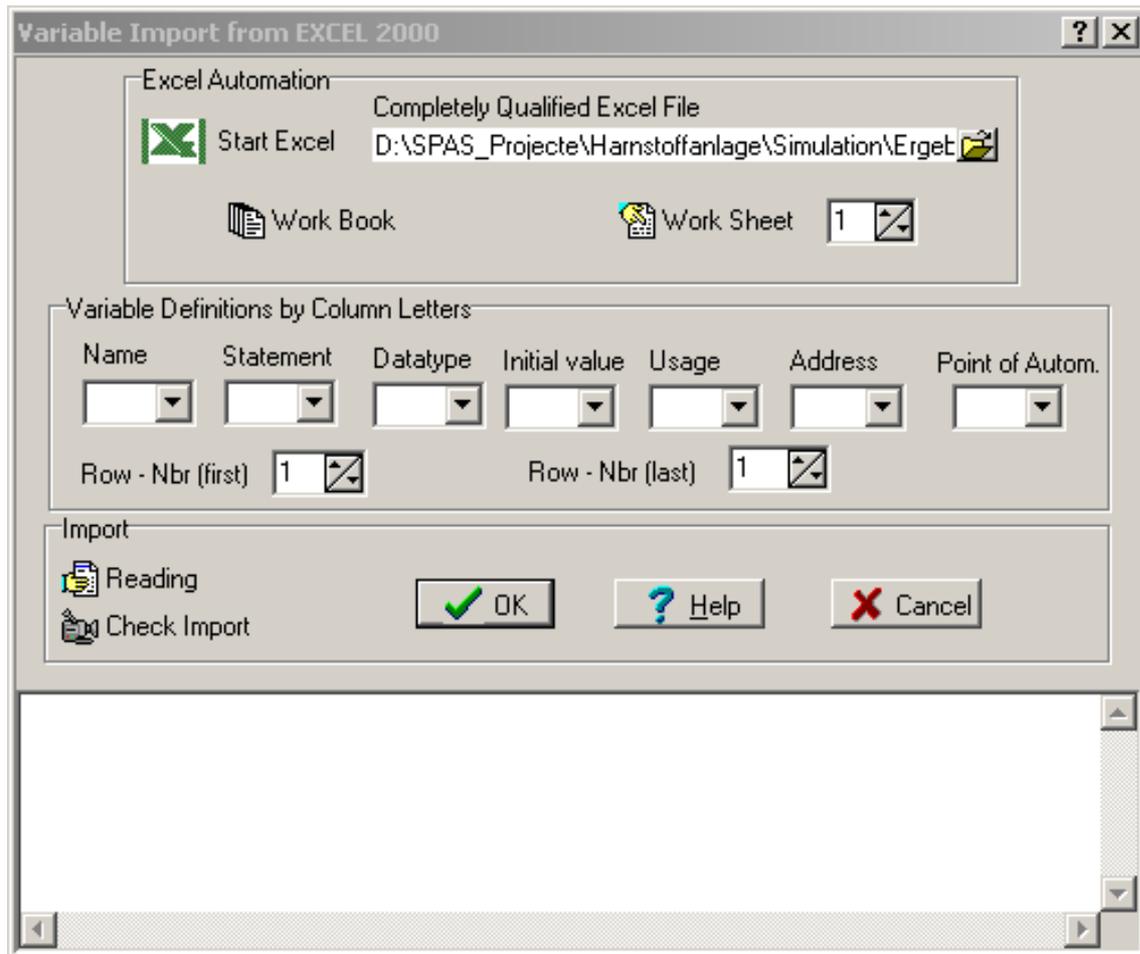
**Anmerkung**

Beachten Sie die Schreibweise der Namen [siehe Seite 172], Datentypen [siehe Seite 147] und Verwendung (Usage) [siehe Seite 195] Klicken Sie zu Ihrer Unterstützung auf die Stichworte.

Das nachfolgende Formblatt soll nun erläutert werden:

40

|    | A               | B                                 | C      | D     | E        | F              | G          | H                     | I         | J         | K              | L      | M                                    |
|----|-----------------|-----------------------------------|--------|-------|----------|----------------|------------|-----------------------|-----------|-----------|----------------|--------|--------------------------------------|
| 1  | Act. Name       | Actual Statement                  | Prefix | Addr. | Qt. Type | USAGE          | Init. Val. | Point of Automation   | TP Name   | PRAP Name | Form. Param.   | USAGE  | Formal Comment                       |
| 2  | Halltor_geoeffn | Halltor geöffnet                  | E      | 10    | BOOL     | Address Input  | GS 0308    |                       | Innenbel  | Lampe     | Lx_einschalten | Input  | Schließbedingung erfüllt             |
| 3  | Halle_beleucht  | Halle beleuchtet                  | A      | 10    | BOOL     | Address Output | EU 9302    |                       | Innenbel  | Lampe     | Lx_EIN         | Output | Lampe einschalten                    |
| 4  |                 |                                   |        |       |          |                |            |                       |           |           |                |        |                                      |
| 5  | Halltor_geschl  | Halltor geschlossen               | E      | 00    | BOOL     | Address Input  | GS 0608    |                       | Aussenbel | Lampe     | Lx_einschalten | Input  | Schließbedingung erfüllt             |
| 6  | Hallenruf_bel   | Hallenrufkabel betätigen          | A      | 00    | BOOL     | Address Output | EU 9602    |                       | Aussenbel | Lampe     | Lx_EIN         | Output | Lampe einschalten                    |
| 7  |                 |                                   |        |       |          |                |            |                       |           |           |                |        |                                      |
| 8  | Tor_oeffn       | Tor öffnen                        | M      | 10    | BOOL     | Address Memory |            | KDP von Koordinierung | Tor_AUF   | St_Dr_Anl | Mot_ein        | Input  | Motor einschalten                    |
| 9  | Ferngest_AUF    | Öffnen Ferngesteuert              | M      | 10    | BOOL     | Address Memory |            | KDP von Koordinierung | Tor_AUF   | St_Dr_Anl | FgS            | Input  | Ferngesteuert schalten               |
| 10 | YHd_oeffn       | Öffnen von Hand                   | E      | 12    | BOOL     | Address Input  | HS 0506    |                       | Tor_AUF   | St_Dr_Anl | YHd_Mot_ein    | Input  | Motor von Hand einschalten           |
| 11 | NOT_Tast        | Schließen NOT-AUS                 | E      | 30    | BOOL     | Address Input  | HZ 0502    |                       | Tor_AUF   | St_Dr_Anl | NOTAUS         | Input  | NOT-Aus                              |
| 12 | (I_AUF)         | Schaltwert zum Öffnen erreicht    | M      | 00    | BOOL     | Address Memory |            | KDP von L_MOT_AUF     | Tor_AUF   | St_Dr_Anl | I_Schalt       | Input  | Strom hat Weicherschaltwert erreicht |
| 13 | Oeffn_Haupt     | Hauptschütz Öffnen EIN            | A      | 11    | BOOL     | Address Output | EU 0203    |                       | Tor_AUF   | St_Dr_Anl | S1             | Output | Hauptschütz einschalten              |
| 14 | Oeffn_Stern     | Sternschütz Öffnen EIN            | A      | 12    | BOOL     | Address Output | EU 0203    |                       | Tor_AUF   | St_Dr_Anl | S2             | Output | Sternschütz einschalten              |
| 15 | Oeffn_Drei      | Dreieckschütz Öffnen EIN          | A      | 13    | BOOL     | Address Output | EU 0203    |                       | Tor_AUF   | St_Dr_Anl | S3             | Output | Dreieckschütz einschalten            |
| 16 | Oeffn_AUS       | Öffnen ausgeschaltet              |        |       | BOOL     | Local Static   |            | KDP nach L_MOT_AUF    | Tor_AUF   | St_Dr_Anl | Mot_aus        | Output | Motor ausgeschaltet                  |
| 17 | T_Anl_AUF       | Anlauftimer Öffnen                | T1     |       | TON      | Address Memory |            |                       | Tor_AUF   | St_Dr_Anl | T_Anl          | Input  | Anlauf-TIMER                         |
| 18 |                 |                                   |        |       |          |                |            |                       |           |           |                |        |                                      |
| 19 | Tor_Schli       | Tor schließen                     | M      | 10    | BOOL     | Address Memory |            | KDP von Koordinierung | Tor_ZU    | St_Dr_Anl | Mot_ein        | Input  | Motor einschalten                    |
| 20 | Ferngest_ZU     | Schließen Ferngesteuert           | M      | 11    | BOOL     | Address Memory |            | KDP von Koordinierung | Tor_ZU    | St_Dr_Anl | FgS            | Input  | Ferngesteuert schalten               |
| 21 | YHd_Schli       | Schließen von Hand                | E      | 02    | BOOL     | Address Input  | HS 0508    |                       | Tor_ZU    | St_Dr_Anl | YHd_Mot_ein    | Input  | Motor von Hand einschalten           |
| 22 | (NOT_Tast)      | Schließen NOT-AUS                 | E      | 30    | BOOL     | Address Input  | HZ 0502    |                       | Tor_ZU    | St_Dr_Anl | NOTAUS         | Input  | NOT-Aus                              |
| 23 | (L_ZU)          | Schaltwert zum Schließen erreicht | M      | 01    | BOOL     | Address Memory |            | KDP von L_MOT_ZU      | Tor_ZU    | St_Dr_Anl | L_Schalt       | Input  | Strom hat Weicherschaltwert erreicht |
| 24 | Schli_Haupt     | Hauptschütz Schließen EIN         | A      | 01    | BOOL     | Address Output | EU 0901    |                       | Tor_ZU    | St_Dr_Anl | S1             | Output | Hauptschütz einschalten              |
| 25 | Schli_Stern     | Sternschütz Schließen EIN         | A      | 02    | BOOL     | Address Output | EU 0901    |                       | Tor_ZU    | St_Dr_Anl | S2             | Output | Sternschütz einschalten              |
| 26 | Schli_Drei      | Dreieckschütz Schließen EIN       | A      | 03    | BOOL     | Address Output | EU 0901    |                       | Tor_ZU    | St_Dr_Anl | S3             | Output | Dreieckschütz einschalten            |
| 27 | Schli_AUS       | Schließen ausgeschaltet           |        |       | BOOL     | Local Static   |            | KDP nach L_MOT_ZU     | Tor_ZU    | St_Dr_Anl | Mot_aus        | Output | Motor ausgeschaltet                  |
| 28 | T_Anl_ZU        | Anlauftimer Schließen             | T2     |       | TON      | Address Memory |            |                       | Tor_ZU    | St_Dr_Anl | T_Anl          | Input  | Anlauf-TIMER                         |
| 29 |                 |                                   |        |       |          |                |            |                       |           |           |                |        |                                      |
| 30 | (I_AUF)         | Schaltwert zum Öffnen erreicht    | M      | 00    | BOOL     | Address Memory |            | KDP nach Tor_AUF      | L_MOT_AUF | L_MOT     | L_Schalt       | Output | Weicherschaltwert erreicht           |
| 31 | (Oeffn_AUS)     | Öffnen ausgeschaltet              |        |       | BOOL     | Local Static   |            | KDP von Tor_AUF       | L_MOT_AUF | L_MOT     | Mot_aus        | Input  | Motor ausgeschaltet                  |
| 32 | (L_Soll_AUF)    | Soll-Stromwert beim Öffnen        |        |       | INT      | Constant       | 12345      |                       | L_MOT_AUF | L_MOT     | L_Soll         | Input  | Weicherschaltwert                    |
| 33 | (I_LIST_AUF)    | Ist-Stromwert beim Öffnen         | PEV252 |       | INT      | Address Input  | EF 0101    |                       | L_MOT_AUF | L_MOT     | L_LIST         | Input  | Motorstrom                           |
| 34 |                 |                                   |        |       |          |                |            |                       |           |           |                |        |                                      |
| 35 | (L_ZU)          | Schaltwert zum Schließen erreicht | M      | 01    | BOOL     | Address Memory |            | KDP nach Tor_ZU       | L_MOT_ZU  | L_MOT     | L_Schalt       | Output | Weicherschaltwert erreicht           |
| 36 | (Schli_AUS)     | Schließen ausgeschaltet           |        |       | BOOL     | Local Static   |            | KDP von Tor_ZU        | L_MOT_ZU  | L_MOT     | Mot_aus        | Input  | Motor ausgeschaltet                  |
| 37 | (L_Soll_ZU)     | Soll-Stromwert beim Schließen     |        |       | INT      | Constant       | 12333      |                       | L_MOT_ZU  | L_MOT     | L_SOLL         | Input  | Weicherschaltwert                    |
| 38 | (I_LIST_ZU)     | Ist-Stromwert beim Schließen      | PEV250 |       | INT      | Address Input  | EF 0201    |                       | L_MOT_ZU  | L_MOT     | L_LIST         | Input  | Motorstrom                           |
| 39 |                 |                                   |        |       |          |                |            |                       |           |           |                |        |                                      |



- "Start Excel"  
dient dazu, dieses Programm eigenständig zu starten, damit Sie Ihre Tabelle sichtbar und bearbeitbar machen können.
- Im Feld "Completely Qualified Excel File"  
wird der Ordner "Shared Repository" angezeigt. Sie müssen eventuell den Ordner wechseln, Ihre Variablen-Datei auswählen und öffnen.
- Klick auf "Work Book"  
Zeigt die Anzahl der Blätter an, die Sie in Excel unter "Extras|Optionen|Allgemein - Blätter in neuer Arbeitsmappe" angegeben haben. Wenn Sie das nicht getan haben, dann sind generell drei Blätter (Work sheets) angelegt.
- "Work Sheet" so einstellen, daß Sie Ihre Variablen-tabelle importieren und nichts anderes der Excel-datei. Anschließend Klick auf "Work Sheet". Eine Anzeige folgt nur im Fehlerfall!
- "Variable Definitions by Column Letters" müssen jetzt ausgefüllt werden, wobei Sie die Spalten Ihrer Tabelle eingeben. Sie erkennen die Freizügigkeit bezüglich der Spalten-Inhalte! Legen Sie also nach Ihrer Vorstellung die Tabelle an.



### Tipp

Falls Sie z.B. keine Initialwerte eintragen wollen, dann brauchen Sie diese Spalte nicht anzulegen. Geben Sie einen Buchstaben an, der außerhalb der Liste liegt, also z.B. "K"

Als Beispiel-Liste<sup>41</sup> sehen Sie die zum Lehrbeispiel. Sie ist nach Instanzen geordnet, siehe Spalte F, "TP-Name".

- Die "Row Nbr"  
first und last dienen zum Bestimmen eines Listenausschnittes. Einstellen NICHT VERGESSEN!!  
In der Beispiel-Liste ist first immer 2, damit die Titelzeile nicht gelesen werden kann.
- Die Vorbereitung zum "Import"  
geschieht durch Klick auf "Reading"
- Klick auf "Check Import"  
zeigt Ihnen im unteren Feld das Ergebnis, wobei die Daten durch Komma getrennt sind.  
Leerzeilen enthalten nur Kommata. Beachten Sie den Text zur Beispieldatei. [siehe Seite 201]
- Klick auf "OK"  
beendet die gesamte Aktion und importiert endgültig in PROGRAM bzw. RESOURCE.
- Klick auf "Cancel" unterbindet den Import, daher können Sie jederzeit "Probe-Importe" durchführen.

## A.214 Excel-Liste, nach Teilprozessen geordnet

Beim Aufbau der Variablenliste ist zu beachten, daß Zeilen, in denen unter "Name" nichts eingetragen ist, oder der Name in Klammern steht, z.B. (I\_AUF auf Zeile 12), nicht gelesen werden. Sie erscheinen als Leerzeile.

Damit folglich jede Variable nur einmal gelesen wird, sind die Namen der Eingangs- oder Koppelvariablen eingeklammert, die wiederholt auftreten.

Ausgangsvariable sind sowieso einmalig, es sei denn, Sie verwenden eine Ausgangs- auch als

41

|    | A              | B                                 | C      | D     | E        | F              | G                    | H                   | I         | J              | K            | L                                  | M              |
|----|----------------|-----------------------------------|--------|-------|----------|----------------|----------------------|---------------------|-----------|----------------|--------------|------------------------------------|----------------|
|    | Act. Name      | Actual Statement                  | Prefix | Addr. | Qt. Type | USAGE          | Init. Val.           | Point of Automation | TP Name   | PRAP Name      | Form. Param. | USAGE                              | Formal Comment |
| 1  | Halltor_groewh | Hallentor geöffnet                | E      | 10    | BOOL     | Address Input  | GS 0308              | Innenbel            | Lampe     | Lx_einschalten | Input        | Schließbedingung erfüllt           |                |
| 2  | Halle_beleucht | Halle beleuchtet                  | A      | 10    | BOOL     | Address Output | EU 9302              | Innenbel            | Lampe     | Lx_EIN         | Output       | Lampe einschalten                  |                |
| 3  |                |                                   |        |       |          |                |                      |                     |           |                |              |                                    |                |
| 4  | Halltor_geschl | Hallentor geschlossen             | E      | 00    | BOOL     | Address Input  | GS 0608              | Aussenbel           | Lampe     | Lx_einschalten | Input        | Schließbedingung erfüllt           |                |
| 5  | Hallenruf_bel  | Hallenrufaktiv beleuchten         | A      | 00    | BOOL     | Address Output | EU 9602              | Aussenbel           | Lampe     | Lx_EIN         | Output       | Lampe einschalten                  |                |
| 6  |                |                                   |        |       |          |                |                      |                     |           |                |              |                                    |                |
| 7  |                |                                   |        |       |          |                |                      |                     |           |                |              |                                    |                |
| 8  | Tor_oefwh      | Tor öffnen                        | M      | 10    | BOOL     | Address Memory | KDP von Koordination | Tor_AUF             | St_Dr_Anl | Mot_ein        | Input        | Motor einschalten                  |                |
| 9  | ferngest_AUF   | Öffnen ferngesteuert              | M      | 10    | BOOL     | Address Memory | KDP von Koordination | Tor_AUF             | St_Dr_Anl | FgS            | Input        | Ferngesteuert schalten             |                |
| 10 | YHd_oefwh      | Öffnen von Hand                   | E      | 12    | BOOL     | Address Input  | HS 0206              | Tor_AUF             | St_Dr_Anl | YHd_Mot_ein    | Input        | Motor von Hand einschalten         |                |
| 11 | NOT_Tast       | Öffnen NOT-AUS                    | E      | 30    | BOOL     | Address Input  | H2 0502              | Tor_AUF             | St_Dr_Anl | NOTAUS         | Input        | NOT-Aus                            |                |
| 12 | (I_AUF)        | Schaltwert zum Öffnen erreicht    | M      | 00    | BOOL     | Address Memory | KDP von I_MDT_AUF    | Tor_AUF             | St_Dr_Anl | I_Schalt       | Input        | Strom hat Weichschaltwert erreicht |                |
| 13 | Oefwh_Haupt    | Hauptschütz Öffnen EIN            | A      | 11    | BOOL     | Address Output | EU 9203              | Tor_AUF             | St_Dr_Anl | S1             | Output       | Hauptschütz einschalten            |                |
| 14 | Oefwh_Stern    | Sternschütz Öffnen EIN            | A      | 12    | BOOL     | Address Output | EU 9203              | Tor_AUF             | St_Dr_Anl | S2             | Output       | Sternschütz einschalten            |                |
| 15 | Oefwh_Drei     | Dreieckschütz Öffnen EIN          | A      | 13    | BOOL     | Address Output | EU 9203              | Tor_AUF             | St_Dr_Anl | S3             | Output       | Dreieckschütz einschalten          |                |
| 16 | Oefwh_AUS      | Öffnen ausgeschaltet              |        |       | BOOL     | Local Status   | KDP nach I_MDT_AUF   | Tor_AUF             | St_Dr_Anl | Mot_aus        | Output       | Motor ausgeschaltet                |                |
| 17 | T_Anl_AUF      | AnlaufTIMER Öffnen                | T1     |       | TON      | Address Memory |                      | Tor_AUF             | St_Dr_Anl | T_Anl          | Input        | Anlauf-TIMER                       |                |
| 18 |                |                                   |        |       |          |                |                      |                     |           |                |              |                                    |                |
| 19 | Tor_Schli      | Tor schließen                     | M      | 10    | BOOL     | Address Memory | KDP von Koordination | Tor_ZU              | St_Dr_Anl | Mot_ein        | Input        | Motor einschalten                  |                |
| 20 | ferngest_ZU    | Schließen ferngesteuert           | M      | 11    | BOOL     | Address Memory | KDP von Koordination | Tor_ZU              | St_Dr_Anl | FgS            | Input        | Ferngesteuert schalten             |                |
| 21 | YHd_Schli      | Schließen von Hand                | E      | 02    | BOOL     | Address Input  | HS 0508              | Tor_ZU              | St_Dr_Anl | YHd_Mot_ein    | Input        | Motor von Hand einschalten         |                |
| 22 | (NOT_Tast)     | Schließen NOT-AUS                 | E      | 30    | BOOL     | Address Input  | H2 0502              | Tor_ZU              | St_Dr_Anl | NOTAUS         | Input        | NOT-Aus                            |                |
| 23 | (L_ZU)         | Schaltwert zum Schließen erreicht | M      | 01    | BOOL     | Address Memory | KDP von I_MDT_ZU     | Tor_ZU              | St_Dr_Anl | I_Schalt       | Input        | Strom hat Weichschaltwert erreicht |                |
| 24 | Schli_Haupt    | Hauptschütz Schließen EIN         | A      | 01    | BOOL     | Address Output | EU 9501              | Tor_ZU              | St_Dr_Anl | S1             | Output       | Hauptschütz einschalten            |                |
| 25 | Schli_Stern    | Sternschütz Schließen EIN         | A      | 02    | BOOL     | Address Output | EU 9501              | Tor_ZU              | St_Dr_Anl | S2             | Output       | Sternschütz einschalten            |                |
| 26 | Schli_Drei     | Dreieckschütz Schließen EIN       | A      | 03    | BOOL     | Address Output | EU 9501              | Tor_ZU              | St_Dr_Anl | S3             | Output       | Dreieckschütz einschalten          |                |
| 27 | Schli_AUS      | Schließen ausgeschaltet           |        |       | BOOL     | Local Status   | KDP nach I_MDT_ZU    | Tor_ZU              | St_Dr_Anl | Mot_aus        | Output       | Motor ausgeschaltet                |                |
| 28 | T_Anl_ZU       | AnlaufTIMER Schließen             | T2     |       | TON      | Address Memory |                      | Tor_ZU              | St_Dr_Anl | T_Anl          | Input        | Anlauf-TIMER                       |                |
| 29 |                |                                   |        |       |          |                |                      |                     |           |                |              |                                    |                |
| 30 | I_AUF          | Schaltwert zum Öffnen erreicht    | M      | 00    | BOOL     | Address Memory | KDP nach Tor_AUF     | I_MDT_AUF           | I_MDT     | I_Schalt       | Output       | Weichschaltwert erreicht           |                |
| 31 | (Oefwh_AUS)    | Öffnen ausgeschaltet              |        |       | BOOL     | Local Status   | KDP von Tor_AUF      | I_MDT_AUF           | I_MDT     | Mot_aus        | Input        | Motor ausgeschaltet                |                |
| 32 | I_Soll_AUF     | Soll-Stromwert beim Öffnen        |        |       | INT      | Constant       | 12345                | I_MDT_AUF           | I_MDT     | I_Soll         | Input        | Weichschaltwert                    |                |
| 33 | I_LIST_AUF     | Ist-Stromwert beim Öffnen         | PEV252 |       | INT      | Address Input  | EF 0101              | I_MDT_AUF           | I_MDT     | I_LIST         | Input        | Motorstrom                         |                |
| 34 |                |                                   |        |       |          |                |                      |                     |           |                |              |                                    |                |
| 35 | L_ZU           | Schaltwert zum Schließen erreicht | M      | 01    | BOOL     | Address Memory | KDP nach Tor_ZU      | I_MDT_ZU            | I_MDT     | I_Schalt       | Output       | Weichschaltwert erreicht           |                |
| 36 | (Schli_AUS)    | Schließen ausgeschaltet           |        |       | BOOL     | Local Status   | KDP von Tor_ZU       | I_MDT_ZU            | I_MDT     | Mot_aus        | Input        | Motor ausgeschaltet                |                |
| 37 | I_Soll_ZU      | Soll-Stromwert beim Schließen     |        |       | INT      | Constant       | 12333                | I_MDT_ZU            | I_MDT     | I_Soll         | Input        | Weichschaltwert                    |                |
| 38 | I_LIST_ZU      | Ist-Stromwert beim Schließen      | PEV256 |       | INT      | Address Input  | EF 0201              | I_MDT_ZU            | I_MDT     | I_LIST         | Input        | Motorstrom                         |                |
| 39 |                |                                   |        |       |          |                |                      |                     |           |                |              |                                    |                |

|    | A               | B                                 | C      | D     | E             | F              | G          | H                     | I         | J              | K            | L      | M                                   |
|----|-----------------|-----------------------------------|--------|-------|---------------|----------------|------------|-----------------------|-----------|----------------|--------------|--------|-------------------------------------|
|    | Act. Name       | Actual Statement                  | Prefix | Addr. | Dr. Type      | USAGE          | Init. Val. | Point of Automation   | TP Name   | PRAP Name      | Form. Param. | USAGE  | Formal Comment                      |
| 1  | Halltor_geoeffn | Hallentor geöffnet                | E      | 10    | BOOL          | Address Input  | 05 0301    | Innenbeleuchtung      | Lampe     | La_einschalten | La_EIN       | Input  | Schaltbedingung erfüllt             |
| 2  | Halle_beleucht  | Halle beleuchtet                  | A      | 10    | BOOL          | Address Output | EU 0302    | Innenbeleuchtung      | Lampe     | La_einschalten | La_EIN       | Output | Lampe einschalten                   |
| 3  |                 |                                   |        |       |               |                |            |                       |           |                |              |        |                                     |
| 4  | Halltor_geschl  | Hallentor geschlossen             | E      | 03    | BOOL          | Address Input  | 05 0601    | Aussenbeleuchtung     | Lampe     | La_einschalten | La_EIN       | Input  | Schaltbedingung erfüllt             |
| 5  | Hallenour_bel   | Hallenourakt beleuchten           | A      | 03    | BOOL          | Address Output | EU 0602    | Aussenbeleuchtung     | Lampe     | La_einschalten | La_EIN       | Output | Lampe einschalten                   |
| 6  |                 |                                   |        |       |               |                |            |                       |           |                |              |        |                                     |
| 7  |                 |                                   |        |       |               |                |            |                       |           |                |              |        |                                     |
| 8  | Tor_oeffn       | Tor öffnen                        | M      | 00    | BOOL          | Address Memory |            | KDP von Koordinierung | Tor_AUF   | St_Dr_Anl      | Mot_ein      | Input  | Motor einschalten                   |
| 9  | ferngest_AUF    | Öffnen ferngesteuert              | M      | 10    | BOOL          | Address Memory |            | KDP von Koordinierung | Tor_AUF   | St_Dr_Anl      | FgS          | Input  | Ferngesteuert schalten              |
| 10 | VH4_oeffn       | Öffnen von Hand                   | E      | 12    | BOOL          | Address Input  | HS 0201    |                       | Tor_AUF   | St_Dr_Anl      | VH4_Mot_ein  | Input  | Motor von Hand einschalten          |
| 11 | NOT_Tast        | Öffnen NOT-AUS                    | E      | 38    | BOOL          | Address Input  | H2 0202    |                       | Tor_AUF   | St_Dr_Anl      | NOTAUS       | Input  | NOT-Aus                             |
| 12 | (L_AUF)         | Schaltwert zum Öffnen erreicht    | M      | 00    | BOOL          | Address Memory |            | KDP von L_MOT_AUF     | Tor_AUF   | St_Dr_Anl      | L_Schalt     | Input  | Strom hat Vektorschaltwert erreicht |
| 13 | Oeffn_Haupt     | Hauptschütz Öffnen EIN            | A      | 11    | BOOL          | Address Output | EU 0203    |                       | Tor_AUF   | St_Dr_Anl      | S1           | Output | Hauptschütz einschalten             |
| 14 | Oeffn_Stern     | Sternschütz Öffnen EIN            | A      | 12    | BOOL          | Address Output | EU 0203    |                       | Tor_AUF   | St_Dr_Anl      | S2           | Output | Sternschütz einschalten             |
| 15 | Oeffn_Drei      | Dreieckschütz Öffnen EIN          | A      | 13    | BOOL          | Address Output | EU 0203    |                       | Tor_AUF   | St_Dr_Anl      | S3           | Output | Dreieckschütz einschalten           |
| 16 | Oeffn_AUS       | Öffnen ausgeschaltet              |        |       | BOOL          | Local Static   |            | KDP nach L_MOT_AUF    | Tor_AUF   | St_Dr_Anl      | Mot_aus      | Output | Motor ausgeschaltet                 |
| 17 | T_Anl_AUF       | Anlauftimer Öffnen                | T1     |       | TON           | Address Memory |            |                       | Tor_AUF   | St_Dr_Anl      | T_Anl        | Input  | Anlauf-TIMER                        |
| 18 |                 |                                   |        |       |               |                |            |                       |           |                |              |        |                                     |
| 19 | Tor_Schli       | Tor schließen                     | M      | 01    | BOOL          | Address Memory |            | KDP von Koordinierung | Tor_ZU    | St_Dr_Anl      | Mot_ein      | Input  | Motor einschalten                   |
| 20 | ferngest_ZU     | Schließen ferngesteuert           | M      | 11    | BOOL          | Address Memory |            | KDP von Koordinierung | Tor_ZU    | St_Dr_Anl      | FgS          | Input  | Ferngesteuert schalten              |
| 21 | VH4_Schli       | Schließen von Hand                | E      | 02    | BOOL          | Address Input  | HS 0501    |                       | Tor_ZU    | St_Dr_Anl      | VH4_Mot_ein  | Input  | Motor von Hand einschalten          |
| 22 | (NOT_Tast)      | Schließen NOT-AUS                 | E      | 39    | BOOL          | Address Input  | H2 0502    |                       | Tor_ZU    | St_Dr_Anl      | NOTAUS       | Input  | NOT-Aus                             |
| 23 | (L_ZU)          | Schaltwert zum Schließen erreicht | M      | 01    | BOOL          | Address Memory |            | KDP von L_MOT_ZU      | Tor_ZU    | St_Dr_Anl      | L_Schalt     | Input  | Strom hat Vektorschaltwert erreicht |
| 24 | Schli_Haupt     | Hauptschütz Schließen EIN         | A      | 01    | BOOL          | Address Output | EU 0501    |                       | Tor_ZU    | St_Dr_Anl      | S1           | Output | Hauptschütz einschalten             |
| 25 | Schli_Stern     | Sternschütz Schließen EIN         | A      | 02    | BOOL          | Address Output | EU 0501    |                       | Tor_ZU    | St_Dr_Anl      | S2           | Output | Sternschütz einschalten             |
| 26 | Schli_Drei      | Dreieckschütz Schließen EIN       | A      | 03    | BOOL          | Address Output | EU 0501    |                       | Tor_ZU    | St_Dr_Anl      | S3           | Output | Dreieckschütz einschalten           |
| 27 | Schli_AUS       | Schließen ausgeschaltet           |        |       | BOOL          | Local Static   |            | KDP nach L_MOT_ZU     | Tor_ZU    | St_Dr_Anl      | Mot_aus      | Output | Motor ausgeschaltet                 |
| 28 | T_Anl_ZU        | Anlauftimer Schließen             | T2     |       | TON           | Address Memory |            |                       | Tor_ZU    | St_Dr_Anl      | T_Anl        | Input  | Anlauf-TIMER                        |
| 29 |                 |                                   |        |       |               |                |            |                       |           |                |              |        |                                     |
| 30 | L_AUF           | Schaltwert zum Öffnen erreicht    | M      | 00    | BOOL          | Address Memory |            | KDP nach Tor_AUF      | L_MOT_AUF | L_MOT          | L_Schalt     | Output | Vektorschaltwert erreicht           |
| 31 | (Oeffn_AUS)     | Öffnen ausgeschaltet              |        |       | BOOL          | Local Static   |            | KDP von Tor_AUF       | L_MOT_AUF | L_MOT          | Mot_aus      | Input  | Motor ausgeschaltet                 |
| 32 | L_Soll_AUF      | Soll-Stromwert beim Öffnen        | INT    |       | Constant      |                | 12345      |                       | L_MOT_AUF | L_MOT          | L_Soll       | Input  | Vektorschaltwert                    |
| 33 | L_Ist_AUF       | Ist-Stromwert beim Öffnen         | PEV252 | INT   | Address Input |                | EF 0301    |                       | L_MOT_AUF | L_MOT          | L_Ist        | Input  | Motorstrom                          |
| 34 |                 |                                   |        |       |               |                |            |                       |           |                |              |        |                                     |
| 35 | L_ZU            | Schaltwert zum Schließen erreicht | M      | 01    | BOOL          | Address Memory |            | KDP nach Tor_ZU       | L_MOT_ZU  | L_MOT          | L_Schalt     | Output | Vektorschaltwert erreicht           |
| 36 | (Schli_AUS)     | Schließen ausgeschaltet           |        |       | BOOL          | Local Static   |            | KDP von Tor_ZU        | L_MOT_ZU  | L_MOT          | Mot_aus      | Input  | Motor ausgeschaltet                 |
| 37 | L_Soll_ZU       | Soll-Stromwert beim Schließen     | INT    |       | Constant      |                | 12333      |                       | L_MOT_ZU  | L_MOT          | L_Soll       | Input  | Vektorschaltwert                    |
| 38 | L_Ist_ZU        | Ist-Stromwert beim Schließen      | PEV250 | INT   | Address Input |                | EF 0201    |                       | L_MOT_ZU  | L_MOT          | L_Ist        | Input  | Motorstrom                          |
| 39 |                 |                                   |        |       |               |                |            |                       |           |                |              |        |                                     |

## A.215 Excel-Liste, nach Adressen geordnet

Beim Aufbau der Variablenliste ist zu beachten, daß Zeilen, in denen unter "Name" nichts eingetragen ist, nicht gelesen werden. Sie erscheinen als Leerzeile.

|    | A               | B                                 | C      | D     | E             | F              | G          | H                 | I                 | J              | K            | L      | M                          |
|----|-----------------|-----------------------------------|--------|-------|---------------|----------------|------------|-------------------|-------------------|----------------|--------------|--------|----------------------------|
|    | Act. Name       | Actual Statement                  | Prefix | Addr. | Data Type     | Usage          | Init. Val. | Point of Automati | TP Names          | PRAP Name      | Formal Para. | Usage  | Formal Comment             |
| 1  | Halltor_geschl  | Hallentor geschlossen             | E      | 03    | BOOL          | Address Input  | 05 0301    | Aussenbeleuchtung | Lampe             | La_einschalten | La_EIN       | Input  | Schaltbedingung erfüllt    |
| 2  | VH4_Schli       | Schließen von Hand                | E      | 02    | BOOL          | Address Input  | HS 0501    |                   | Tor_ZU            | St_Dr_Anl      | VH4_Mot_ein  | Input  | Motor von Hand einschalten |
| 3  |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 4  | Halltor_geoeffn | Hallentor geöffnet                | E      | 10    | BOOL          | Address Input  | 05 0301    | Innenbeleuchtung  | Lampe             | La_einschalten | La_EIN       | Input  | Schaltbedingung erfüllt    |
| 5  |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 6  |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 7  | VH4_oeffn       | Öffnen von Hand                   | E      | 12    | BOOL          | Address Input  | HS 0201    |                   | Tor_AUF           | St_Dr_Anl      | VH4_Mot_ein  | Input  | Motor von Hand einschalten |
| 8  |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 9  |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 10 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 11 | NOT_Tast        | Öffnen NOT-AUS                    | E      | 38    | BOOL          | Address Input  | H2 0202    |                   | Tor_AUF_Tor_ZU    | St_Dr_Anl      | NOTAUS       | Input  | NOT-Aus                    |
| 12 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 13 | Hallenour_bel   | Hallenourakt beleuchten           | A      | 03    | BOOL          | Address Output | EU 0602    | Aussenbeleuchtung | Lampe             | La_einschalten | La_EIN       | Output | Lampe einschalten          |
| 14 | Schli_Haupt     | Hauptschütz Schließen EIN         | A      | 01    | BOOL          | Address Output | EU 0501    |                   | Tor_ZU            | St_Dr_Anl      | S1           | Output | Hauptschütz einschalten    |
| 15 | Schli_Stern     | Sternschütz Schließen EIN         | A      | 02    | BOOL          | Address Output | EU 0501    |                   | Tor_ZU            | St_Dr_Anl      | S2           | Output | Sternschütz einschalten    |
| 16 | Schli_Drei      | Dreieckschütz Schließen EIN       | A      | 03    | BOOL          | Address Output | EU 0501    |                   | Tor_ZU            | St_Dr_Anl      | S3           | Output | Dreieckschütz einschalten  |
| 17 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 18 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 19 | Halle_beleucht  | Halle beleuchten                  | A      | 10    | BOOL          | Address Output | EU 0302    | Innenbeleuchtung  | Lampe             | La_einschalten | La_EIN       | Output | Lampe einschalten          |
| 20 | Oeffn_Haupt     | Hauptschütz Öffnen EIN            | A      | 11    | BOOL          | Address Output | EU 0203    |                   | Tor_AUF           | St_Dr_Anl      | S1           | Output | Hauptschütz einschalten    |
| 21 | Oeffn_Stern     | Sternschütz Öffnen EIN            | A      | 12    | BOOL          | Address Output | EU 0203    |                   | Tor_AUF           | St_Dr_Anl      | S2           | Output | Sternschütz einschalten    |
| 22 | Oeffn_Drei      | Dreieckschütz Öffnen EIN          | A      | 13    | BOOL          | Address Output | EU 0203    |                   | Tor_AUF           | St_Dr_Anl      | S3           | Output | Dreieckschütz einschalten  |
| 23 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 24 | L_AUF           | Schaltwert zum Öffnen erreicht    | M      | 00    | BOOL          | Address Memory |            | KDP TP1 nach TP2  | L_MOT_AUF_Tor_AUF | L_MOT          | L_Schalt     | Output | Vektorschaltwert erreicht  |
| 25 | L_ZU            | Schaltwert zum Schließen erreicht | M      | 01    | BOOL          | Address Memory |            | KDP TP4 nach TP5  | L_MOT_ZU_Tor_ZU   | L_MOT          | L_Schalt     | Output | Vektorschaltwert erreicht  |
| 26 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 27 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 28 | Oeffn_AUS       | Öffnen ausgeschaltet              |        |       | BOOL          | Local Static   |            | KDP TP2 nach TP1  | Tor_AUF_L_MOT_AUF | St_Dr_Anl      | Mot_aus      | Output | Motor ausgeschaltet        |
| 29 | Schli_AUS       | Schließen ausgeschaltet           |        |       | BOOL          | Local Static   |            | KDP TP5 nach TP4  | Tor_ZU_L_MOT_ZU   | St_Dr_Anl      | Mot_aus      | Output | Motor ausgeschaltet        |
| 30 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 31 | ferngest_AUF    | Öffnen ferngesteuert              | M      | 10    | BOOL          | Address Memory |            | KDP TP0 nach TP2  | Koordin_Tor_AUF   | St_Dr_Anl      | FgS          | Input  | Ferngesteuert schalten     |
| 32 | ferngest_ZU     | Schließen ferngesteuert           | M      | 11    | BOOL          | Address Memory |            | KDP TP0 nach TP5  | Koordin_Tor_ZU    | St_Dr_Anl      | FgS          | Input  | Ferngesteuert schalten     |
| 33 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 34 | Tor_oeffn       | Tor öffnen                        | M      | 00    | BOOL          | Address Memory |            | KDP TP0 nach TP2  | Koordin_Tor_AUF   | St_Dr_Anl      | Mot_ein      | Input  | Motor einschalten          |
| 35 | Tor_Schli       | Tor schließen                     | M      | 01    | BOOL          | Address Memory |            | KDP TP0 nach TP5  | Koordin_Tor_ZU    | St_Dr_Anl      | Mot_ein      | Input  | Motor einschalten          |
| 36 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 37 | L_Ist_ZU        | Ist-Stromwert beim Schließen      | PEV250 | INT   | Address Input |                | EF 0301    |                   | L_MOT_ZU          | L_MOT          | L_Ist        | Input  | Motorstrom                 |
| 38 | L_Ist_AUF       | Ist-Stromwert beim Öffnen         | PEV252 | INT   | Address Input |                | EF 0301    |                   | L_MOT_AUF         | L_MOT          | L_Ist        | Input  | Motorstrom                 |
| 39 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |
| 40 | T_Anl_AUF       | Anlauftimer Öffnen                | T1     |       | TON           | Address Memory |            |                   | Tor_AUF           | St_Dr_Anl      | T_Anl        | Input  | Anlauf-Timer               |
| 41 | T_Anl_ZU        | Anlauftimer Schließen             | T2     |       | TON           | Address Memory |            |                   | Tor_ZU            | St_Dr_Anl      | T_Anl        | Input  | Anlauf-Timer               |
| 42 |                 |                                   |        |       |               |                |            |                   |                   |                |              |        |                            |

## A.216 Task-gesteuerter Timer

Neben den Standard-Funktionsbausteinen für TIMER, wie in SPAS der Typ TON verwendet wird, kann auch ein zyklischer Aufruf einer Addierfunktion für Zeitwerte bis zu 596,5 Stunden bzw. reichlich 24 Tagen genutzt werden.

Die Addierfunktion,

z.B. Zeitwert:= Zeitwert + Taktzeit;

kann entweder als FUNCTION oder eingebettet in einen PRAP geschrieben werden. Der Datentyp für beide Größen ist TIME [siehe Seite 147], USAGE für "Zeitwert" ist Input/Output und für "Taktzeit" Constant. Die Taktzeit muß gleich der zyklischen TASK sein, die speziell für die Zeitwert-Funktion einzurichten ist.

Natürlich muß ein PROGRAM angelegt werden, das die Zeitfunktion im Rahmen einer FUNCTION oder

des PRAP aufruft, wobei auch dem formalen Parameter "Zeitwert" ein direkt adressierter oder lokaler Speicherplatz zugewiesen wird, den Sie am besten in der Ressource deklarieren, weil Sie dann von allen Programmen darauf zugreifen können.

Zwei weitere Hinweise sind noch wichtig:

- Sie können in einem beliebigen PRAP oder einer FUNCTION auf die Variable des Zeitwertes zugreifen. Projektieren Sie dazu die Vergleichs-Funktion COMPARISON in einer von Ihnen einzurichtenden Bedingung / Prozeßvariable.
- Nutzen Sie irgendwo im Projekt eine Möglichkeit aus, um der Variablen "Zeitwert" wieder den Wert Null zuzuweisen. Dies kann z.B. nach dem Ende eines Programmdurchlaufes sein, wenn dieser kürzer als 24,85 Tage = 596,4 Stunden ist.

#### BEISPIEL:

Sie arbeiten mit STEP 7 und benötigen Zeiten zwischen 4 und 240 Stunden. Die Genauigkeit ist unkritisch, so nehmen Sie den OB 30 mit 5s Taktzeit und Priorität 10, weil es einen OB mit einer größeren Taktzeit nicht gibt. Also projektieren Sie in der Zeitwert-Funktion die Taktzeit = 5s.

## A.217 Zeitglied-Projektierung - Timer Configuration

Das Grundwissen zum Timer-Entwurf lesen Sie bitte bei Zeitglied - Timer [siehe Seite 26] nach.

Die Konfigurierung eines Zeitgliedes/Timers enthält vier Größen, von denen drei in den folgenden Punkten aufgeführt sind:

### • **Timer**

ist das durch die Operationsvariable je nach Operationenfolge ein- und ausgeschaltete Zeitglied. Es handelt sich laut IEC 1131-3 um einen Standardfunktions-Baustein.

Die Felder Name, Statement [siehe Seite 191] und "Variable Usage" bilden die Aussage über Bezeichnung und Art des Timers.

"Variable Usage" ist eine Aussage über den Speicherort des Timers. Schreibt der Hersteller der Software [siehe Seite 7] nichts anderes vor, sollte USAGE = "Local Static" sein, falls der Timer nicht als Variable (mit Aktualparameter-Übergabe im Instanzbaustein) gehandhabt wird.

Der "Data Type" ist mit "TON" festgelegt und darf nicht geändert werden! Streng genommen ist dies der Typ des Standard-Funktionsblocks.

### • **Timer Value**

Die Felder Name, Statement und Variable Usage [siehe Seite 191] bestimmen die Eigenschaften der Größe Timer Value = Zeitwert. Geben Sie zuerst "Name" ein, danach alles andere:

- Der Datentyp ist "TIME" und darf nicht verändert werden.
- Der Initialwert, also der Anfangswert der Zeitdauer ist einzutragen oder das Feld leer lassen. (AccessInitialValue [siehe Seite 137])

- **Elapsed Time** ist der momentane Zeitwert und als Variable vom Datentyp TIME verfügbar, also in der Timer-Konfiguration nicht weiter zu beachten. Für Anzeige, d.h. Ausgabe der aktuellen Zeit nutzen sie die Funktion Copy??? und für Vergleichsfunktionen [siehe Seite 204] verwenden Sie als Istwert ebenfalls diesen Timer-Ausgang in der Form Name.ET, wobei ".ET" nur im Formelausdruck einzutragen ist!

**Timer Output Q** kann nur zugehörig zu einer P-Variablen bearbeitet werden. Schließen Sie also die Bearbeitung der Timer-Konfiguration ab und wenden sich dem Timer-Ausgang zu. Wählen Sie bei Choose a Function:

"Timer Output Q".

## A.218 Vergleicherefunktionen - Comparison Configuration

### Allgemeines

Im Thema Vergleicherefunktion - Comparison [siehe Seite 34] finden Sie Grundsätzliches beschrieben. Wichtig ist,

- daß Sie möglichst nur bereits deklarierte Variable verwenden (siehe: Eingabehilfe [siehe Seite 196]).
- die Symbole  und  richtig bedienen. (siehe CompareFormula [siehe Seite 144])
- daß auf beiden Seiten der Vergleichereoperation auch arithmetische Ausdrücke stehen dürfen, so z.B.  $a < b + c$
- daß mehrere Vergleichere-Ausdrücke mit AND, OR oder XOR zu einer Formel verbunden werden dürfen.
- daß den Vergleichere-Ausdruck, also die gesamte Formel, durch ";" abgeschlossen wird.

Sie können von hier aus zu den Erklärungen der Felder gelangen:

- Name [siehe Seite 172]
- Statement [siehe Seite 172]
- Variable Usage [siehe Seite 144]
- Formula [siehe Seite 144]

## A.219 Zählglied-Projektierung - Counter Configuration

Das Grundwissen zum Counter-Entwurf lesen Sie bitte bei Zählglied - Counter [siehe Seite 29] nach.

Beachten Sie die Hinweise zu Zählern in den wählbaren Sprachen. [siehe Seite 7]

Die Konfigurierung eines Zählgliedes/Counters enthält sechs Größen, von denen fünf in den folgenden vier Punkten aufgeführt sind:

- **Counter**  
ist das durch die Operationsvariable je nach Operationenfolge ein- und ausgeschaltete Zählglied. Es handelt sich laut IEC 1131-3 um einen Standardfunktions-Baustein.  
Die Felder Name und Statement [siehe Seite 146] bilden die Aussage über die Art des Counters. "Variable Usage" ist eine Aussage über den Speicherort des Zählers. [siehe Seite 7] und ist abhängig von der gewählten Sprache, im allgemeinen "Local Static", wenn der Counter, das Zeitglied, nicht als Variable ( mit Aktualparameter-Übergabe im Instanzbaustein) gehandhabt wird.  
Der Datentyp, "Data Type", ist mit CTUD festgelegt (Counter Up Down). Der Precompiler setzt den richtigen Typen ein durch Ermittlung, welche Zähl-Signale es gibt.
- **Counter Value**  
Die Felder Name, Statement und Variable Usage [siehe Seite 146] bestimmen die Eigenschaften der Größe "Counter Value" = Zählwert. Geben Sie zuerst "Name" ein, danach alles andere. Der Datentyp ist sprachabhängig, im allgemeinen "INT".
- **Die Zählgänge V und R** können sowohl Operationsvariable als auch Prozeßvariable sein. Wählen Sie bei Choose a Function:  
"Count Up / Count Down".
- **Current Value** ist der laufende Zählerwert und kann als Variable vom Datentyp INT verwendet

werden. Dazu nutzen Sie bitte die Kopier- oder Arithmetikfunktion [siehe Seite 205], die Sie in der betreffenden Operation projektieren, wo der aktuelle Zählerinhalt benötigt wird. Tragen Sie als Operand den "NAME" des Zählers ein und ergänzen ihn nur in dem Formelausdruck mit ".CV".

In Vergleichsfunktionen [siehe Seite 204] verfahren Sie bitte ebenso und verwenden "NAME.CV" als ISTWERT.

**Counter Output Q** kann nur zugehörig zur P-Variablen bearbeitet werden. Schließen Sie also die Bearbeitung der Counter-Konfiguration ab und wenden sich dem Counter-Ausgang Q zu. Wählen Sie bei Choose a Function:

"Counter Output Q".

## A.220 Flankenbewertung - Raising and Falling Edges Detection

### Allgemeines

Im Thema Flanken-Funktionen [siehe Seite 36] finden Sie Grundsätzliches beschrieben. Sie gehört laut IEC 1131-3 zu den Standard-Funktionsbausteinen. Allerdings gibt es hier keine Wahl bezüglich Speicherort (Usage) der FB. Und Datentypen gibt es auch keine auszuwählen. Daher sind nur die Eingangs- und die Ausgangsvariable zu projektieren.

Wählen Sie im Feld "Name" bzw. aus der Tabelle diejenige Variable aus, deren Flankenbewertung erforderlich ist, sie wird auch "Primary Variable" genannt. Falls keine Variable zutrifft, dann führen Sie eine neue mit Klick auf  ein.

Beachten Sie die richtige Bedienung der Symbole  und . Nur mit dem letztgenannten erzielen Sie eine Übernahme der Daten sowohl in die Datenbank als auch in den Funktionenzusammenhang der Flankenbewertung, mit dem "POST-HAKEN" wird nur in die Datenbank neu eingetragen.

### Primary Variable:

Die Eingangsvariable [siehe Seite 150], deren Flanke(n) ausgewertet werden soll(en), bitte nach Klick auf   eintragen oder auswählen.



### Anmerkung

Wollen Sie die Primary Variable gegen eine andere auswechseln - nicht einfach ändern - dann müssen Sie **zuerst das Formelobjekt löschen**.

## A.221 Kopier- und arithmetische Funktionen - Calculate(multiple) und Calculate(single)

### Allgemeines

- Kopier- und arithmetische Funktionen sind hier vereinigt, weil das Zuweisungszeichen "[:=" einer Kopierfunktion entspricht.  
Zur Eingabe der Formeln siehe Abschnitt "Wichtige Hinweise zum Formulieren ..." weiter unten.
- Im Fall einer arithmetischen Funktion erfolgt die Zuweisung des Ergebnisses an die links stehende Variable, anderenfalls die Zuweisung des Wertes der rechts - allein - stehenden Variablen.
- Calculate(multiple) und Calculate(single)  
unterscheiden sich grundsätzlich durch die Häufigkeit der Abarbeitung während einer **stabilen** Situation [siehe Seite 183], in der die Berechnung durchgeführt wird:
  - Es bedeutet "multiple" die normale Abarbeitung der Berechnungsfunktion, d.h. bei jedem Arbeitszyklus der Ressource (z.B. SPS) innerhalb der Situation wird berechnet.

- Es bedeutet "single" eine einmalige Abarbeitung der Berechnungsfunktion, d.h. nur im ersten Arbeitszyklus der Ressource innerhalb der Situation wird berechnet.

In den Themen (aus dem Buch "Entwurfsprozeß") Kopierfunktion - Copy [siehe Seite 37] und Arithmetik-Funktionen [siehe Seite 37] finden Sie Grundsätzliches beschrieben.

### **Wichtige Hinweise zum Formulieren der Ausdrücke**

Denken Sie daran,

- daß Sie nur bereits deklarierte Variable verwenden (siehe: Eingabehilfe für Formelausdrücke [siehe Seite 196]).
- daß der Ergebnisspeicher mit seinem DATENTYP das zu erwartende Ergebnis auch aufnehmen kann. Die rechts vom Zuweisungszeichen stehenden Operanden werden weitgehend<sup>42</sup> automatisch angepaßt.  
Für die Kopierfunktion gilt die automatische Datentyp-Anpassung in vorgegebenen Grenzen [siehe Seite 148]. Damit ist sie nutzbar für die "Umrechnung" eines Datentypen.
- daß Arithmetik-Ausdrücke, die den Ergebnisspeicher enthalten, nur mit "Calculate(single)" sinnvolle Ergebnisse bringen. Z.B. sollte  $A:=A+1$  mit "single" programmiert werden, weil sonst A letztendlich die Anzahl der Arbeitszyklen mit enthält. Sollte bei  $B:=A+1$  in B ein Teil von A enthalten sein, gilt dies ebenso; denn B könnte ein DINT- und A ein INT-Datentyp sein, wobei A der niederwertige Teil von B ist.  
Es gibt allerdings Fälle, in denen dieser Effekt praktisch genutzt werden kann!!
- die Symbole für "POST" und "ACCEPT" richtig bedienen: WICHTIG [siehe Seite 153]



### **Anmerkung**

Momentanwerte der Zeit- und Zählglieder sind mit "Name".ET (Datentyp TIME) bzw. "Name".CV (Datentyp INT) in den Formelausdruck einzutragen.

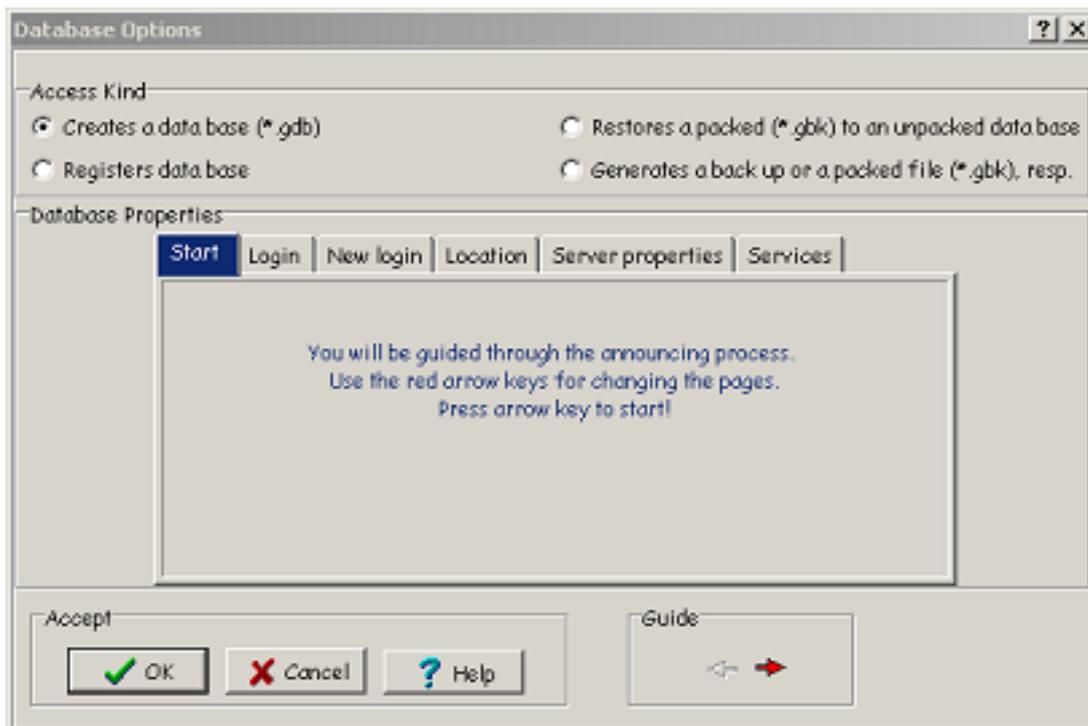
Sie können von hier aus zu den Erklärungen der Felder gelangen:

- Name [siehe Seite 172]
- Statement [siehe Seite 172]
- Variable Usage [siehe Seite 145]
- Formula [siehe Seite 145]

## **A.222 Database Options**

Das Formular "Database Options" bietet grundlegende Funktionen bezüglich des Zugriffs auf die Datenbank.

<sup>42</sup> Die Standardfunktionen der Arithmetik sind in der IEC 1131-3 bezüglich Datentypen so gefordert, daß sie automatisch alle anderen auf den "größten" Datentyp anpassen. [siehe Seite 148] Falls also der Ergebnis-Datentyp REAL sein sollte und alle Eingänge z.B. INT, steht die Datentypwandlung INT->DINT->REAL automatisch in dem Formelausdruck.



Im oberen Feld "Access Kind" ist die Markierung im linken oberen Feld voreingestellt für Neuanlegen (siehe erster Punkt) einer Datenbank (\*.gdb) in einem von Ihnen bereitgestellten Ordner/Verzeichnis. Wenn Sie eine vorhandene Datenbank Registrieren wollen (siehe zweiter Punkt), ist die Markierung dort durch Klick anzubringen.

Eine Datenbank in stark komprimierter Form als "Back Up"(siehe vierter Punkt) speichern bedeutet, rechts unten zu markieren.

Ein Back Up (siehe dritter Punkt) wieder verwenden wollen ("Restore"), muß durch Markierung rechts oben erfolgen.



### Anmerkung

Auf der START-Seite ist geschrieben, daß Sie unbedingt mit Klick auf die Pfeile  und  operieren sollten, um in der richtigen Reihenfolge die Eintragungen vornehmen zu können. Ein Klick auf die Blätter öffnet sie zwar, aber sie sind nicht aktiviert.

Nur in einem blau unterlegten Blatt-Titel sind auch Eintragungen oder Änderungen möglich.

- **Beim Anlegen eines neuen Projektes**

müssen Sie der Reihe nach einige Formblätter ausfüllen. Der Datenbank-Name und der Projekt-Name sind später NICHT ÄNDERBAR.

- "Login" ist für ein neues Projekt nicht unbedingt neu zu wählen. Mit einem neuen Login sollte auch ein zugehöriges Passwort [siehe Seite 177] einzutragen sein. Falls Sie ein vorhandenes Login verwenden wollen, dann klicken Sie sich vorwärt bis zum Blatt "Server Properties" und erhalten mit Klick auf  alle Login-Bezeichner angezeigt(zwischendurch nochmal mit OK bestätigen). Durch Markieren, Kopieren und "zurück-klicken" bis "Login" ist durch Einfügen in "Database Login" die Übernahme erledigt. Allerdings müssen Sie das Passwort dafür wissen!
- "New Login" ist nur wichtig, wenn Sie in "Login" einen neuen Text eingetragen haben. Dann sollten Sie sich mit Vor- und Zunamen eintragen. "First Name" und "Last Name" sind Ihrer freien Wahl überlassen und nicht signifikant für den Zugang zum Projekt, also zur Datenbank.

- **"Location"**  
Mit Klick auf  suchen Sie das Verzeichnis, in das Ihr neues Projekt abgelegt werden soll.  
"Database Name" ist durch Sie bei einem neuen Projekt einzutragen. Keine Umlaute oder ß verwenden!  
"Projekt Name" ist die Projektbezeichnung. Wählen Sie hier einen technologisch relevanten Bezeichner.
- **"Server Properties"**  
bietet Ihnen die Wahl des lokalen Servers oder über das Netz erreichbare, abgesetzte (remote) zu nutzen.
- **"OK"** button click sorgt für das Anlegen einer Datenbank. Die entsprechenden Meldungen können Sie im Sheet "Compiler Notes" oder unterhalb der Arbeitsfläche in der "Message Box" nachlesen. Anschließend Öffnen des Projektes mit Klick auf !
- **Das Registrieren eines vorhandenen Projektes**  
Bezeichnet wird ein Projekt als "vorhanden", wenn sich eine Projektdatei (-name-.gdb) in einem Ordner befindet, die Sie erstmalig öffnen wollen.
  - **"Login"**  
Hier müssen Sie wissen, welche Worte bzw. Zeichenketten einzugeben sind.
  - **"New Login"** ist nur wichtig, wenn Sie in "Login" einen neuen Text eingetragen haben. Dann sollten Sie sich mit Vor- und Zunamen eintragen. "First Name" und "Last Name" sind Ihrer freien Wahl überlassen und nicht signifikant für den Zugang zum Projekt, also zur Datenbank. Ein neuer Text für das "Data base login" liegt vor, wenn er nicht bei "Server properties" mit Klick auf  aufgeführt wird. (Mit Pfeiltasten vor- und zurück!)
  - **"Location"**  
Öffnen Sie nach Klick auf  das Verzeichnis der Projektdatenbank. Danach  anklicken und die im Verzeichnis zutreffende Datei \*.gdb markieren.  
Geben Sie einen "Project Name" ein. Sie dürfen mit jeder neuen Registrierung einen anderen Projektnamen eintragen.
  - **"Server Properties"**  
Stellen Sie hier den zutreffenden Server ein.
  - **"OK"-Button Klick**  
Anschließend können Sie das Projekt nach Klick auf  öffnen.
- **Restore a packed data base - Aus einer gepackten Projektdatei \*.gbk wieder eine verwendbare \*.gdb herstellen.**
  - **"Location"**  
Zunächst ab Blatt Start bis Location "vorklicken", dort nach Klick auf  das Verzeichnis wählen und darunter nach Klick auf  das File \*.gbk.
  - **"Server Properties"**  
Stellen Sie hier den zutreffenden Server ein.
  - **"Services"**  
Nach Klick auf  öffnen. Kontrollieren Sie, daß die richtige Datei gewählt wurde! Danach mit

Klick auf  den Ablauf starten.

- "OK"-Button Klick  
Anschließend können Sie das Projekt nach Klick auf  öffnen.
- **Generate a back up - Aus einer Projektdatei \*.gdb eine gepackte \*.gbk herstellen.**
  - "Location"  
Nach Klick auf  das Verzeichnis wählen und darunter nach Klick auf  das File \*.gdb.
  - "Server Properties"  
Stellen Sie hier den zutreffenden Server ein.
  - "Services"  
Kontrollieren Sie, daß die richtige Datei gewählt wurde! Danach mit Klick auf  den Ablauf starten.  
Abschluß mit Klick auf "OK". Die datei wird im gleichen Vezeichnis/Ordner abgelegt wie das Original.



### Tipp

Falls gewünscht, ändern Sie den Namen der gbk-Datei mit Hilfe des Explorers.

Im folgenden sind noch die Hinweis-Texte zu den Editierfeldern aufgelistet:

- Login [siehe Seite 158]
- New Login [siehe Seite 160]
- Location [siehe Seite 156]
- Server Properties [siehe Seite 163]
- Services [siehe Seite 163]

Falls Sie das Einführungs-Beispiel suchen, dann hier klicken. [siehe Seite 73]

## A.223 Instanz strukturierter Datentypen

Die Instanzierung einer unter der Ressource deklarierten formalen Datenstruktur erfolgt zunächst durch Einfügen einer Strukturinstanz  in ein PROGRAM. Geben Sie einen Namen und Text für Description ein und wählen Sie, falls mehr als ein Strukturtyp existiert, den zutreffenden im Feld "TYPES". Im Bereich PARAMETER sehen Sie alle deklarierten Variablen.

Gehen Sie nun wie folgt vor:

- Memberdialog mit Strukturinstanz schließen (Klick auf OK).
- Öffnen des Memberdialoges mit dem PROGRAM und Klick auf .
- Klick auf  und Einschreiben des Namens (bei NAME)der Strukturinstanz, den Sie soeben vergeben haben.

- Geben Sie nach dem Namen einen Punkt ein, dann sehen Sie alle deklarierten Variablen. Wählen Sie die betreffende aus.
- Danach bei Parameter "actual" und bei "Usage" das Passende wählen:  
Falls eine Adresse bei "ADDRESS" erforderlich ist, dann Address Input, Output oder Memory. Falls keine Adresse, dann "Local Static"
- Zum Abschluß müssen Sie auf  klicken. Nun sehen Sie die neue Strukturvariable im Projektmember PROGRAM unter PARAMETER eingetragen:  
Strukturinstanzname.Variablenname
- Wiederholen Sie für jede Variable, die Sie verwenden wollen, diese Prozedur mit Klick auf .  
Sie müssen nicht alle Variablen auf diesem Wege instanzieren.
- Kontrollieren Sie am Ergebnis, ob keine Fehler vorhanden sind, indem Sie den Memberdialog mit PROGRAM beenden (Klick auf OK!) und den mit der Strukturinstanz eröffnen. Dort sehen Sie die zugeordneten Variablen komplett.

## A.224 Import von Typen-PRAP und anderen

Bereits mit dem PRAP-Editor entworfene PRAP können Sie importieren, indem einfach die beim Export entstandene Datei \*.txt geöffnet wird, die anderen (\*.prp und \*.mtx) werden automatisch mit verwendet.

Unbedingt nach der Auswahl zunächst diese Datei Einlesen, d.h. ein Klick auf  ist nötig.

Tragen Sie wenigstens in das Feld "NAME" einen Bezeichner [siehe Seite 172] ein; DESCRIPTION, also ein Kommentar [siehe Seite 143] kann auch später eingeschrieben werden.

## A.225 Export von Typen-PRAP

Einmal entworfene PRAP können für andere Konfigurationen, Ressourcen oder Projekte zur Verfügung gestellt werden, wenn sie exportiert werden. Dabei entstehen drei Text-Dateien mit den Erweiterungen .prp, .mtx und .txt (insgesamt \_VAR.txt). Sorgen Sie dafür, daß keine der Dateien verändert wird, ansonsten ist kein Import möglich.

Sie müssen den Speicherort, also den Pfad und Dateinamen angeben. Angezeigt wird der Pfad, den Sie unter Shared Repository [siehe Seite 210] eingegeben haben.

Aber Sie können auch den Speicherort ändern. Klicken Sie auf , wählen Sie das gewünschte Verzeichnis und tragen den Filename ein.

## A.226 Wahl der Sprache, Erläuterungen zur Projektgestaltung

Jede Ressource kann für eine Steuerungseinheit verschiedener Hersteller geplant werden. Wählen Sie daher aus dem Angebot der Sprachen die zutreffende aus.

Beachten Sie dabei einige Hinweise zu den wählbaren Sprachen [siehe Seite 7]

### Explanations

Das Feld dient der Eintragung von Bemerkungen und Hinweisen zum Aufbau des Teilprojektes der betreffenden Ressource.

### Shared Repository:

Das ist ein Verzeichnis, in dem Sie sowohl das Compilat ablegen können als auch andere Export-Produkte wie Datenbank oder PRAP. Eventuell müssen Sie ein neues anlegen.

**Anmerkung**

Beachten Sie unbedingt, dieses Verzeichnis als Unterverzeichnis der aktuellen Datenbank anzulegen.

Mit Klick auf  finden Sie das Verzeichnis der Datenbank (File|Open ..).

## A.227 Project Member - PROGRAM

- Allgemeines  
siehe PROGRAM einfügen, bearbeiten [siehe Seite 58]  
Ein PROGRAM, auch Typ-Programm, vereinigt Teilprozesse in Gestalt von Instanzen und enthält für diese Instanzen erforderliche Variable. Die Programm-Instanz [siehe Seite 212] ruft gemäß zugewiesener Task die hier aufgeführten Instanzen in der angegebenen Reihenfolge auf.
- Identification  
Setzen Sie hier eine Kurzbezeichnung [siehe Seite 172] ein, die in der Ressource einmalig ist.
- Description  
Bezeichnen Sie hier den technologischen Abschnitt, der durch die Instanzen (Teilprozesse) gesteuert wird. Beachten Sie dabei Comment [siehe Seite 143]
- PARAMETER  
Die globalen Variablen [siehe Seite 154] (PARAMETER) dienen auch der Kommunikation [siehe Seite 170] zwischen den Bausteinen, die zu "PROGRAM "gehören.

**Anmerkung**

Externe Variable liefern die Ressource und die Konfiguration (Konfiguration in dieser Version nicht), also dort deklarierte globale Variable.

Ein Klick auf  öffnet das Eingabefeld für in PROGRAM lokal deklarierte Variable, sie sind innerhalb des PROGRAM für die Instanzen zu deklarieren. [siehe Seite 213] Diese sind bei der Instanziierung [siehe Seite 212] den Variablen der Typen-PRAP zuzuordnen.

Wählen Sie zur Darstellung der diesem PROGRAM zugeordneten Variablen in 'Filter Options' "Member" und suchen Sie in 'Filter Values' das zutreffende PROGRAM aus. Danach Klick auf "Filter Off".

Fügen Sie mit  eine neue Variable hinzu. Nach dem Ausfüllen der Eigenschaften jeder Variablen **müssen Sie** auf  klicken, um die Speicherung dieser Variablen, zugeordnet zum Programm, in der Datenbank zu veranlassen. Ein Klick auf  ergibt nicht die Zuordnung!

**Warnung**

Vergessen Sie nicht, am Ende den OK-Button zu betätigen, sonst ist alle Mühe umsonst gewesen!  
"CANCEL" nimmt alle Änderungen zurück.

**Noch einige wichtige Hinweise:**

- Sie können auch die Variablen aus einer Excel-Liste importieren [siehe Seite 198].
- Ein Doppelklick auf einen 'NAME' unter 'PARAMETER' öffnet die Variablen-tabelle einschließlich der Auswahl der Variablen grün unterlegt.

- **ÄNDERUNGEN** sind wie folgt auszuführen:  
Doppelklick auf die Variable, danach Änderungen in der Variablen-tabelle ausführen und mit Klick auf  bestätigen.  
Die Änderung muß links unter 'PARAMETER' sichtbar sein, falls es Datentyp oder Usage betrifft.

## A.228 Instanzbildung: Zuordnung Formal- zu Aktualparameter

Die im Memberdialogfenster - links - angezeigten Formalparameter müssen Sie den im Programm [siehe Seite 213] deklarierten Aktualparametern zuordnen.



### Tip

Lassen Sie sich beim Suchen helfen. Klicken Sie auf  und markieren zunächst, was Sie suchen wollen. Dann geben Sie die ersten Zeichen des Suchbegriffes ein und klicken auf "Search". Sie erhalten alle Zeichenketten mit den Anfangszeichen in der Tabelle aufgelistet.

Da alle Variablen aufgelistet sind, lohnt eventuell die Filterung. Sie können mit "Filter Options" und Wahl bei "Filter Values" die Anzeige mit Klick auf "Filter OFF/ON" verändern.



### Anmerkung

Achten Sie auf die Zugehörigkeit einer Variablen (Anzeige im Feld "Member"), weil in verschiedenen Members gleiche Namen zugelassen sind. Die Zugehörigkeit muß immer das gleiche PROGRAM bzw. die RESSOURCE sein.

Gehen Sie wie folgt vor:

- Links in der PARAMETER-Tabelle die Formalvariable markieren (event. Hinweis, Variable sei nicht zugeordnet, mit Klick auf OK quittieren)
- Rechts in der Tabelle die Aktualvariable, die zugewiesen werden soll, markieren
- Mit einem Klick auf  erscheint in der linken Tabelle die ausgewählte Variable anstelle des "none". Dabei erfolgt die Anzeige mit <Member>.Variable. <Member> kann strukturierter Datentyp oder PROGRAM-Name sein.



### Tip

Sollte das falsch gewesen sein, dann Zeile markieren, rechte Maustaste und "disconnect" anklicken, dann erscheint wieder "none". Danach erneut zuweisen.

Wollen Sie mehrere aufeinanderfolgende Zeilen korrigieren, dann diese Zeilen markieren und "disconnect" anklicken.

HINWEIS: Fehlende Aktualparameter ergänzen Sie zuvor im entsprechenden PROGRAM bzw. in der RESSOURCE.

## A.229 Project Member - PROGRAM-Instance

- Allgemeines  
siehe auch: PROGRAM Instance, einfügen, bearbeiten [siehe Seite 60]  
Die Programm-Instanz "teilt dem Betriebssystem der SPS" mit, unter welchen zeitlichen oder

ereignisbedingten Vorschriften (Tasks) der Programmtyp abgearbeitet werden soll.

- Identification  
Tragen Sie hier eine Kurzbezeichnung ein, die besonderen Bedingungen [siehe Seite 172] genügen soll.
- Description  
Die Beschreibung darf ausführlicher sein, muß jedoch im Rahmen der Vorschrift [siehe Seite 143] bleiben.
- RELATIONS  
Es ist auf jeden Fall der Programm-Typ zu wählen. Danach erfolgt die Zuweisung einer Task, die bei der "zuständigen" Ressource deklariert wurde.
- PARAMETER  
Dieses Feld bleibt leer.

## A.230 Deklaration der Aktual-Parameter und Adressen in PROGRAM-TYPE



### Anmerkung

Vorbemerkung:

VAR\_EXTERNAL sind solche Variablen, die bereits in der Ressource deklariert sind, der PROGRAM untergeordnet ist. Sie müssen hier nochmal aufgenommen werden.

VAR\_GLOBAL sind Deklarationen aller innerhalb des PROGRAM erforderlichen Aktual-Parameter mit den zugehörigen direkten Adressen. Hierzu gehören alle Variablen, die der Kommunikation der Unter-Programme innerhalb des Programms, z.B. Koppelungen, dienen und nicht schon in der Ressource stehen!

Wenn Sie ändern müssen, dann finden Sie unten Hinweise...



Die Bedeutung der Datenbank-Symbole entnehmen Sie bitte den Hinweisen auf der unteren Zeile des Fensters.

Tragen Sie in NAME [siehe Seite 172] den Variablenbezeichner ein, der hier als Symbol für die anzugebende Hardwareadresse gilt.

Im Feld USAGE. [siehe Seite 195] erstreckt sich die Auswahl auf Constant, Address Input, Address Output, Address Memory (E., A., M. bzw. %I., %O., %M.), Local Static und Local Temporary.

In das Feld ADDRESS [siehe Seite 123] geben Sie nur noch die Ziffernfolge für die Adresse ein! Falls erforderlich, die Byte-Bit-Trennung nicht vergessen. Nur in einem Projekt für STEP 7 werden auch die Präfixe wie z.B. E,A,MB,PEW akzeptiert.

Im Feld DATATYPE [siehe Seite 147] wählen Sie den zutreffenden Typ. Übrigens muß dieser mit dem des Formalparameters übereinstimmen, wenn der Aktualparameter mit seiner Adresse bei der Instanzierung zugewiesen wird.

Bei Klick auf  erfolgt die Übernahme der bei "NAME" eingetragenen Zeichenfolge einschließlich Datentyp, Adresse und "USAGE".

Zum Schluß nicht vergessen auf "OK" zu klicken!!



### Tip

Da es hier keine "Statements" zu den "Names" gibt, klicken Sie nach dem "OK" auf "View|Variables" und tragen dort unter "Varlist Comment" ein. Das ist auch für alle Änderungen eine günstige Möglichkeit! (Filterfunktionen nutzen)

### Wichtiger Hinweis bei Änderungen

Das Änderungsprocedere ist nur in folgender Form möglich, weil Sie die zu ändernde Variable - einschließlich ihrer Eigenschaften - in dem Memberdialogfeld suchen müssen, wo diese Variable deklariert wurde.

Also NICHT unter NAME in der Variablen-tabelle suchen und ändern!! Sie erfahren aber, wenn die Variable markiert ist, mit rechtem Mausklick, wo sie deklariert ist.

Suchen Sie also im Memberdialogfeld und markieren die Variable. Grün unterlegt finden Sie die im Memberdialogfeld angeklickte Variable im Access Field, also der Variablen-tabelle. Nun können Sie alles ändern, auch den Namen und danach unbedingt auf  klicken. Sie können den Erfolg oder Mißerfolg links im Memberdialogfeld sehen.

## A.231 PRAP Drucken

Ein im Feld "Charts" dargestellter PRAP kann nach rechtem Mausklick und anschließendem Klick auf "Print" gedruckt werden.

Das Ausdrucken von Prozeßablaufplänen ist in der Version 1 noch etwas schwierig, weil die Druck-Grenzen auf dem Panel "Charts" noch nicht mit den wirklichen Grenzen übereinstimmen. Das macht sich besonders bei mehrseitigen PRAP bemerkbar. Dennoch ist am Ende das Ergebnis recht gut, wenn Sie etwas Geduld aufbringen.

Sie sollten wie folgt vorgehen:

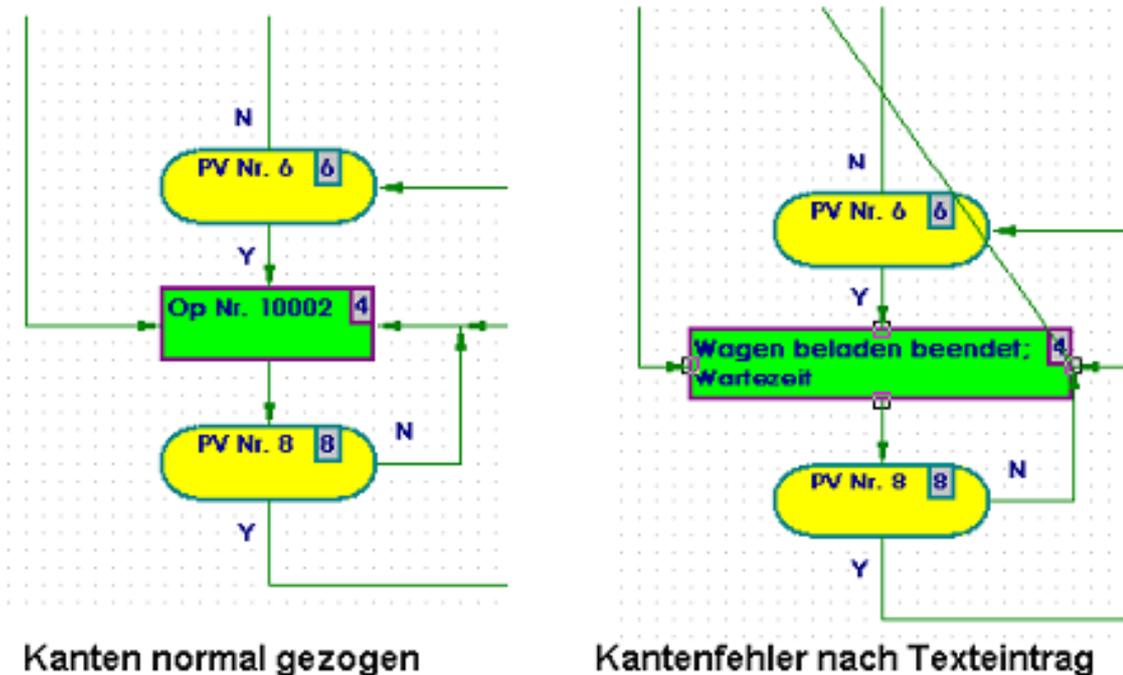
- Klicken Sie "Printer Select" und entscheiden vor allem, ob Sie Hochformat oder Querformat drucken wollen. Danach Klick auf "OK".
- Normalerweise sehen Sie nach dem OK auf dem Formblatt PRAP Print Preview linksseitig bei Hochformat ein ebensolches zunächst leeres Feld; bei Querformat entweder dasselbe oder ein "halbseitig" leeres Blatt.
- Die Einstellung der Resolution<sup>43</sup> zwischen 300 und maximalen Pixel/inch sollte bei 300 beginnen. Angezeigt wird die max.Pixel-Zahl des Druckers. Ferner ist die Schriftgröße - Text Size - wichtig, weil sich danach die Größe der Operations und Conditions richtet. Klicken Sie auf "Preview", eventuell auch mehrfach in kurzen Abständen, um einen Eindruck vom Druckbild zu erhalten.
- Nur bei klar erkennbarer Struktur des PRAP sind die Einstellungen sinnvoll.
- PRAP, die mehrere Seiten benötigen, müssen Sie so auf die Blätter verteilen, daß eine sinnvolle Trennung der Struktur erfolgt. Da die Darstellung der Seitengrenzen im Panel "Charts" noch mangelhaft ist, müssen Sie durch wiederholtes Verschieben der PRAP-Teile die günstigste Darstellung im Druckbild ermitteln. Die Anzahl der Druckseiten ist nicht immer richtig, häufig wird eine Leerseite am Ende gedruckt.

## A.232 Probleme beim Kanten zeichnen

<sup>43</sup> Die Größe des Bildes ist am kleinsten bei der maximalen Pixelzahl > 300, weil auf dem Panel "Charts" die Auflösung 300 ist.

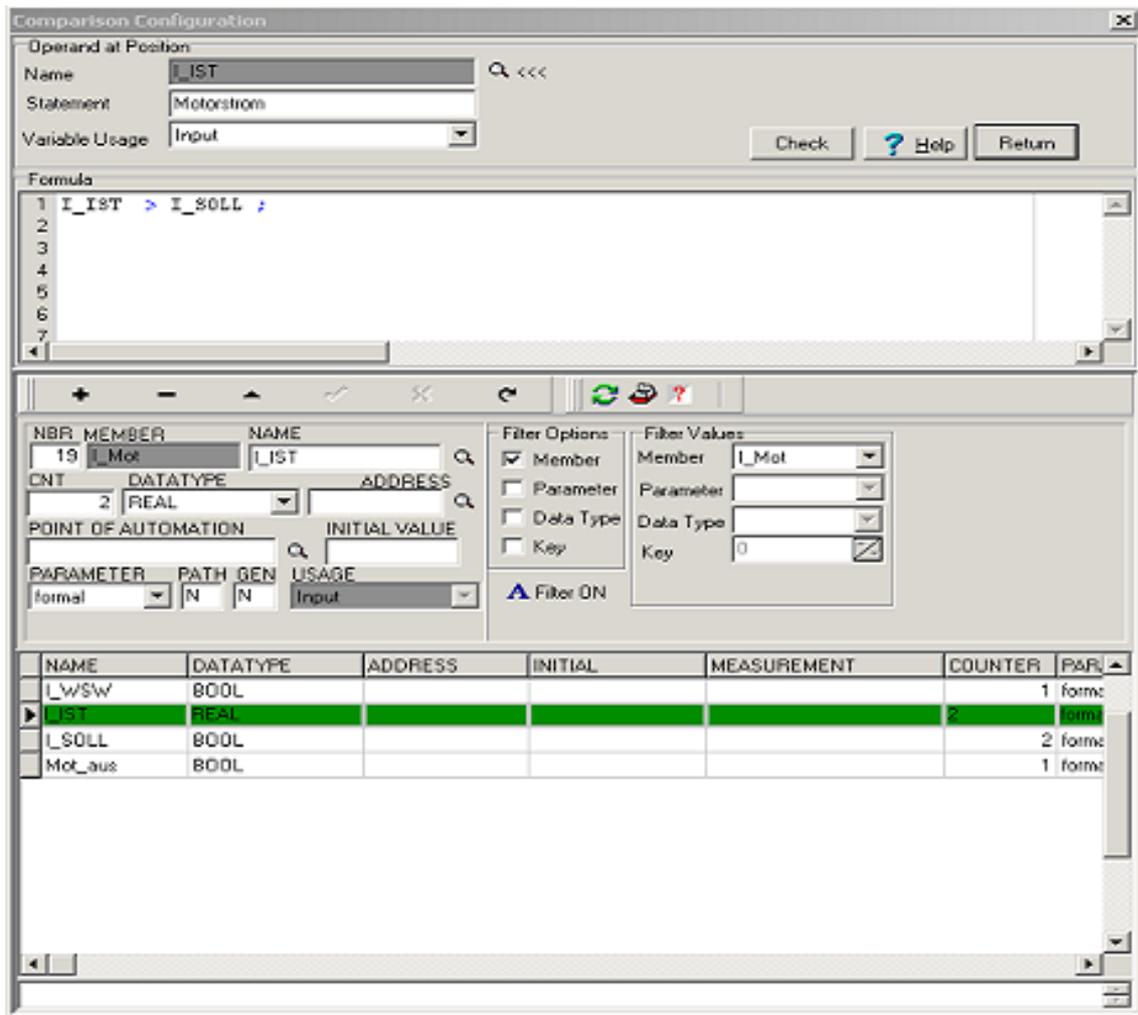
- Ziehen Sie NUR eine Kante in einem NICHT MARKIERTEN OBJEKT. Also erst Klick auf das freie Feld, dann mit dem Zeichnen beginnen.
- Haben Sie zu einem Objekt hinführende Kanten zunächst getrennt gezeichnet, dann haben sie auch keine gemeinsame Linie. Wollen Sie das ändern, d.h. alle Linien auf dem gemeinsamen Weg zu einer zusammenführen, dann legen Sie an den Kanten-Einmündungen immer zwei Kanten übereinander. Fügen Sie eventuell an der passenden Stelle einen Knoten ein und achten darauf, daß die beiden Knoten auch wirklich zusammentreffen, dann werden sie automatisch vereinigt und Pfeile erscheinen an der Einmündung.
- Eine versehentlich gezogene Kante, die noch nicht in einem Zielobjekt angekommen ist, entfernen Sie durch loslassen der linken Maustaste und Klick auf RECHTE Maustaste.
- Sie haben in einem PRAP schon Kanten zwischen die Graphikelemente gezogen und schreiben den Text in eine Operation oder eine Bedingung. Das Graphikelement vergrößert sich, und plötzlich entsteht ein wirres Kanten-Bild.  
Markieren Sie das Element, das Ihr Kanten-Bild zerstört hat, ziehen Sie das Element so weit weg, bis das Kanten-Bild wieder vernünftig aussieht. Dann durch Markieren der Kanten die Knoten sichtbar machen und aus dem Bereich des Graphikelementes herausziehen.

Ein Beispiel soll zeigen, welcher Kanten-Fehler auftreten kann:



## A.233 Projektierung der Vergleichsfunktion

Falls Sie den PRAP-Entwurf sehen wollen: PRAP Motorstrom [siehe Seite 102]



Sie erkennen im Bild zwei Felder, wobei das obere die Vergleichfunktion enthält, das untere die Variablenliste. Nutzen Sie bitte auch die Hilfeangebote zu den Formblättern!

Die grundlegenden Ausführungen finden Sie hier [siehe Seite 34].

#### **Die Handlungsschritte:**

Es wird davon ausgegangen, daß Sie nur das obere Feld vor sich sehen, noch keine Vergleichfunktion vorfinden und die Variablen I\_IST und I\_SOLL noch nicht eingetragen sind.

- Geben Sie diese jetzt ein, indem Sie nach Klick auf die Variablenliste (Access Table) sehen. Klick auf leitet den Vorgang ein. Die Variablen in das Feld "NAME" nacheinander eintragen. Beachten Sie bitte die Ausführung unter 2.
- Gehen Sie davon aus, daß der Analog-Digital-Wandler die Stromwerte als INTEGER-Werte ausgibt. Deshalb ordnen Sie beiden den Datentyp INT zu. USAGE zu I\_IST muß Input, zu I\_Soll Constant sein. Den wirklichen Wert zu I\_Soll übergeben Sie in den Instanzen, weil die Höhe des Stromwertes auch in jedem Teilprozeß (=Instanz) verschieden sein kann.
- Setzen Sie nach der Eingabe der Variablen den Cursor in das Feld "Formula" und drücken Sie Strg+Leerzeichen (Ctrl+Space). Die beiden Variablen wählen Sie aus und fügen sie zur Formel ein, das Vergleichszeichen können Sie mit der Tastatur eingeben oder Sie drücken Strg+J und wählen das Zeichen aus.
- Am Ende das Semikolon als Formelabschluß setzen. Danach "Check" anklicken.
- Wenn alles in Ordnung ist, dann Klick auf "Return", und die Vergleichfunktion sehen Sie im Feld "Formula":

Nun den Vorgang mit Klick auf "OK" beenden.



### Warnung

Falls Sie die Formel mit der Tastatur eintragen, sind die Variablen aber der Datenbank nicht "bekannt". Daher werden Sie z.B. mit Klick auf "Check" die Fehler lesen können. Falls Sie nicht "Return" anklicken, ist die Formel nicht in der Datenbank enthalten.

## A.234 Kanten zeichnen und Probleme

Es gelten folgende Regeln für Kanten:

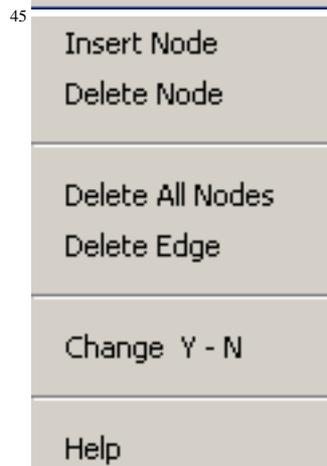
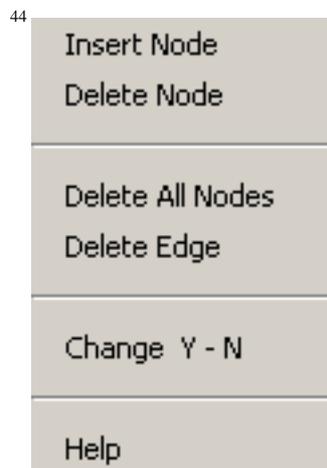
- Die Verbindungen erfolgen stets von Objektmitte zu Objektmitte. Objekte sind Operationen (Operations), Bedingungen (Conditions) und Knoten einer bestehenden Verbindung.
- Der Beginn einer Verbindung heißt auch wegführende Kante vom Objekt. Das Ende einer Verbindung wird auch hinführende Kante zum Objekt genannt.
- Durch die Zieh-Richtung wird die Richtung der Verbindung bestimmt. Der Pfeil wird automatisch am Ende eingetragen.
- Ein Klick am Anfang einer Verbindung setzt deren Beginn. Beim Ziehen einer Verbindung stets Maustaste loslassen. Jeder Klick während des Ziehens erzeugt einen Knoten, der zugleich eine mögliche Knickstelle der Verbindung ist.

- Das Ende eines Ziehvorgangs hat
  - in einem Objekt
  - in einem Knoten einer bestehenden Verbindung
  - exakt an einer anderen Verbindung, sobald ein Kreuz erscheint (ein Knoten wird hier automatisch erzeugt)

durch einen Klick zu erfolgen.

- Einen Knoten (Knickpunkt) verlagert man durch Hinführen des Cursors, im Knotenfeld linke Maus-Taste drücken und halten, bis durch Verschieben der Maus die gewünschte Lage erreicht ist. Dabei ist ein neuer Cursor zu erkennen und die Verbindung bis zur Mitte des Objektes zu sehen.
- Veränderungen aller Art an Kanten sind mit dem Kanten-Menü<sup>44</sup> durchführbar.
- Jede wegführende Kante einer Condition muß bewertet sein. Führt man den Cursor an eine solche Kante, dann öffnet ein Klick auf die rechte Maustaste das Kanten-Menü<sup>45</sup>, das unten "Change Exit Y <-> N" stehen hat. Das bedeutet, die Kanten-Bewertung zu ändern.
- Wenn Sie das Ziehen einer Kante beenden wollen und daher unterbrechen:
  - Sofort die ESCAPE-Taste betätigen und anschließend die linke Maustaste
  - einfach die rechte Maustaste betätigen

Probleme beim Kanten zeichnen:KantenFehler [siehe Seite 214]



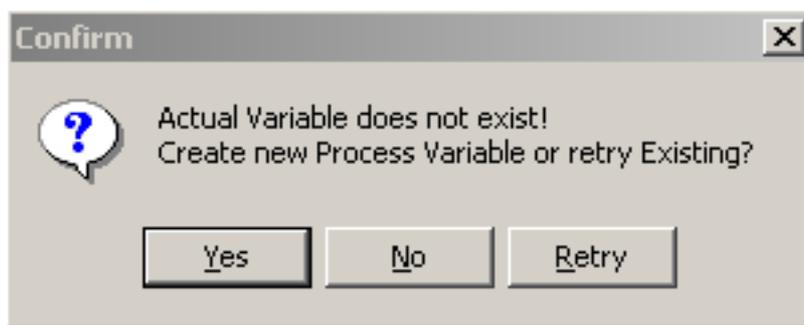
## A.235 Mehrfach-Verwendung von Prozeß-Variablen

Es gibt zwei Arten der Mehrfachverwendung bzw. Wiederverwendung

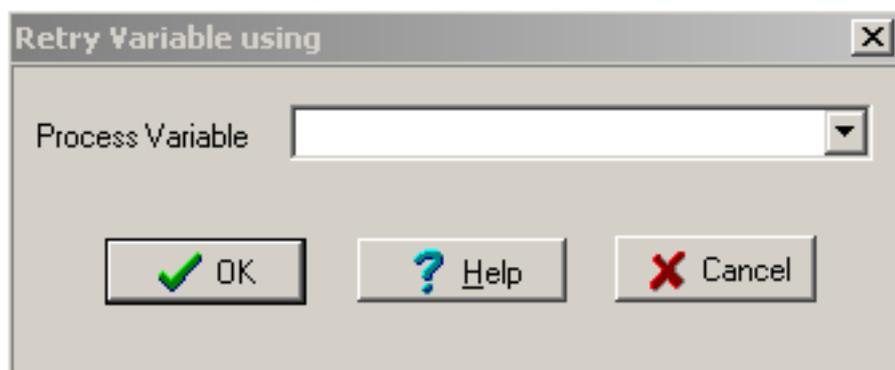
- Prozeß-Variable [siehe Seite 183] werden sehr häufig in einem PRAP mehrfach verwendet. Dort, wo ein VARIABLE NAME (FLAG)- kurz Name - mit STATEMENT, z.B. "Soll\_T" mit "Solltemperatur erreicht?", erstmalig als P-Variable in einer Situation [siehe Seite 183] oder einem Übergangs-Prozeßzustand [siehe Seite 193] eingeführt wird, erhält der Name in der Datenbank auch einen Index, der für Sie als Nutzer keine Rolle spielt. Wollen Sie diese Variable in einer "neuen" Condition wieder verwenden, gleichgültig an welcher Stelle im PRAP, dann suchen Sie sich aus dem "Vorrat" bereits vergebener Namen einen aus. Zunächst erscheint nach dem Klick auf die rechte Maustaste:



Mit Klick auf "Data Access" erhalten Sie zur Entscheidung, ob eine neue Variable angelegt werden soll oder ob Sie eine vorhandene wieder verwenden wollen :



Ein Klick auf "Retry" gibt Ihnen alle bisher im PRAP existierenden<sup>46</sup> zur Auswahl:



Ein neuer Index mit dem selben Namen wird in der Datenbank eingetragen. Sie können am unteren Bildrand nach "Name:" am Ende z.B. (7-1) lesen. Die "7" ist der neue Index, die "1" ist die laufende

<sup>46</sup> Es existieren nur solche P-Variable, für die "NAME" eingetragen wurde!!

Nummer des "Statement", der Variablen-Benennung. So bedeutet (3-3) der dritte Bezeichner, zugleich mit Index 3, ist an dieser Stelle im PRAP erstmalig erschienen. Diese Nummer des Statements finden Sie im graphischen Symbol der P-Variablen eingerahmt wieder.

- Die Wiederverwendung einer Prozeßvariablen aus einem anderen PRAP als dem in Bearbeitung befindlichen entspricht einer Teilprozeß-Kopplung [siehe Seite 170], da jede Prozeßvariable dort zugeordnet ist oder sein sollte, wo durch die Steuerungseingriffe (Operationsvariablen) deren Wert verändert wird. Der in Bearbeitung befindliche PRAP "benötigt" aber diese Variable zur Formulierung oder Gestaltung eines Übergangs-Prozeßzustandes.  
Es handelt sich zunächst um eine NEUE P-Variable, also "Yes" anklicken. Danach Klicken Sie auf  und nutzen den Hilfetext zu "NAME". Entweder mit  oder Klick auf den Help-Button.