



TeCAD Dokumentation

Copyright AVI GmbH

Version 2.2 - 2007

Inhaltsverzeichnis

1. TeCAD	1
1.1. Leistungsumfang von TeCAD	1
2. Prozessablaufplan	2
2.1. Entwicklung eines PRAP	2
2.2. Grundsätze zum Zeichnen eines PRAP	3
2.2.1. Grafikelemente	3
2.2.1.1. Operationen	3
2.2.1.1.1. Initialoperation	5
2.2.1.2. Bedingungen	5
2.2.1.3. Konnektoren	6
2.2.1.4. Freier Text	6
2.2.1.5. Verbindungen	6
2.2.2. Prozessvariablen	7
2.2.3. Operationsvariablen	7
2.2.4. Variablen	7
2.3. Strukturen im PRAP	9
2.3.1. Kombinatorische Strukturen	9
2.3.2. Bedingungsstabilität	10
2.3.3. Strukturstabilität	11
2.3.4. Prüfroutinen	12
2.3.5. Nichtbinäre Funktionen	13
2.3.5.1. Zeitglieder	13
2.3.5.2. Zeitdauerwerte	17
2.3.5.3. Ausdrucksfunktion	17
2.3.5.4. Flankenerkennung	18
2.3.5.5. Anweisungsfunktionen	18
2.4. Zusammenfassung Prozessablaufplan	18
3. Entwurfs-Prozess	19
3.1. Informationsflußmodell	19
3.2. Nachbilden von Prozessgrößen	19
3.3. Einleitung zum Modell	20
3.4. Die Binäre Prozessanalyse	21
4. Bearbeiten von Prozessablaufplänen	22
4.1. Bildschirmdarstellung	22
4.2. Einfügen von Elementen	22
4.3. Bearbeiten von Elementen	22
4.3.1. Markieren von Elementen	23
4.3.2. Ziehmarken	23
4.4. Zeichnen von Verbindungen	23
4.4.1. Verbindungspunkte	23
4.5. Verwenden der Zwischenablage	24
4.6. Undo/Redo	24
4.7. Bild exportieren	24
4.8. Fensterauswahl	25
4.8.1. Grafikfenster	25
4.8.2. Prozessvariablenliste	26
4.8.3. Operationsvariablenliste	26
4.8.4. Variablenliste	26
4.8.4.1. Variablen aus DB lesen	27
4.8.4.2. Variablen in DB schreiben	28
4.8.4.3. Variablen mit DB abgleichen	28
4.8.4.4. Variablen mit DB prüfen	29

4.8.4.5. CSV Dateien	29
4.8.4.5.1. CSV Export von Variablen	29
4.8.4.5.2. CSV Import von Variablen	30
4.8.4.6. Variablen Id's löschen	30
4.8.4.7. Adressen der Variablen löschen	30
4.9. Eigenschaftsfenster	31
4.9.1. Eigenschaften von Konnektoren	32
4.9.2. Eigenschaften von Bedingungen	33
4.9.3. Eigenschaften von Operationen	34
4.9.4. Eigenschaften von Prozessvariablen	35
4.9.5. Eigenschaften von Operationsvariablen	36
4.9.6. Eigenschaften von Variablen	37
4.9.7. Eigenschaften von Verbindungslinien	37
5. Übersetzen	39
6. Dateieigenschaften	40
6.1. Dialog Eigenschaften Auftragsdaten	40
6.2. Dialog Eigenschaften Übersetzung	41
6.3. Dialog Eigenschaften Variablenamen	43
6.4. Dialog Eigenschaften Alle Eigenschaften	45
6.5. Dialog Eigenschaften Operanden	46
7. Optionen	48
7.1. Ansicht	48
7.2. Drucken	49
7.3. Allgemein	50
8. Drucken	51
8.1. Formblatt	52
8.1.1. Variable Texte in Formblättern	52
8.1.2. Erstellen von eigenen Formblättern	52
8.1.3. Dokumentation des Formblatt-Dateiformates	53
9. Syntaxprüfung	56
9.1. Designfehler	56
9.1.1. PRAP muß mindestens eine Operation enthalten	56
9.1.2. Keine Initialoperation festgelegt	56
9.1.3. Ungültiger Knotentyp	56
9.1.4. Verbindung zeigt ins leere	57
9.1.5. Linienanfangspunkt ist nicht belegt	57
9.1.6. Knoten ist doppelt im Pfad	57
9.1.7. Bedingung ist in keinem Pfad enthalten	57
9.1.8. Pfad enthält keine Bedingungen	57
9.1.9. Ungültige Bewertung des Ausganges	57
9.1.10. Bedingung muß zwei komplementär bewertete Ausgänge haben	58
9.1.11. Es sind nur 2 Ausgänge bei einer Bedingung erlaubt	58
9.1.12. Ausgänge müssen unterschiedlich bewertet werden	58
9.1.13. Bedingung ist nicht mit einer Prozessvariable verknüpft	58
9.1.14. Konnektor darf nur für ankommende Pfeile benutzt werden	59
9.1.15. Konnektor hat keinen passenden Partner	59
9.1.16. Es darf nur einen Ziel-Konnektor geben	59
9.1.17. Konnektor darf nur einen Ausgang haben	59
9.1.18. Konnektor hat keine Verbindung	59
9.1.19. Konnektor ist in keinem Pfad enthalten	59
9.1.20. Operation hat keine Übergangspfade	60
9.1.21. PRAP muß stark zusammenhängend sein	60
9.1.22. Programmfehler	60
9.2. Variablenfehler	61
9.2.1. Leere Parameter	61
9.2.2. Variable nicht definiert	61
9.2.3. Variable hat nicht erforderlichen Typ	61

9.2.4. Variable mehrfach beschrieben	61
9.2.5. Variable sollte beschreibbar sein	61
9.2.6. Gleiche Adressen in Variablen	61
9.2.7. Zeitglieder starten	61
10. System Informationen	63
10.1. Technische Voraussetzungen	63
10.2. Installation und Deinstallation	63
10.3. Hardwaremäßiger Softwareschutz	63
10.3.1. Umfang der Lizenzen	64
10.3.2. Treiber	64
10.3.3. Hilfsprogramme	64
10.3.4. Updates	64
10.4. Einschränkungen der Demo-Version	64
10.5. Bezugsquellen	65
11. Referenz	66
11.1. Menü und Schaltflächen	66

Abbildungsverzeichnis

2.1. Operation	4
2.2. Bedingung	5
2.3. Konnektoren	6
2.4. Kombinatorische Strukturen	9
2.5. Bedingungsstabilität	10
2.6. Strukturstabilität	11
2.7. Zeitglieder	13
2.8. PRAP - Zeitglied	14
2.9. Zeitglied - Operationsvariable	15
2.10. Zeitglied - Prozessvariable	16
3.1. Informationsflussmodell	19
3.2. Einleitung zum Modell	20
4.1. Fensterauswahl	25
4.2. Grafikfenster	25
4.3. Eigenschaften Beispiel	31
4.4. Eigenschaften des Konnektors	33
4.5. Eigenschaften der Bedingung	33
4.6. Eigenschaften der Operation	34
4.7. Eigenschaften von Prozessvariablen	35
4.8. Eigenschaften von Operationsvariablen	36
4.9. Eigenschaften von Variablen	37
4.10. Eigenschaften der Verbindungslinie	37
6.1. Eigenschaften Auftragsdaten	40
6.2. Eigenschaften Übersetzung	42
6.3. Eigenschaften Variablennamen	44
6.4. Eigenschaften - Alle Eigenschaften	45
6.5. Eigenschaften Operanden	46
7.1. Optionen-Dialog Ansicht	48
7.2. Optionen-Dialog Drucken	49
7.3. Optionen-Dialog Allgemein	50

Tabellenverzeichnis

8.1. vordefinierte Eigenschaften	52
11.1. Menü und Schaltflächen	66

Kapitel 1. TeCAD

1.1. Leistungsumfang von TeCAD

- Grafisches Bearbeiten von Prozessablaufplänen
- Projektieren von Prozess- und Operationsvariablen
- Verwalten der Variablenliste der einzelnen Teilprozesse
- Prüfung der logischen Korrektheit und Variablenbelegung
- Export des Plans als Bilddatei
- Ex- und Import der Variablenliste im CSV-Format
- Übersetzen von Prozessablaufplänen in strukturierten Text (nicht in Light-Version)
- Variablenabgleichfunktion mit TeVar
- OLE-Schnittstelle (wird von Studio benutzt)

In der LT-Version von TeCAD ist die Übersetzungsfunktion nicht vorhanden und die Variableneigenschaften sind nicht änderbar. Außerdem existieren keine Abgleichfunktionen für die Variablendatenbank.

TeCAD bietet Ihnen die Möglichkeit Steuerungsabläufe technologisch zu planen und für alle Beteiligten transparent sichtbar zu machen.

Wenn Sie mit dem System noch nicht vertraut sind, beginnen Sie mit dem Theoriebereich.

Theorie: [Die binäre Prozessanalyse](#)

Praxis: [Bearbeiten von Prozessablaufplänen](#)

Kapitel 2. Prozessablaufplan

Prozessablaufpläne werden mit dem Modul TeCAD bearbeitet.

Der Prozessablaufplan oder kurz PRAP ist ein gerichteter, stark zusammenhängender Graph, der auf automatentheoretischen Grundlagen beruht. D.h. er erfüllt automatentheoretische Forderungen wie Widerspruchsfreiheit und Vollständigkeit.

Der PRAP verfügt über zwei Grundmengen, die Menge der **Operationen** und die Menge der **Bedingungen**. Die Menge der Operationen und die Menge der Bedingungen sind die Knotenmengen des Graphen. Zur Darstellung dieser Knoten bedient man sich graphischer Elemente. Die Verbindungslinien sind die gerichteten Verbindungen zwischen diesen Knoten. Sie werden durch Pfeile dargestellt.

Konnektoren dienen als Hilfe, um z.B. Verbindungen ohne Kreuzungen darzustellen. Zu einer Operation können viele Pfeile hinführen, sie muss aber nur durch einen verlassen werden. Zu einer Bedingung können viele Pfeile hinführen, sie muss aber nur von zwei Pfeilen verlassen werden. Diese sind mit der Bewertung "ja" bzw. "nein" zu versehen.

Der PRAP als prozessorientiertes Beschreibungsmittel legt den funktionellen Zusammenhang zwischen Prozesseingang und -ausgang fest (siehe: **Informationsflussmodell**), der später von einem Steuergerät (SPS, Leitsystem, Microcontroller) realisiert werden kann. Vor der Erstellung eines PRAP muss der Gesamtprozess in Teilprozesse zerlegt werden (siehe: Thema Prozesszerlegung in der Studio-Hilfe).

siehe auch: [Bearbeiten von Prozessablaufplänen](#)

2.1. Entwicklung eines PRAP

Zuerst müssen die Größen am Teilprozesseingang und -ausgang ermittelt werden. (siehe: **Informationsflussmodell**).

Die Eingangsgrößen werden auf binär bewertete **Prozessvariablen** abgebildet. Die Prozessvariablen erhalten dabei eine verbale Beschreibung für den 'ja' und 'nein' Zustand. Binäre Eingangsgrößen können direkt verwendet werden.

Analoge Eingangsgrößen werden über **Ausdrucksfunktionen** auf binäre Prozessvariablen abgebildet. Die beiden Alternativen werden ebenfalls verbal beschrieben.

Die Ausgangsgrößen werden den binären **Operationsvariablen** zugeordnet. Jede Operationsvariable erhält eine verbale Beschreibung für den Ein- und Auszustand.

Binäre Ausgangsgrößen können direkt abgebildet werden. Analoge Ausgangsgrößen werden über Wertzuweisung oder **Anweisungsfunktionen** abgebildet. Beim Zeichnen des PRAP werden die Prozessvariablen in den **Bedingungen** verwendet, um Übergänge zwischen den Operationen zu beschreiben. In jeder **Operation** wird eine eindeutige Belegung der Operationsvariablen definiert.

Bei Entwicklung eines PRAP werden die Stufen Prozesszerlegung, Zuweisen von Variablen und Bearbeiten des Graphen im Allgemeinen mehrmals durchlaufen.

siehe auch:

[Grundsätze zum Zeichnen eines PRAP](#)

Strukturen im PRAP

2.2. Grundsätze zum Zeichnen eines PRAP

1. Erste **Operation** platzieren. Diese ist im Allgemeinen die **Initialoperation** und kann entsprechend markiert werden.
2. Platzieren von Folgeoperationen nach dieser Operation.
3. Festlegen der Belegung der **Operationsvariablen** in den Operationen.
4. Platzieren von **Bedingungen** zwischen den Operationen und Zeichnen der Verbindungen mit ja/nein Bewertung.
5. Ist eine Prozessgrösse nicht erfassbar, kann sie durch einen Zeit- oder Zählbaustein nachgebildet werden.
6. Alle weiteren möglichen Operationen sind mit einer Untersuchung, der für die Steuerung notwendigen Operationenfolge, zu ermitteln.
7. Beim Übergang von einer Operation zur Folgeoperation können sich beliebig viele der Operationsvariablen bezüglich ihrer Bewertung ändern. Zu beachten ist dabei, dass immer alle Operationsvariablen in Betracht gezogen werden.
8. Wenn es mehrere Folgeoperationen zu einer Operation gibt, müssen sich die Übergänge gegenseitig ausschliessen.
9. Ausser beim Übergang zur anderen Operation muss immer auch ein Pfad zur Ausgangsoperation zurückführen, da diese Bedingung die Stabilität der Operation gewährleistet.

2.2.1. Grafikelemente

Im **PRAP** können verschiedene Objekte dargestellt werden. Dazu gehören:

- Knoten des PRAP-Graphen:
 - **Operationen**
 - **Bedingungen**
 - **Konnektoren**
- **Verbindungspunkte** als gerichtete Verbindungen zwischen den Knoten
- **Freier Text** als Kommentar

2.2.1.1. Operationen

Die Operationen werden im Prozessablaufplan durch ein Rechteck dargestellt. Sie beschreiben den

Ausgangszustand des Prozessablaufes, wenn diese Operation aktiv ist. Es kann immer nur eine Operation gleichzeitig in einem PRAP aktiv sein. So können Teilprozesse angesteuert oder Aggregate eingeschaltet werden.

In der graphischen Darstellung der Operation werden die aktivierten Operationsvariablen angezeigt. Alternativ dazu kann für die Anzeige ein Kommentar eingegeben werden. Beim Ablauf des PRAP sind die aktivierten Operationsvariablen dann eingeschaltet und alle anderen ausgeschaltet.

In jeder Operation wird also die gesamte Ausgangsbelegung beschrieben.

D.h.

- die Belegung der Operationsvariablen ist in jeder Operation eindeutig.
- Mehrere Operationen können die gleiche Belegung von Operationsvariablen haben.

Die Operationsvariable wird eindeutig mit einem technologischen Begriff bezeichnet.

Beispiele für Operationsvariable:

- Kolonne 3 zuschalten,
- Dampfschieber auffahren,
- Motor einschalten
- Pumpe EIN-schalten
- Wartezeit läuft

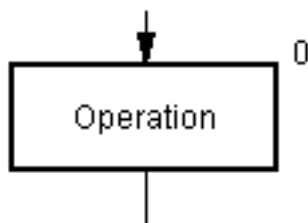
Nicht jedoch:

- Pumpe EIN / AUS,
- Schieber AUF / ZU.

An der oberen rechten Ecke der Operation steht die Operationsnummer. Die Operationsnummer wird automatisch fortlaufend für alle Operationen innerhalb eines Prozessablaufplanes vergeben.

Die Start- oder auch **Initialoperation** ist durch einen doppelten Rand gekennzeichnet.

Abbildung 2.1. Operation



Hinweis:

Eine Operationsvariable kann Prozessvariable in einem anderen Teilprozess werden, insoweit ein Zusammenhang mit ihm hergestellt werden soll.

Der gewählte technologische Begriff sollte dann in dem zugeordneten Teilprozess in gleicher Weise bezeichnet sein.

Wenn Operationsvariablen nicht direkt mit Binärvariablen verknüpft sind (Funktion direkt), wird die Funktion in der Reihenfolge der aktivierten Operationsvariablen nach Abarbeitung des Automaten ausgeführt.

siehe auch: [Grafikelemente](#)

2.2.1.1.1. Initialoperation

Diese Operation ist die Startoperation eines Prozessablaufplanes. Ein PRAP enthält maximal eine solche Operation.

2.2.1.2. Bedingungen

Die Bedingung wird im Prozessablaufplan als abgerundetes Rechteck dargestellt. Sie kennzeichnet im Prozess eine Prozessvariable. Die Bedingung hat immer etwas mit einer Prüfung oder Messung zu tun.

Die einzelnen Elemente für die Bedingungen werden auch als Prozesszustandsvariable oder Prozessvariable bezeichnet. In der Bedingung steht die Aussage der verknüpften Prozessvariablen.

Beispiele für die Bezeichnung von Bedingungen:

- NOT-AUS
- Dampfschieber Endlage ZU
- Trocknersystem angefahren
- Wartezeit abgelaufen
- Druck grösser 3 bar

Die Prozessvariable verzweigt den Prozess entsprechend ihrer prozessualen Aussage. Abgehende Verbindungslinien erhalten die Bewertung 'j' oder 'n'. Welche Linie mit 'j' oder 'n' weitergeführt werden soll, das entscheidet der Entwickler. Der Zustand der abgehenden Linie kann über das Kontextmenü geändert werden. Bei der PRAP-Prüfung wird man darauf hingewiesen, wenn Verbindungen doppelt sind oder fehlen.

Mit den Bedingungen werden die Übergangs- und Haltepfade in einem Prozessablaufplan formuliert.

An der oberen rechten Ecke der Bedingung steht die Nummer der Prozessvariablen.

Abbildung 2.2. Bedingung



Hinweis:

Bedingungen in einem Prozessablaufplan eines Teilprozesses können von Operationen aus einem anderen Teilprozess ausgehen, sollten dazu aber in gleicher Weise bezeichnet sein. Diese sind dann sogenannte Koppelvariablen.

Jede Bedingung ist mit einer Prozessvariable verknüpft. Wenn Prozessvariablen nicht direkt mit Binärvariablen verknüpft sind (Funktion direkt), wird die Funktion vor Abarbeitung des Automaten in der Reihenfolge der Prozessvariablen berechnet (Ausnahme: Zeitglied abgelaufen wird zusammen mit Zeitglied ansteuern am Ende berechnet) .

siehe auch: Andere [Grafikelemente](#)

2.2.1.3. Konnektoren

Konnektoren werden durch einen Kreis oder eine Ellipse dargestellt. Konnektoren können entweder Quelle oder Ziel einer Verbindung sein. Dementsprechend sind nur abgehende oder hinführende Verbindungslinien erlaubt.

Abbildung 2.3. Konnektoren



Konnektoren werden verwendet, um Verbindungslinien aufzutrennen. Das geschieht zweckmässigerweise an Seitengrenzen, bei langen Verbindungen und um Kreuzungen zu vermeiden.

siehe auch: Andere [Grafikelemente](#)

2.2.1.4. Freier Text

Neben den Grafikelementen kann auch freier Text in den Plan eingefügt werden. Dieser kann zum Kommentieren des Planes benutzt werden.

Nach dem Einfügen hat das Textfeld eine Standardgrösse. Diese kann nachträglich geändert werden. Der Text passt sich automatisch der Feldgrösse an.

siehe auch: Andere [Grafikelemente](#)

2.2.1.5. Verbindungen

Verbindungslinien stellen eine logische Verknüpfung zwischen den Grafikelementen her. Eine Verbindungslinie ist immer gerichtet, d.h. sie hat eine Quelle und ein Ziel. Es können mehrere Linien hintereinander vorhanden sein oder mehrere Linien an einem Punkt zusammenlaufen und eine Linie weiterführen.

Siehe auch: [Zeichnen von Verbindungslinien](#)

2.2.2. Prozessvariablen

Prozessvariablen bilden die Eingangsgrößen des [Prozessablaufplanes](#).

Sie stellen die mit ja/nein (binär) bewerteten Ausgangsvariablen eines technologischen 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 zu bedienenden Menschen (Operateur).
- Reaktionen von Zeit- und Zählgliedern auf Grund ihrer Ansteuerung
- Ergebnisaussage einer [Ausdrucksfunktion](#)
- natürliche und erzeugte Koppelvariable aus anderen Teilprozessen, den 'Empfang' von Steuerinformationen im Rahmen der Koppelbeziehungen.

Prozessvariablen werden mit [Bedingungen](#) in einem PRAP verknüpft, um Übergänge zwischen den [Operationen](#) zu definieren.

2.2.3. Operationsvariablen

Operationsvariablen sind die Ausgangsgrößen eines [Prozessablaufplanes](#).

Sie können sein:

- Stellgrößen für alle prozesstechnologischen Eingriffe
- Informationen an den Operateur im Rahmen der Mensch-Maschine-Kommunikation
- Erzeugte Koppelvariable zur funktionellen Beeinflussung anderer Teilprozesse
- Ansteuersignale für Zeit- und Zählgliedern, Analogwertausgaben, Anstoss von [Rechenoperationen](#)

Die Belegung der Operationsvariablen wird in einer [Operation](#) eindeutig festgelegt.

2.2.4. Variablen

Variablen beschreiben die konkreten Speicherstellen bzw. Ein- und Ausgänge in einem Steuergerät. Variablen werden entsprechend der Norm IEC 61131-3 deklariert. Jede Variable hat mindestens einen Namen und einen Typ.

Variablen haben folgende Eigenschaften:

Name

Bezeichnung der Variablen. Entsprechend der Norm dürfen nicht alle Zeichen und keine Leerzeichen verwendet werden. Die entsprechende TeVAR-Spalte ist NAME.

Typ

Datentyp der Variablen. Es sind alle Typen, die in dem Zielsystem zulässig sind verwendbar. Die entsprechende TeVAR-Spalte ist TYP.

Operand

Der Operand bezeichnet die Adresse der Variablen. Nicht alle Variablen müssen Adressen haben. Die entsprechende TeVAR-Spalte ist OP.

Kommentar

Ein freier Text zum Kommentieren der Variable. Die entsprechende TeVAR-Spalte ist COMMENT.

Initialwert

Der Wert wird beim Start des Steuergerätes angenommen. Im Moment dient dieses Feld nur zur Kommentierung. Die entsprechende TeVAR-Spalte ist INITVALUE.

TeVar-ID

Wenn die Variablenliste mit TeVAR abgeglichen wurde, wird hier die eindeutige ID der Variablen automatisch eingetragen. Die entsprechende TeVAR-Spalte ist ID.

Verwendung

Dieses Feld zeigt die Verwendung der Variablen im PRAP an. Es wird automatisch aktualisiert.

Parameter

Zeigt an, dass diese Variable ein Parameter der generierten POE sein soll. Dies wird bei Bibliotheksprozessen eingesetzt um mehrere Instanzen des Prozesses bilden zu können.

Parameter haben normalerweise keinen Operanden. Falls die Variable trotzdem einen Operanden besitzt, wird sie als Direktvariable generiert und nicht als Parameter. Die Parametereigenschaft wird dann bei der Instanziierung umgespeichert.

Temp

Zeigt an, dass diese Variable nur temporär verwendet wird. Das bedeutet dass der Wert der Variablen während des Programmablaufs verloren geht. Der Wert ist also am Anfang immer undefiniert. Diese Option sollte nur dann verwendet werden, wenn der Wert der Variablen überhaupt nicht verwendet werden soll, oder wenn genau klar ist, dass die Auswertung des Wertes nach einer Wertzuweisung erfolgt. Schauen Sie sich im Zweifel den generierten ST-Code an, um die Reihenfolge der Bearbeitung zu sehen.

Temp-Variablen müssen keinen Operanden besitzen. Sie werden dann im VAR_TEMP-Block des ST-Codes generiert.

Lokal

Zeigt an, dass die Variable nur in diesem PRAP verwendet werden soll. Dies wird beim Abgleich mit TeVAR geprüft.

In Funktionsblöcken müssen diese Variablen keinen Operanden besitzen. Sie werden dann als statische Variablen generiert.

In der Variablenliste sind alle Variablen die in der übersetzten POE (Programm-Organisations-Einheit) deklariert sein sollen. Darin sind alle Variablen enthalten, die in Prozess- und Operationsvariablen verwendet werden.

2.3. Strukturen im PRAP

Der PRAP zeichnet sich u.a. dadurch aus, dass er automatentheoretische wie auch graphentheoretische Forderungen erfüllt. Die Erfüllung dieser und anderer Forderungen muss zwangsläufig erfolgen, deshalb kann eine [Syntaxprüfung](#) erfolgen.

Außerdem ist der PRAP hinsichtlich seiner Stabilität, Sicherheit, Dynamik und Vereinfachbarkeit zu überprüfen.

Ein Durchlauf durch eine Operation ist kritisch zu betrachten und wenn möglich zu vermeiden. Beachten Sie die Eigenschaften [bedingungsstabil](#) und [strukturstabil](#) eines Übergangs von einer Operation zur Folgeoperation.

Es ist zu darauf zu achten, dass nicht alle Übergänge gleichzeitig auftreten. Es bilden sich dadurch instabile Zyklen, die unbedingt vermieden werden müssen.

Wenn Unsicherheiten bezüglich der Zuverlässigkeit bei Übergängen von einer Operation zur Folgeoperation bestehen, so sollten Sie strukturstabile Übergänge bevorzugen.

Das dynamische Verhalten der [Prozessvariablen](#) bei Änderung von [Operationsvariablen](#) ist unbedingt zu beachten.

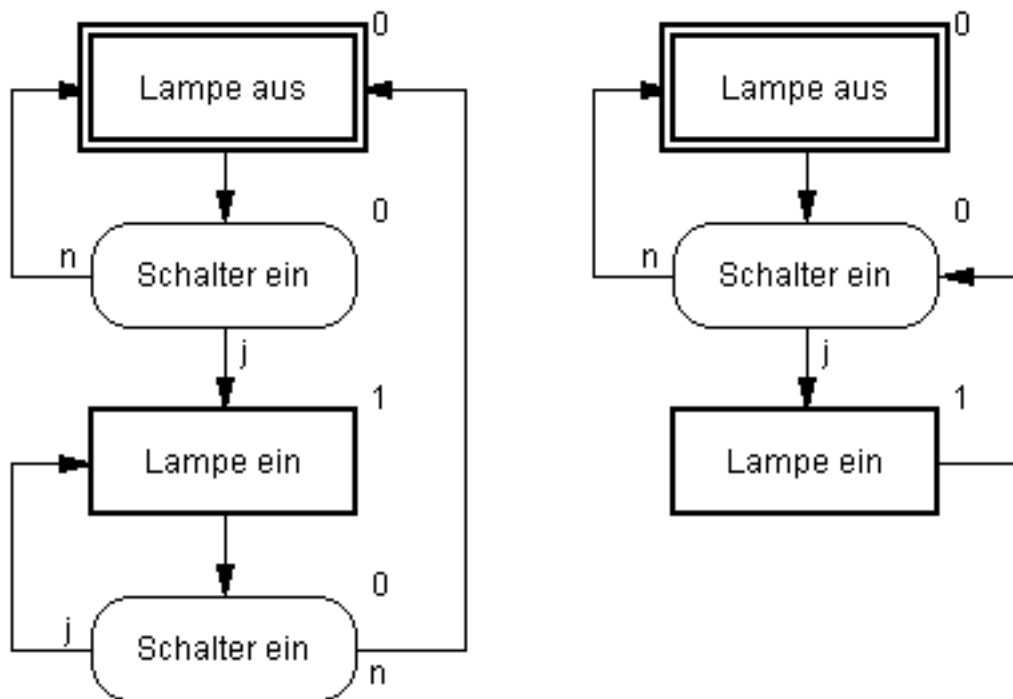
Durch Einführung [kombinatorischer Strukturen](#) kann die Gestalt des PRAP mitunter deutlich vereinfacht werden.

2.3.1. Kombinatorische Strukturen

Diese Strukturen können dann auftreten, wenn mehrere [Operationen](#) direkt von [Prozessvariablen](#) abhängig sind. In vielen Fällen kann die Darstellung dann stark vereinfacht werden und der PRAP wird übersichtlicher.

Im folgenden Beispiel ist der Zustand der Lampe direkt vom Zustand des Schalters abhängig. Durch Umzeichnen erreicht man eine kompaktere Darstellung. Die Funktion bleibt aber die gleiche.

Abbildung 2.4. Kombinatorische Strukturen



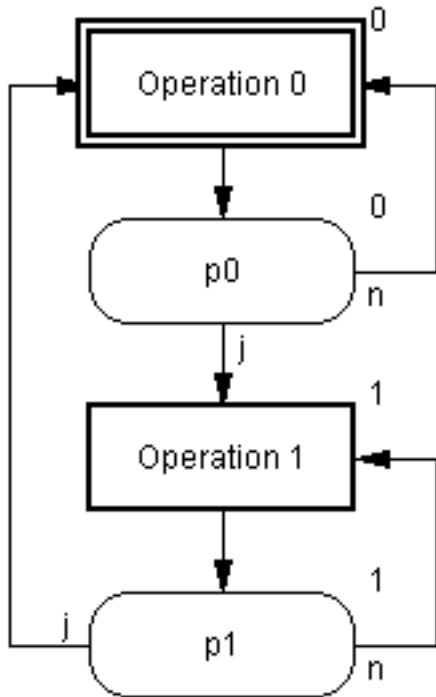
2.3.2. Bedingungsstabilität

Ein Übergang von einer **Operation** zur Folgeoperation heißt bedingungsstabil, wenn die Haltebedingungen der Folgeoperation nicht abhängig ist von den Übergangsbedingungen von Operation zur Folgeoperation. Die Abhängigkeit kann durchaus technologisch bedingt sein. Da im PRAP keine Informationen über technologische Abhängigkeiten vorliegen, liegt es in der Verantwortung des Benutzers zu entscheiden, ob eine solche Abhängigkeit vorliegt oder nicht.

Im Zweifelsfall sollten **strukturstabile** Übergänge bevorzugt werden.

In der Abbildung ist zu sehen, dass der Haltezustand der Operation 1 durch die Prozessvariable 1 = nein realisiert wird und nicht durch die Prozessvariable 0.

Abbildung 2.5. Bedingungsstabilität

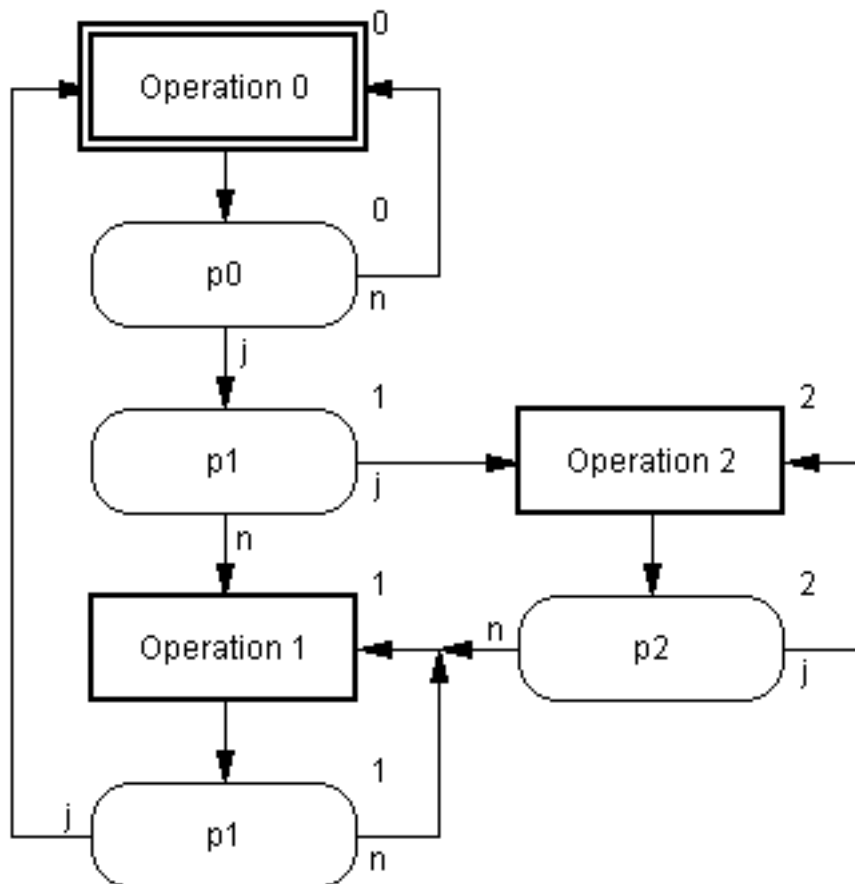


2.3.3. Strukturstabilität

Ein Übergang von einer **Operation** zur Folgeoperation heisst strukturstabil, wenn der Haltezustand der Folgeoperation abhängig von Bedingungen des Übergangszustandes erfüllt wird. Damit ist die Stabilität der Folgeoperation garantiert. Eine Ausnahme bilden dynamische Übergangsbedingungen, die nur für den Moment des Übergangs erfüllt sind (**Flankenerkennung**).

In der Abbildung ist zu sehen, dass im Übergang von Operation 1 zu Operation 2 schon eine Bedingung (die Prozessvariable $p_2 = \text{nein}$) für die Stabilität der Operation 2 (sprich für den Haltezustand der Operation 2) enthalten ist. Genauso wurde mit dem Übergang von Operation 1 zu Operation 3 verfahren.

Abbildung 2.6. Strukturstabilität



2.3.4. Prüfroutinen

Hat man den PRAP nach eigenem Ermessen fertiggestellt, startet man die Prüfroutinen. Die Prüffunktion erreicht man über das Menü **Extras** sowie beim Übersetzen mit **Datei > ST erzeugen**. Hier kann man auswählen, welche Prüfung man ausführen möchte.

In der ersten Phase prüft man das Graphendesign. Hier werden Hinweise gegeben, wo Nachbesserungen im PRAP erforderlich werden.

Die Prüfung wird solange wiederholt, bis Fehlerfreiheit erreicht ist. Zwischendurch sollte das Speichern des PRAP nicht vergessen werden.

Danach ist es ratsam die Variablenprüfung durchzuführen. Und es sollten die noch bestehenden Defekte beseitigt werden.

Wenn Fehler auftreten, wird das fehlerhafte Element markiert (wenn es möglich ist) und die Meldung angezeigt.

Es wird nur Code generiert, wenn keine Fehlermeldung mehr erscheint.

Der strukturierte Text befindet sich dann in einer Datei mit der Endung 'st', die genauso heißt wie die prp-Datei.

Hinweis:

Es ist nicht möglich, den strukturierten Text in einen PRAP zurückzuübersetzen. Deshalb ist es auch nicht ratsam, den strukturierten Text von Hand nachzubearbeiten.

2.3.5. Nichtbinäre Funktionen

Neben der Projektierung der Prozesssteuerung durch binäre Grössen treten im PRAP häufig Aufgaben der **Zeitsteuerung** auf. Ebenso können andere Anweisungen erforderlich sein.

Mit der **Ausdrucksfunktion** können z.B Istwerte mit Sollwerten verglichen werden oder andere beliebige Funktionen aufgerufen werden.

Die **Flankenerkennung** dient zur Vorverarbeitung von Pegelwechseln.

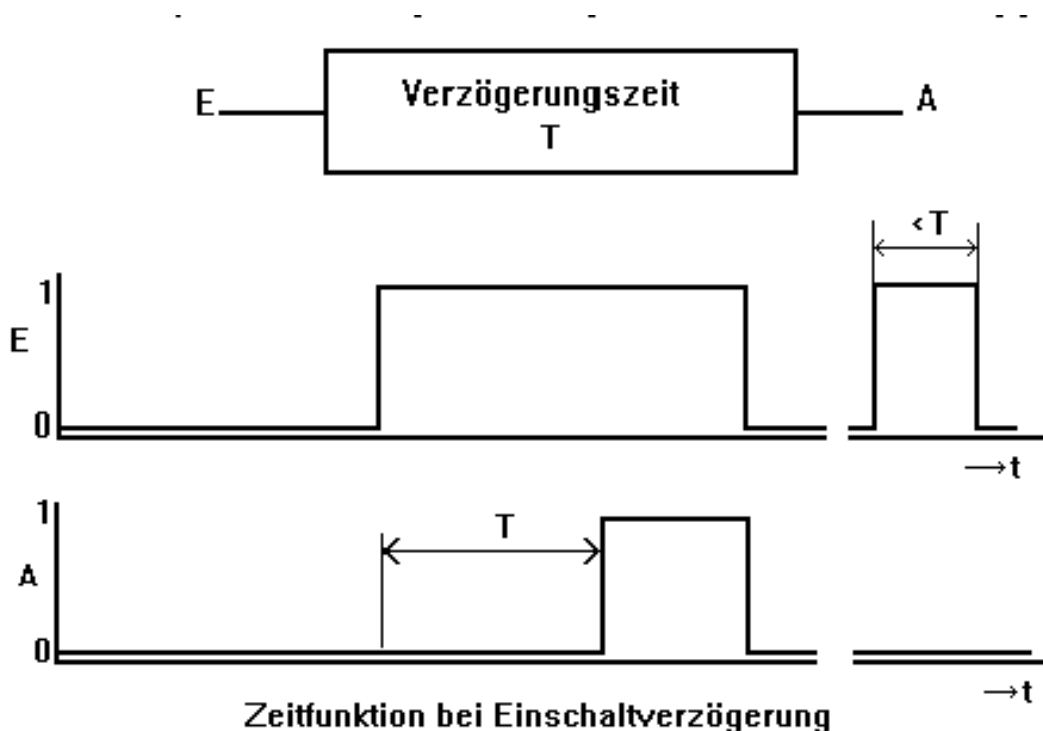
Die **Anweisungsfunktionen** bieten die Möglichkeit in Operationen z.B. Rechnungen oder andere beliebige Anweisungen durchzuführen.

Für alle Funktionen gilt natürlich, dass sie unabhängig vom Hersteller einer SPS notiert werden. Es gibt aber manche Funktionen, die wegen der Eigenheiten der Produkte, also der SPS, nur mit Einschränkungen oder überhaupt nicht projektierbar sind. Die erforderlichen Hinweise finden Sie im entsprechenden Text zu den jeweiligen Übersetzern.

2.3.5.1. Zeitglieder

Im Rahmen der **binären Prozessanalyse** werden ausschliesslich Zeitglieder mit Einschaltverzögerung benötigt, da im PRAP alle Zeitfunktionen damit erfassbar sind. Das Projektieren von Zeitgliedern erfolgt selbstverständlich unabhängig vom Hersteller der praktisch eingesetzten SPS.

Abbildung 2.7. Zeitglieder



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)
- Ausgabe A schaltet ein, sobald T beendet ist
- Bei kürzerer Einschaltdauer als T bleibt A ausgeschaltet
- A schaltet zusammen mit E aus.

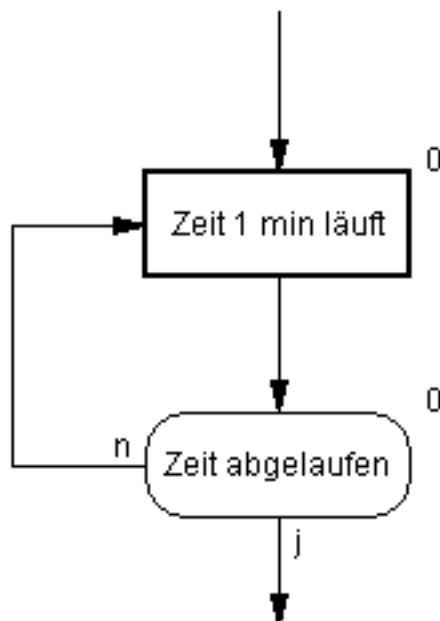
Im PRAP verwenden Sie für E eine technologisch passende Bezeichnung einer **Operationsvariablen** mit der Funktion 'Zeitglied erzeugen'. Die Zeitwerte müssen entsprechend der IEC-1131-Norm als **Zeitdauerwerte** angegeben werden.

Alternativ dazu kann der Zeitwert auch von einer Variablen gelesen werden. Dann ist der Bezeichner der Variablen anzugeben.

Für den Ausgang formulieren Sie die technologisch sinnvolle **Prozessvariable** mit der Funktion 'Zeitglied abgelaufen'. Diese Form des Vorgehens entspricht dem **Informationsflussmodell**.

Beispiele:

Abbildung 2.8. PRAP - Zeitglied



In der Operation mit der Operationsvariable 'Zeit 1 min läuft' wird der Timer über die Funktion 'Zeitglied ansteuern' (siehe Beispiel Operationsvariable) gestartet. Der Zustand wird dann über die Bedingung so lange gehalten, bis der Timer in der Bedingung abgelaufen ist. Welche Parameter für die Operationsvariable und der Prozessvariable notwendig sind, ist in den folgenden Abschnitten 'Operationsvariable' und 'Prozessvariable' beschrieben.

Operationsvariable

- Zeit 1 min läuft

Abbildung 2.9. Zeitglied - Operationsvariable

The image shows a software dialog box titled "Eigenschaften der Operation". It is divided into several sections:

- Geometrie**: A collapsed section.
- Operation**:
 - O-Variable(n)**: A list box containing "<< Neu >>" and "Zeit 1 min läuft". "Zeit 1 min läuft" is selected.
 - Kommentar**: An empty text area with scroll arrows.
- Details**:
 - Aussage**: Text input field containing "Zeit 1 min läuft".
 - neg. Aussage**: Text input field containing "nicht Zeit 1 min läuft".
 - Funktion**: A dropdown menu showing "Zeitglied ansteuern".
 - Timer**: Text input field containing "T1".
 - Zeitwert**: Text input field containing "T#60".
 - Hilfsvar. (opt.)**: An empty text input field.
- Variable**: A collapsed section.

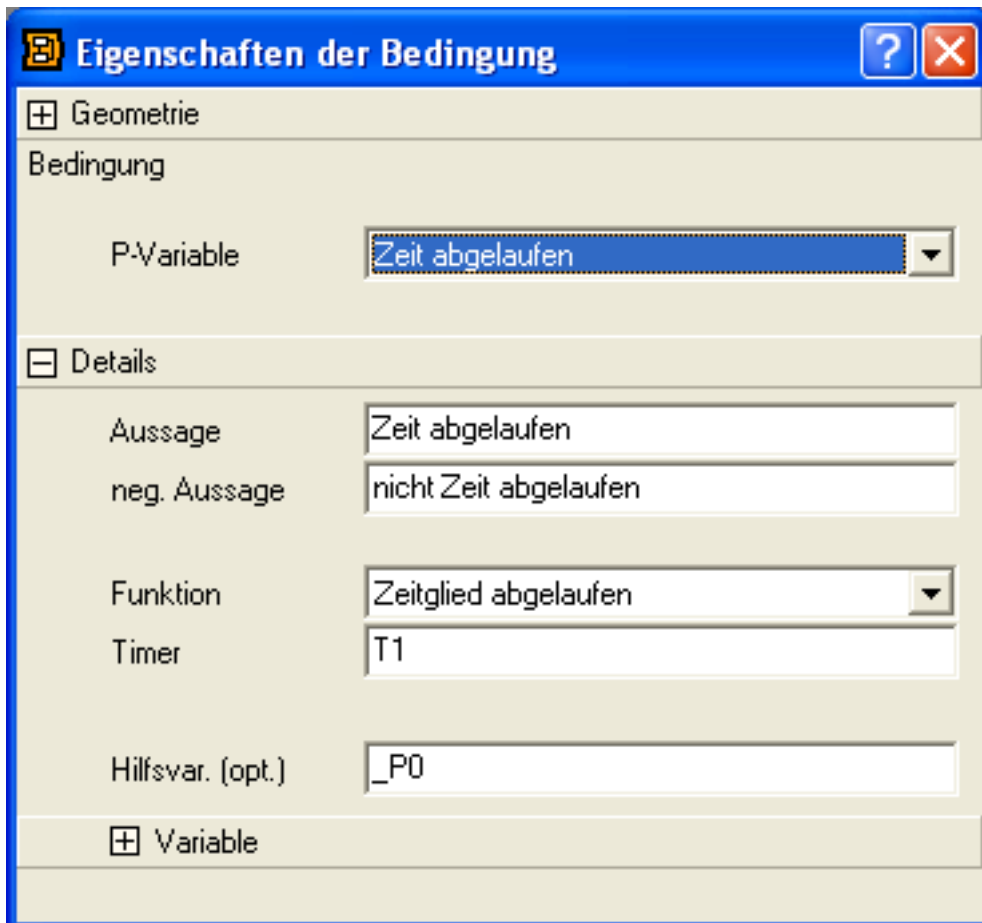
Zuerst müssen Sie über '<< Neu >>' eine Operationsvariable anlegen. Dann wählen Sie die Funktion 'Zeitglied ansteuern' aus. In den Parametern 'Timer' und 'Zeitwert' der Funktion, geben Sie den

Namen und den Zeitwert in Sekunden an. Der Zeitwert ist immer mit T# und dann die Zeit in Sekunden anzugeben.

Prozessvariable

- Zeitglied abgelaufen

Abbildung 2.10. Zeitglied - Prozessvariable



Zuerst müssen Sie über '<< Neu >>' eine Prozessvariable anlegen. Dann wählen Sie die Funktion 'Zeitglied abgelaufen' aus. In dem Parameter 'Timer' geben Sie den Namen des Timers an, den Sie in der Operationsvariable definiert haben.

Die **Operation**, in der Sie das Zeitglied einschalten müssen, wird durch den Prozessablauf bestimmt. Sie können einen solchen Zeitbaustein auch über mehrere Operationen eingeschaltet lassen. 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.

Um diese Funktionen bedarfsgerecht ein- und auszuschalten, achten Sie auf die Operationenfolge im PRAP. In Folgeoperationen eingeschriebenes '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 Zeitwertes. Sie können so auch dasselbe Zeitglied mit verschiedenen Zeiten und Operationsvariablen mehrmals in einem PRAP benutzen.

Jedes Zeitglied wird nur dann in dem betreffenden Teilprozess bzw. dessen AWL generiert, wenn in der entsprechenden Operationsvariable die Sonderfunktion 'Zeitglied erzeugen' eingestellt wurde. Das Zeitglied muss einen eindeutigen Bezeichner erhalten. Achten Sie auf Überschreitungen des Maximalwertes bei Zeit- und Zählgliedern, die sehr vom Hersteller der SPS abhängen.

Die Variable des Zeitgliedes hat nach IEC-Standard den Typ 'TON'. Zielsystemabhängig können auch andere Typen eingesetzt werden. Bei S7 z.B. 'TIMER'.

2.3.5.2. Zeitdauerwerte

Konstante Zeitwerte müssen links mit den Schlüsselwörtern T#, TIME#, t# oder time# beginnen. Die Zeitangabe muss ohne Leerstellen in folgender Form folgen: <Stunden>h<Minuten>m<Sekunden>s<Millisekunden>ms. Teile davon können weggelassen werden. Die niederwertigste Zeiteinheit kann auch als reelle Zahl ohne Exponent geschrieben werden.

Beispiele:

- T#12h34m"
- T#36m21s
- t#80ms
- time#3.5h
- TIME#3h20.33m

2.3.5.3. Ausdrucksfunktion

Ausdrücke werden verwendet um nichtbinäre Eingangsgrößen auf binäre Aussagen abzubilden. Dabei sind alle im strukturierten Text zulässigen Ausdrücke mit dem Ergebnistyp BOOL zulässig.

Häufig verwendete Ausdrücke sind Vergleichsfunktionen. Dort bringt eine **Prozessvariable** zum Ausdruck, dass eine Prozessgröße wie Füllstand, Position oder Druck usw. einen bestimmten Wert erreicht, über- oder unterschritten hat. Das Vergleichsergebnis entspricht der Aussage ja/nein dieser Variablen.

Beispiele:

- Aussage: Füllstand grösser Max
neg. Aussage: Füllstand nicht grösser Max
Ausdruck: FUELLSTAND > 100
- Aussage: Istwert weicht vom Sollwert um mehr als 10% ab
neg. Aussage: Istwert weicht vom Sollwert nicht um mehr als 10% ab
Ausdruck: ABS(ISTWERT-SOLLWERT)>SOLLWERT*0.1

Sie können auch Verknüpfungen zwischen BOOL-Werten verwenden um z.B. Sammelsignale aus mehreren Sensoren zu bilden.

Beispiel:

- Aussage: Band läuft schief
neg. Aussage: Band läuft nicht schief
Ausdruck: SCHIELFLAUF1 OR SCHIEFLAUF2

2.3.5.4. Flankenerkennung

Mit der Flankenerkennung haben Sie die Möglichkeit, Prozesssignale und deren technologische Bedeutung am besten durch einen Pegelwechsel darzustellen und entsprechend auszuwerten.

Tasten an Bedientableaus sind dafür das beste Beispiel.

Die Prozessvariable hat die Eigenschaft, nur für den Zeitpunkt des Pegelwechsels erfüllt zu sein. Mit Flanken können also auch bedingungsstabile Übergänge zwischen Operationen projiziert werden.

Sie sollten diese Funktion trotzdem nur für oben genannte Fälle einsetzen. Auf keinen Fall sollten Koppelgrößen mit Flanken ausgewertet werden. Der Prozess erhält dadurch eine Eigendynamik, die sich bei der Fehlersuche nur schwer nachvollziehen lässt.

2.3.5.5. Anweisungsfunktionen

Mit Anweisungen können bei aktivierter Operationsvariable komplexe Funktionen ausgeführt werden.

Eine Anweisung entspricht dabei der Syntax im strukturierten Text.

Normalerweise ist eine Anweisung gleichzeitig eine Zuweisung eines Ausdrucks zu einer Variablen. In diesem Fall ist im Eingabefeld 'Ziel' ein Variablenname einzutragen und im Feld 'Ausdruck' der Ausdruck der dem Ziel zugewiesen werden soll.

Falls die Anweisung keine Zuweisung ist, wird das Eingabefeld 'Ziel' freigelassen.

2.4. Zusammenfassung Prozessablaufplan

Operationsvariablen sind Ausgaben des Teilprozesses, während Prozessvariablen die Eingaben sind.

Die Operations- und Prozessvariablen werden textuell entsprechend dem technologischen Kontext bezeichnet. Die Bezeichnung ist so zu wählen, dass die unterlagerte Funktion genau beschrieben wird.

Operations- und Prozessvariablen sind direkt oder über Funktionen mit dem Variablenhaushalt des Teilprozesses verknüpft. Die Variablen bilden die Schnittstelle zum technologischen Prozess oder auch zu anderen Teilprozessen.

Kapitel 3. Entwurfs-Prozess

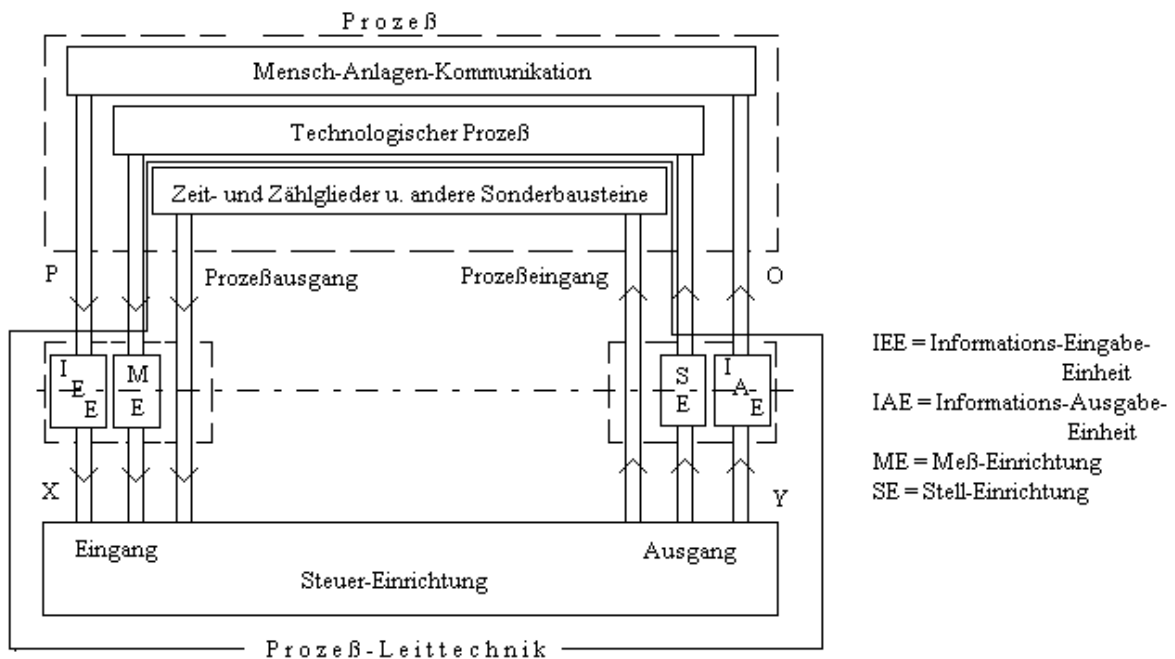
3.1. Informationsflußmodell

Das in der Abbildung dargestellte Informationsflußmodell zeigt den Fluss der Information zwischen gesteuertem Prozess und der Steuereinrichtung z.B. einer SPS. Wie die Abbildung zeigt, wird die am Prozessausgang P auftretende Anzahl **Prozessvariable** in Eingangsvariable des Eingangs X der Steuereinrichtung transformiert. Jede am Prozesseingang O wirkende Operation ist Abbild des Ausgangs Y der Steuereinrichtung. Die Ausgangsvariablen werden in die **Operationsvariable** überführt. Die technische Realisierung erfolgt u.a. durch die Prozess-Mess- und Stelltechnik.

Hierzu werden oft sogenannte **nichtbinäre Funktionen** wie Zeit- und Zählglieder, Vergleicherefunktionen, Analogwertausgaben (für Grenzwerte, Zeit- und Zählwerte usw.) und andere benötigt.

Das Verhalten des Prozesses unter dem Einfluss der ihn steuernden, beeinflussenden Operationen spiegelt sich in Änderungen der Prozessvariablen wider und ist im Steuerungsmodell **Prozessablaufplan** (PRAP) dokumentiert.

Abbildung 3.1. Informationsflußmodell



3.2. Nachbilden von Prozessgrößen

Ist eine Prozessgröße nicht erfassbar, kann sie durch einen Zeit- oder Zählbaustein nachgebildet werden.

In vielen Fällen kann auf die Nachbildung verzichtet werden, wenn die erwartete Prozessgröße durch eine Berechnung oder Abbildung eindeutig beschrieben werden kann.

3.3. Einleitung zum Modell

An das Beschreibungsmodell werden folgende Anforderungen gestellt

- Das Modell ist nicht festgelegt, egal ob der Prozess sequentielles (Schrittfolge, Batchprozess) oder kombinatorisches Verhalten (Funktionsplan, Boolesche Algebra) hat.
- In jedem Prozessschritt müssen Aussagen zu Stabilitätsproblemen visuell erkennbar sein, d.h. Stabilitätsbedingungen sind immanenter Bestandteil des Modells.
- Es muss von den Fachleuten in allen Einzelheiten lesbar sein.
- Das Modell muss geeignet sein, ein SPS- Programm automatisch zu erzeugen, nachdem Hardware-details des Steuerungsprojektes hinzugefügt wurden.

Der Prozessablaufplan (PRAP) ist ein gerichteter, zusammenhängender Graph, der auf automaten-theoretischen Grundlagen beruht. D.h. er erfüllt automaten-theoretische Forderungen wie Widerspruchsfreiheit und Vollständigkeit.

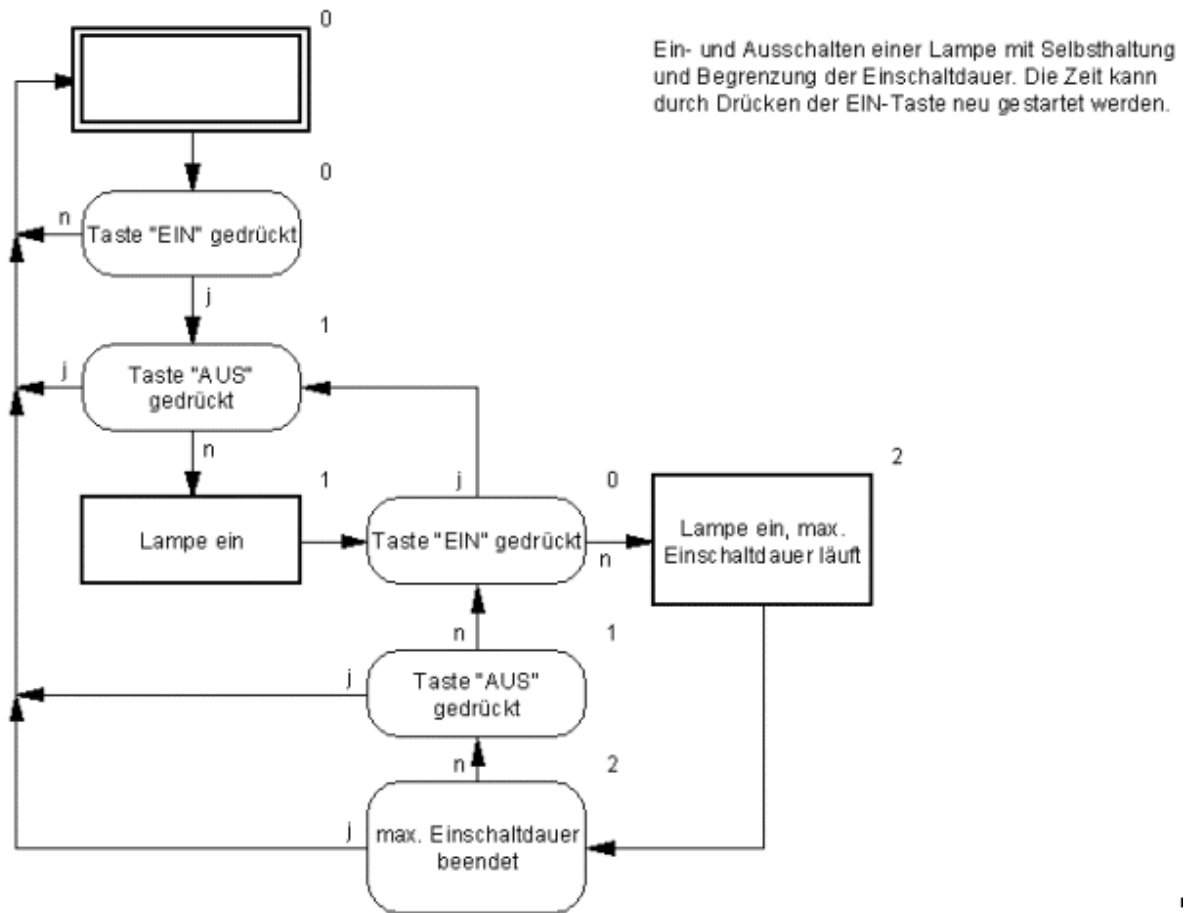
Der PRAP als prozessorientiertes Beschreibungsmittel legt den funktionellen Zusammenhang zwischen Prozesseingaben und -ausgaben (Messgrößen und Steuersignale) fest. Der PRAP kann später von einem Steuergerät (SPS, Leitsystem, Mikrocontroller) realisiert werden.

TeCAD dient der Entwicklung von Prozessablaufplänen, nachdem man sich einen hinreichenden Überblick über das Gesamtsteuerungsproblem durch Prozessanalyse geschaffen hat .

Dabei sind folgende Erkenntnisse aus der Automatentheorie zu berücksichtigen:

- Ein Prozessablauf hat immer einen Anfang (Start der SPS, Initialisierung); dieser Prozessanfang wird als Initialoperation gekennzeichnet.
- In einem Prozessablaufplan ist immer nur eine Operation aktiv.
- Eine Operation wird von einer Vorgängeroperation über Bedingungen erreicht.
- Jede Operation muss von jeder Operation aus (auch indirekt) erreichbar sein. (Die Initialoperation ist davon automatisch betroffen, da sie immer nach der Initialisierung erreicht wird.)

Abbildung 3.2. Einleitung zum Modell



3.4. Die Binäre Prozessanalyse

Das Erarbeiten einer Automatisierungskonzeption ist ohne eine fehlerfrei bzw. fehlerarm und systematisch formulierte Aufgabenstellung nicht sinnvoll. Der [Prozessablaufplan](#) ist ein graphisches Beschreibungsmittel, das in verständlicher Form und fehlerfrei im Sinne der Automatentheorie die binäre Steuerungsaufgabe eines oder mehrerer Prozessschritte formuliert. Um sich dieses Beschreibungsmittels richtig bedienen zu können, muss u.a. der Informationsfluss zwischen gesteuertem Prozess und der Automatisierungseinrichtung geklärt sein.

Dazu dient das [Informationsflussmodell](#).

Kapitel 4. Bearbeiten von Prozessablaufplänen

Ein Prozessablaufplan besteht aus verschiedenen Grafikelementen.

Dazu gehören Operationen, Bedingungen, Konnektoren, Verbindungslinien und Kommentare .

4.1. Bildschirmdarstellung

Die Darstellung entspricht der von Windows-Anwendungen. Deshalb ist die Bedienung mit wenig Aufwand zu erlernen.

Hier noch einige grundlegende Hinweise:

Die Vergrößerung der Darstellung ist in weiten Grenzen frei wählbar. Bei zu kleiner Darstellung werden Details (Schrift) automatisch ausgeblendet.

Wenn während des Bearbeitens Artefakte auf dem Bildschirm zurückbleiben, können Sie durch Wahl des Menüpunktes **Ansicht > Neu zeichnen** das Bild bereinigen.

Sie können im TeCAD mehrere Dateien öffnen. Wenn Sie dieselbe Datei mehrmals öffnen, wird die im Speicher gehaltene Kopie verwendet. Eine Änderung in einem Fenster wirkt sich automatisch auf alle anderen Fenster aus. Sie können so z.B. in einem Fenster eine Übersicht über den PRAP mit 25% Vergrößerung darstellen und in einem anderen Fenster den PRAP bearbeiten.

Eine geöffnete Datei ist für andere Benutzer im Netzwerk gesperrt. Aus dem gleichen Grund kann es passieren, dass Sie eine Fehlermeldung erhalten, wenn Sie versehentlich TeCAD ein zweites Mal starten.

4.2. Einfügen von Elementen

Über die Symbolleiste, das Menü oder die im Menü angezeigten Tastenkürzel wird das Einfügen ausgelöst. Der Mauszeiger ändert sich entsprechend und das Element kann auf dem Bildschirm positioniert werden. Durch einen linken Mausklick wird das Element abgelegt. Der Klickpunkt liegt dabei in der Mitte der Oberkante des Elementes.

Sie können Operationen, Bedingungen, Konnektoren und Kommentare einfügen. Das Einfügen kann mit der mittleren Maustaste wiederholt werden.

Außerdem können Verbindungslinien gezeichnet werden.

4.3. Bearbeiten von Elementen

Die Eigenschaften der Grafikelemente können durch Doppelklick auf das Element bearbeitet werden. Damit wird zuerst das **Eigenschaftsfenster** geöffnet, falls er nicht schon offen ist.

Die Position und Grösse der Elemente können direkt geändert werden. Durch Ziehen mit der Maus oder Betätigen der Pfeiltasten können die **markierten** Elemente in Rasterschritten verschoben werden. Durch Ziehen mit der Maus an den **Ziehmarken** kann die Grösse oder die Lage der Eckpunkte verändert werden.

Mit der Strg-Entf-Taste werden die markierten Elemente gelöscht. Verknüpfte Variablen bleiben dabei erhalten.

4.3.1. Markieren von Elementen

Durch einfaches Anklicken mit der linken Maustaste wird ein einzelnes Element markiert. Die Markierung wird sichtbar durch die an den Rändern des Elementes dargestellten **Ziehmarken**.

Mit der Maus können mehrere Elemente durch Aufziehen eines Rechteckes markiert werden. Wird die Strg-Taste gedrückt gehalten, werden Elemente zusätzlich markiert.

4.3.2. Ziehmarken

Ziehmarken sind kleine quadratische Klötzchen, die im Allgemeinen an den Ecken von Grafikobjekten erscheinen.

4.4. Zeichnen von Verbindungen

Verbindungen können zwischen zwei Knoten oder zwischen einem Knoten und einer Linie gezeichnet werden.

Wenn der Mauszeiger über einen freien **Verbindungspunkt** eines Knotens oder an das Ende einer Verbindungslinie bewegt wird, wird dieser Punkt hervorgehoben und es kann sofort durch Drücken der linken Maustaste mit dem Zeichnen einer Linie begonnen werden.

Als zweite Möglichkeit kann der Befehl 'Verbindung zeichnen' (Strg-L) ausgewählt werden. Dann muss zuerst in einem Knoten oder am Ende einer anderen Verbindungslinie die linke Maustaste gedrückt werden. Die Linie beginnt bei dem nächstliegenden **Verbindungspunkt** des Elements. Mit gedrückter Maustaste wird die Linie zum Zielpunkt gezogen und dort die Maustaste losgelassen.

Durch Loslassen der Maustaste ausserhalb von Knoten und erneutes Klicken und Ziehen können beliebig viele Zwischenpunkte gesetzt werden.

Lässt man die Maustaste auf einem Knoten los, wird die Linie zu dem **Verbindungspunkt** gezogen, der dem letzten Linienpunkt am nächsten liegt.

Wenn die Maustaste auf einer anderen Linie losgelassen wird, wird diese an der Verbindungsstelle aufgetrennt und es wird ein zusätzlicher Pfeil eingefügt.

Bei abgehenden Verbindungen von Bedingungen erhält die Verbindung außerdem die Bewertung 'ja' oder 'nein', je nachdem ob schon eine andere Verbindung vorhanden ist. Diese Bewertung kann nachträglich über das Kontextmenü oder mit Strg-Leertaste geändert werden.

4.4.1. Verbindungspunkte

Jeder PRAP-Knoten besitzt vier Verbindungspunkte an die zu- und abgehende **Verbindungslinien** gezeichnet werden können. Die Punkte liegen rechts, links, oben und unten an den Rändern des Knotens.

Verbindungslinien haften an diesen Punkten, so dass die Linie beim Verschieben des Elementes oder der Linie entsprechend verlängert werden. Wenn eine kürzere Verbindung zu einem anderen freien Verbindungspunkt desselben Grafikelementes möglich ist, wird der Verbindungspunkt gewechselt.

4.5. Verwenden der Zwischenablage

Die Zwischenablage wird zum Kopieren von PRAP-Elementen verwendet.

Beim Kopieren werden die **markierten** Elemente in die Zwischenablage kopiert. Alle Informationen zu den in den Elementen verwendeten Variablen werden mitkopiert.

Das Ausschneiden funktioniert wie das Kopieren, nur dass die markierten Elemente danach gelöscht werden.

Beim Einfügen in einen vorhandenen PRAP können die Elemente mit der Maus positioniert werden. Verwendete Variablen werden an das Ende der vorhandenen Variablenliste angehängt.

Das Einfügen kann mit der mittleren Maustaste wiederholt werden.

4.6. Undo/Redo

Mit der Undo-Funktion können Bearbeitungsschritte rückgängig gemacht werden. In einem Zwischenspeicher werden bis zu 100 Schritte aufbewahrt.

Mit der Redo-Funktion können unmittelbar nach den Undo-Aufruf Änderungen wiederhergestellt werden.

Auf diese Weise können alle Aktionen in einem PRAP einfach ausprobiert werden. Wenn das Ergebnis dann nicht gefällt, kann es einfach rückgängig gemacht werden.

4.7. Bild exportieren

Die Grafik des **PRAP** kann als BMP, WMF oder EMF-Datei exportiert werden. Diese Datei kann dann in anderen Programmen als Grafik eingefügt werden.

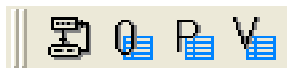
Beim Export als BMP wird die Auflösung und Vergrößerung der aktuellen Bildschirmdarstellung verwendet. Es werden auch Elemente exportiert die nicht zu sehen sind. Wenn Sie z.B. eine BMP-Datei für den späteren Ausdruck erzeugen wollen, wählen Sie vor dem Export eine möglichst hohe Vergrößerung (z.B. 200%)

Die Funktion wird über das Menü **Extras > Plan als Bild speichern...** aufgerufen. Es werden alle **markierten** Elemente exportiert. Wenn nichts markiert ist, wird die gesamte Grafik exportiert.

4.8. Fensterauswahl

Zu einem Prozessablaufplan gehören vier Fenster: Grafikfenster, Prozessvariablenliste, Operationsvariablenliste und Variablenliste.

Abbildung 4.1. Fensterauswahl



Zu den Fenstern kann mit den Buttons in der Werkzeugleiste gewechselt werden. Am Beginn der Bearbeitung ist das Grafikfenster geöffnet.

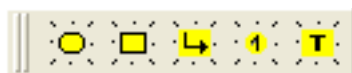
4.8.1. Grafikfenster

Für die Entwicklung eines PRAP benötigt man nur fünf Grundelemente:

- Rechteck als Symbol für eine Operation
- Abgerundetes Rechteck als Symbol für eine Bedingung
- Kleiner Kreis oder Ellipse als Symbol für einen Konnektor
- Linie mit Pfeil zur Darstellung der Verbindung zwischen Operationen und Bedingungen und Vorgabe der Richtung des funktionalen Zusammenhangs
- Textfeld für Hintergrundinformationen zu einer Funktion oder zu einem funktionalen Zusammenhang.

Mit den fünf Grundelementen kann der PRAP intuitiv erstellt werden, indem diese zuerst in der Werkzeugleiste angeklickt werden:

Abbildung 4.2. Grafikfenster



Für Korrekturen im PRAP werden die üblichen bekannten Mittel im Programm angewendet (z.B. Markieren, Verschieben, Löschen (mit Strg-Entf) u.a.) .

Die Initialoperation kann über das Kontextmenü auf eine Operation festgelegt werden.

Die Vergabe von Konnektoren muss eigenständig gepflegt werden. Dazu können auch freie Texte

verwendet werden.

Jedes Element hat vier Verbindungspunkte für Linien, die beliebig verwendet werden können.


Für jede eingefügte Bedingung muss eine abgehende Linie 'ja' und eine 'nein' gezeichnet werden. Die Bewertung der Linie kann über das Kontextmenü geändert werden. Linien können auch an einer anderen Linie beendet werden.

Mit Doppelklick auf ein Element wird das Eigenschaftsfenster geöffnet, welches immer die Eigenschaften des markierten Elements anzeigt.

Hinweis:

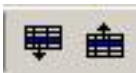
Alle Zeichenoperationen werden im eingestellten Raster ausgeführt. Das Raster kann unter dem Menü **Ansicht** den Bedürfnissen angepasst werden. Wenn Elemente nicht im Raster liegen, können sie mit der Maus nicht angeklickt werden. Über das Menü **Ansicht > An Gittergröße anpassen** kann dieses Problem behoben werden.

4.8.2. Prozessvariablenliste


Die Prozessvariablenliste kann man durch Wechseln zu dem entsprechenden Fenster  anzeigen.

Die Liste kann durch Klicken auf den jeweiligen Spaltentitel sortiert werden.

In der Liste sind nur die Aussage und die negierte Aussage bearbeitbar. Die anderen Einträge müssen über das Eigenschaftsfenster geändert werden.


Über diese beiden Buttons  kann die Reihenfolge in der Liste verändert werden. Außerdem kann dies mittels Verschieben der Zeile mit der Maus geschehen.

4.8.3. Operationsvariablenliste

Die Operationsvariablenliste kann man durch Wechseln zu dem entsprechenden Fenster  anzeigen. Die Anzeige ähnelt stark der Prozessvariablenliste.

Zusätzlich werden die Nummern der Operationen angezeigt, in denen die jeweilige Operationsvariable aktiviert ist. So kann man leicht überflüssige Einträge erkennen.

4.8.4. Variablenliste

Die Variablenliste kann man durch Wechseln zu dem entsprechenden Fenster  anzeigen. Die Liste kann durch Klicken auf den jeweiligen Spaltentitel sortiert werden. Alle Eigenschaften sind direkt bearbeitbar. In der Variablenliste werden alle Variablen angezeigt, die in diesem Teilprozess deklariert sind. Darunter sind auch eine Reihe von Hilfsvariablen, die TeCAD jeweils vor dem Prüfen, Übersetzen und vor Datenbankfunktionen aktualisiert.

Die Namen dieser Variablen werden über die Variablennamenformate gesteuert, die in den Dateiei-

genschaften eingestellt werden können. Diese Variablen beginnen in der Standardeinstellung mit einem Unterstrich '_' und haben den Typ BOOL. Es gibt davon folgende Möglichkeiten:

- `_INIT`: Mit dieser Variable wird die Initialisierung gesteuert.
- `_Opx`: Dies sind die Speicher für die aktive Operation (x= Nummer der Operation).
- `_Px`: Dies sind Speicher für berechnete Prozessvariablen (x wird fortlaufend nummeriert).
- `_Fx`: Dies sind Speicher für Flankenbildungen (x wird fortlaufend nummeriert).

Die `_Px` - und `_Fx` - Variablen sind dann auch in den jeweiligen Funktionen sichtbar. Sie werden nur angelegt, wenn keine Variable von Hand vorgegeben wurde. Die Eigenschaften dieser Variablen können bis auf Name und Typ geändert werden.

In der Variablenliste kann der Abgleich mit TeVar aufgerufen werden. Dazu gibt es die folgenden Funktionen:

- [Variablen aus DB lesen](#)
- [Variablen in DB schreiben](#)
- [Variablen mit DB prüfen](#)
- [Variablen mit DB abgleichen](#)

Zusätzlich gibt es zum Bearbeiten ausgewählter Variablen folgende Funktionen:

- [Variablen Id's löschen](#)
- [Adressen der Variablen löschen](#)

4.8.4.1. Variablen aus DB lesen

Die Variablen-ID ist eine eindeutige Referenz zu einer Variablen. Sie wird nach dem ersten Lesen aus der Datenbank in dem Teilprozess gespeichert. Beim ersten Lesen wird die Variable anhand ihres Namens und der Prozessbezeichnung in der Datenbank gesucht.

Die Felder NAME,OP und COMMENT werden aus der Datenbank übernommen.

Das Feld TYP wird nur übernommen, wenn es im Teilprozess leer ist oder nur die Gross/Kleinschreibung geändert wurde.

Die Werte werden nur übernommen, wenn sie nicht leer sind, damit keine Informationen versehentlich gelöscht werden.

In der Datenbank wird die Querverweisliste aktualisiert.

Hinweise:

Variablennamen können in TeVar führende Ziffern enthalten.

Diese Variablennamen bekommen im Teilprozess automatisch einen Unterstrich vorangestellt, um diese für den strukturierten Text gültig zu machen.

Wichtig

Diese Funktion ist in TeCADLT nicht verfügbar

4.8.4.2. Variablen in DB schreiben

Wenn noch keine Variablen-ID gespeichert ist, wird die Variable wie beim Lesen anhand Ihres Namens und der Prozessbezeichnung in der Datenbank gesucht. Falls keine Variable gefunden wurde, wird in der Datenbank eine neue Variable (mit RESOURCE und PROC) angelegt.

Die Felder NAME,TYP,OP und COMMENT werden in die Datenbank geschrieben.

Die Werte werden nur geschrieben, wenn sie nicht leer sind, damit keine Informationen versehentlich gelöscht werden.

Wenn die Operandenadresse im Teilprozess und in der Datenbank leer ist und die Variable als lokal gekennzeichnet wurde, wird in der Datenbank eine neue Operandenadresse vergeben.

In der Datenbank wird die Querverweisliste aktualisiert.

Hinweise:

Variablenamen können in TeVar führende Ziffern enthalten.

Diese Variablenamen bekommen im Teilprozess automatisch einen Unterstrich vorangestellt, um diese für den strukturierten Text gültig zu machen.

Wichtig

Diese Funktion ist in TeCADLT nicht verfügbar

4.8.4.3. Variablen mit DB abgleichen

Zuerst wird aus der Datenbank gelesen. Eventuelle Fehlermeldungen werden dabei ignoriert. Dann wird in die Datenbank geschrieben.

Wenn die Variablen-ID jetzt noch leer ist, wird sie aus der Datenbank übernommen. Wenn die Operandenadresse leer und die Variable lokal ist, wird zum Schluss das Feld OP aus der Datenbank gelesen.

In der Datenbank wird die Querverweisliste aktualisiert.

Hinweise:

Variablenamen können in TeVar führende Ziffern enthalten.

Diese Variablenamen bekommen im Teilprozess automatisch einen Unterstrich vorangestellt, um diese für den strukturierten Text gültig zu machen.

Wichtig

Diese Funktion ist in TeCADLT nicht verfügbar

4.8.4.4. Variablen mit DB prüfen

Wenn noch keine Variablen-ID gespeichert ist, wird die Variable wie beim Lesen anhand Ihres Namens und der Prozessbezeichnung in der Datenbank gesucht. Es wird geprüft, ob die Felder NAME, TYP und OP mit der Datenbank übereinstimmen.

Dabei wird die Gross/Kleinschreibung bei NAME und TYP nicht beachtet.

In der Datenbank wird die Querverweisliste aktualisiert.

Hinweise:

Variablenamen können in TeVar führende Ziffern enthalten.

Diese Variablenamen bekommen im Teilprozess automatisch einen Unterstrich vorangestellt, um diese für den strukturierten Text gültig zu machen.

Wichtig

Diese Funktion ist in TeCADLT nicht verfügbar

4.8.4.5. CSV Dateien

CSV-Dateien sind Textdateien die zeilenweise Daten enthalten. Die erste Zeile enthält dabei die Namen der Spalten.

In der CSV-Datei sind die Felder durch ein Trennzeichen getrennt. Dieses Trennzeichen ist unabhängig von den Ländereinstellungen immer ein Semikolon (;).

Wenn ein Semikolon in den Daten eines Feldes vorkommt, muß das Feld in Anführungszeichen (") gesetzt werden. Wenn wiederum Anführungszeichen vorkommen, muß diesen ein Backslash (\) vorangestellt werden.

4.8.4.5.1. CSV Export von Variablen

Der Variablenexport kann in der Variablentabelle über das Menü **Extras > CSV Export** aufgerufen werden.

Es werden alle angezeigten Spalten exportiert. Die Spaltennamen sind sprachunabhängig:

- nr
- name
- typ
- op

- comment
- initvalue
- id
- usage
- parameter
- constant
- local

4.8.4.5.2. CSV Import von Variablen

Der Variablenexport kann in der Variablen-tabelle über das Menü **Extras > CSV Import** aufgerufen werden.

Beim Import werden folgende Spalten berücksichtigt:

- name
- typ
- op
- comment
- initvalue
- parameter
- constant
- local

Mindestens sind 'name' und 'typ' erforderlich. Alle anderen Spalten werden ignoriert. Die gelesenen Variablen werden an die vorhandene Liste angefügt. Eventuell auftretene Fehler (z.B. doppelte Namen) werden dabei angezeigt und der Import abgebrochen.

Wichtig

Diese Funktion ist in TeCADLT nicht verfügbar

4.8.4.6. Variablen Id's löschen

Diese Funktion entfernt die Verknüpfung zwischen der Datenbank und der Variablen und wird über den Menüpunkt "Bearbeiten>Variablen Id's löschen" aufgerufen. Danach enthält die "Id" der Variablen den Wert 0.

4.8.4.7. Adressen der Variablen löschen

Bei dieser Funktion werden die Adressen der markierten Variablen aus der Variablenliste entfernt. Die Funktion wird über den Menüpunkt "Bearbeiten>Adressen der Variablen löschen" aufgerufen.

4.9. Eigenschaftsfenster

Das Eigenschaftsfenster ist die zentrale Stelle zur Anzeige und **Bearbeitung** von Elementeigenschaften. Das Aussehen dieses Fensters passt sich immer dem markierten Element an. Im Falle der Dialog noch nicht geöffnet worden ist, müssen sie auf das Element doppelklicken ansonsten nur einfach. In der darunterliegenden Abbildung ist als Beispiel eine komplette Ansicht einer Operation dargestellt.

Abbildung 4.3. Eigenschaften Beispiel

Eigenschaften der Operation
? X

⊕ Geometrie

Operation

O-Variable(n)

Kommentar

⊖ Details

Aussage

neg. Aussage

Funktion

Timer

Zeitwert

Hilfsvar. (opt.)

⊖ Variable

Name

Typ

Operand

Kommentar

TeV-Var-ID

Anfangswert

Verwendung

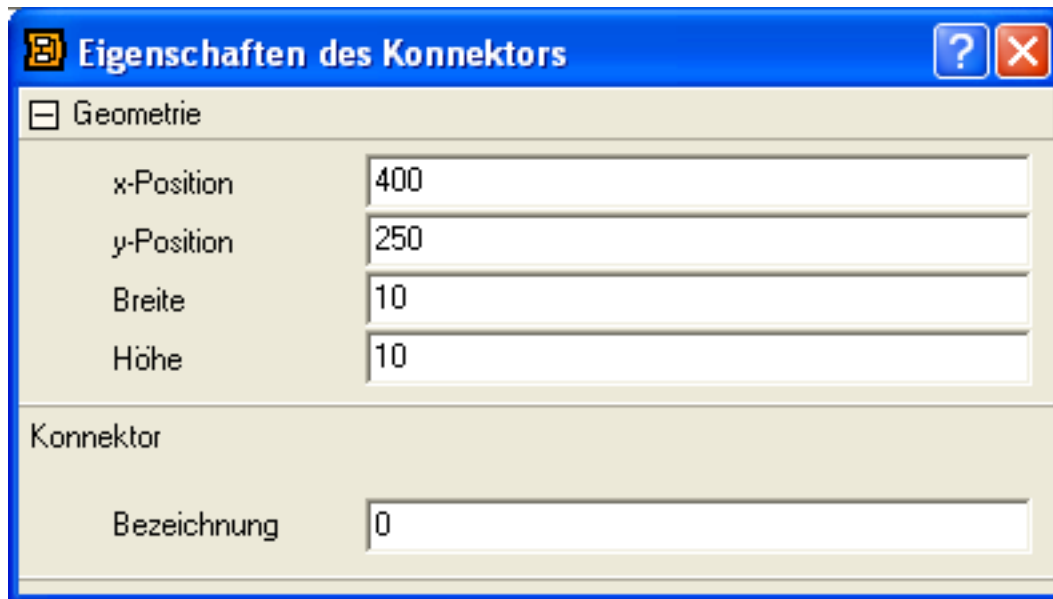
Parameter Temp Lokal

4.9.1. Eigenschaften von Konnektoren

Als Beispiel ist hier beim Konnektor das Geometriefeld eingeblendet. Dieses Feld ist bei allen grafischen Elementen vorhanden. Bei Bedarf können hier die Koordinaten manuell angepasst werden.

Der Konnektor hat ansonsten nur die Bezeichnung.

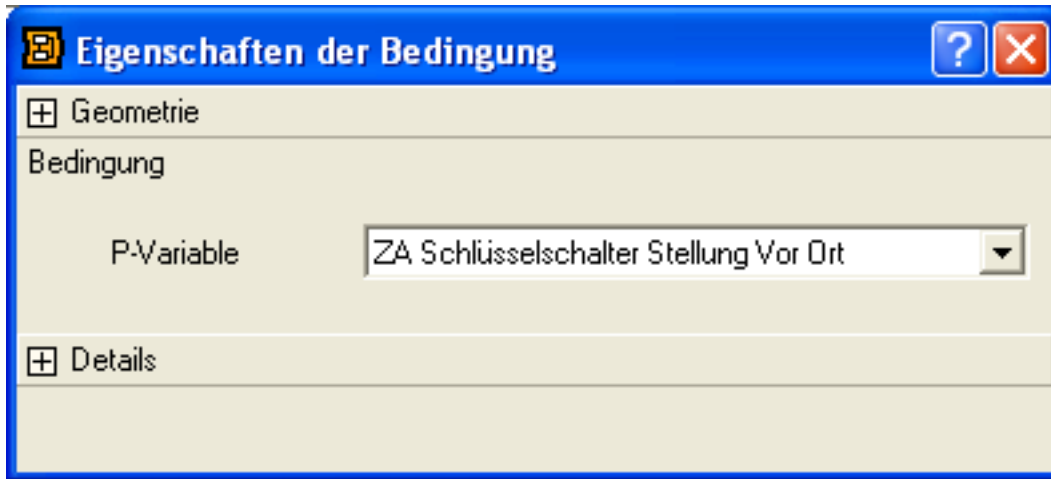
Abbildung 4.4. Eigenschaften des Konnektors



4.9.2. Eigenschaften von Bedingungen

Bei der Bedingung kann eine Prozessvariable ausgewählt werden. Bei Auswahl des Eintrages <<neu>> wird eine neue Prozessvariable angelegt. Bei eingeblendeten Details wird diese Prozessvariable dort angezeigt.

Abbildung 4.5. Eigenschaften der Bedingung



4.9.3. Eigenschaften von Operationen

In der Liste der Operationsvariablen ist die ja- oder nein-Bewertung der jeweiligen Operationsvariable zu sehen. Dies ist die Ausgangsbelegung, wenn die Operation aktiv ist.

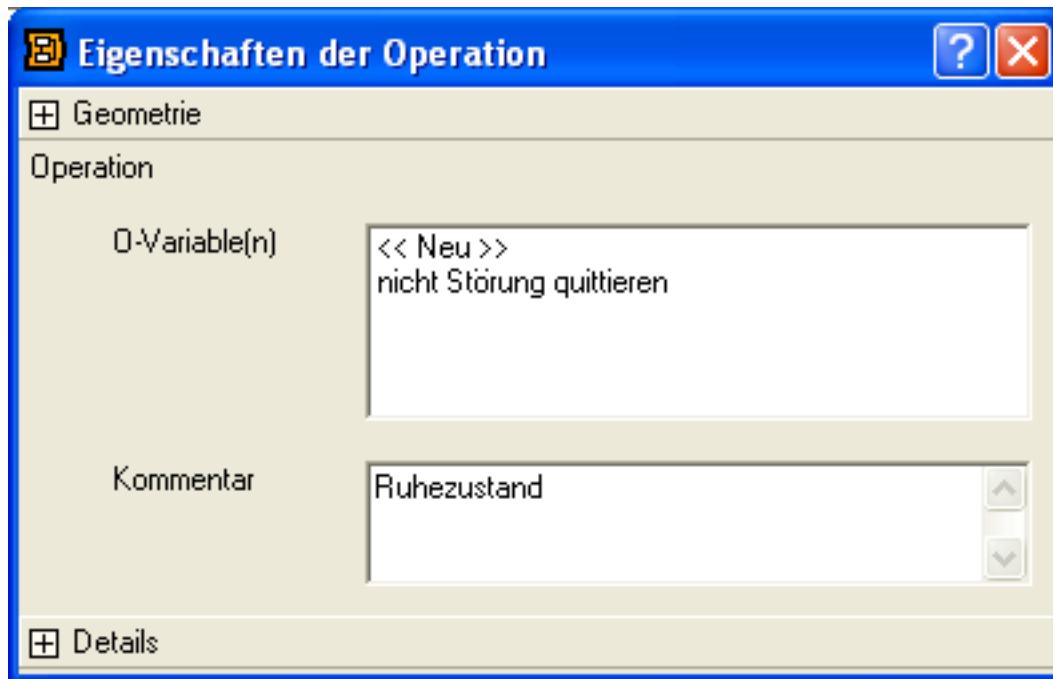
Mit Doppelklick oder über Kontextmenü **Bewertung wechseln** lässt sich die Bewertung umschalten.

Über den Eintrag <<Neu>> oder über Kontextmenü **Neue Operationsvariable** kann man eine neue Operationsvariable anlegen.

Wenn Sie eine Operationsvariable wieder löschen wollen, müssen Sie diese vorher in der Liste auswählen und dann entweder die Entf- Taste betätigen oder im Kontextmenü auf Löschen klicken.

Bei eingeblendeten Details wird die markierte Operationsvariable dort angezeigt . Der alternativ anzuzeigende Kommentar kann hier auch eingegeben werden.

Abbildung 4.6. Eigenschaften der Operation



4.9.4. Eigenschaften von Prozessvariablen

Das Feld 'Aussage' muss zuerst ausgefüllt werden. Die negierte Aussage passt sich automatisch an, indem ein 'nicht' davorgestellt wird. Wenn die negierte Aussage so nicht gefällt, kann sie von Hand geändert werden.

Die negierte Aussage soll das Gegenteil der technologischen Funktion ausdrücken. Dies hängt insbesondere von der Art der Funktion und der Signalerfassung ab.

Z.B. ist das Gegenteil von 'Tür offen' nicht unbedingt 'Tür geschlossen', wenn zwei Endschalter vorhanden sind. Bei einem Magnetventil mit nur einer Ansteuerung kann man 'Ventil geöffnet' und negiert 'Ventil geschlossen' schreiben.

Im Feld 'Funktion' kann man für die Variable die Funktionalität der Variablen im PRAP auswählen. Die Funktionen haben unterschiedliche Parameter, die in den nächsten bis zu drei Feldern eingetragen werden müssen. Die Art der Funktion und der Parameter wird im Hinweistext angezeigt.

In den Parameterfeldern kann über das Kontextmenü eine Variable ausgewählt oder auch neu angelegt werden. Die Variable wird an der Cursorposition eingefügt, so dass ggf. eine Nachbearbeitung nötig ist. Im Falle die Variable nur in der Datenbank enthalten ist, kann der Dialog zum Auswählen der Variablen über das Kontextmenü **Variable >aus DB wählen...** aufgerufen werden. Wenn eine Variable aus der Datenbank ausgewählt wurde, dann wird diese ebenfalls an die Cursorposition eingefügt und zur Variablenliste hinzugefügt.

In den Parameterfeldern gilt ST-Syntax. D.h. der hier eingetragene Text wird in den strukturierten Text direkt übernommen. Dabei entstandene Syntaxfehler werden also erst durch TeCOM entdeckt.

Wenn die Variable eingeblendet ist, wird hier die in einem Parameterfeld markierte Variable angezeigt. Bei mehreren Variablen genügt es, den Variablennamen mit der Maus anzuklicken.

Abbildung 4.7. Eigenschaften von Prozessvariablen

Aussage	ZA Schlüsselschalter Stellung Vor Ort
neg. Aussage	nicht ZA Schlüsselschalter Stellung Vor Ort
Funktion	direkt
Eingangsvar.	ZA_VOBED_VO
+ Variable	

4.9.5. Eigenschaften von Operationsvariablen

Die Eigenschaften der Operationsvariable ähneln sehr stark der Eigenschaften der [Prozessvariablen](#). Es können hier nur andere Funktionen ausgewählt werden.

Abbildung 4.8. Eigenschaften von Operationsvariablen

Aussage	Störung quittieren
neg. Aussage	nicht Störung quittieren
Funktion	direkt
Ausgang	_quitt
+ Variable	

4.9.6. Eigenschaften von Variablen

Die Eigenschaften einer Variable können auf zwei verschiedenen Wegen angezeigt werden. Zum einen die Auswahl einer Variablen in der Variablenansicht oder zum anderen durch Markierung der Variable in den Eingabefeldern der Variablen (siehe [Prozessvariablen](#)). Zum Anlegen einer neuen Variablen können Sie diese entweder einfach in die Liste der Variablenansicht eintragen oder siehe [Prozessvariablen](#).

Abbildung 4.9. Eigenschaften von Variablen

Name	EINLEGB_EIN
Typ	BOOL
Operand	A6.0
Kommentar	Einlegeband EIN
TeVar-ID	96
Anfangswert	
Verwendung	OUT
Parameter	<input type="checkbox"/>
Temp	<input type="checkbox"/>
Lokal	<input type="checkbox"/>

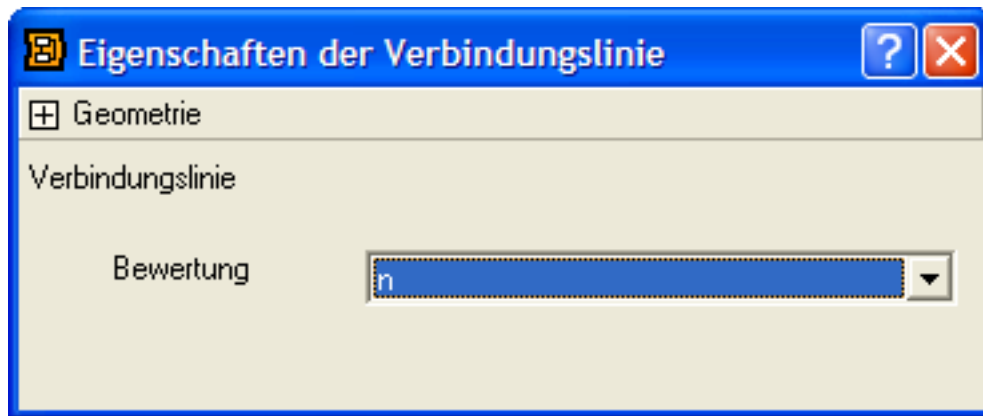
Wichtig

In TeCADLT sind die Eigenschaften nur lesbar

4.9.7. Eigenschaften von Verbindungslinien

Die Eigenschaften der Verbindungslinie werden durch Einfachklick bei geöffnetem Dialog oder durch Doppelklick bei ungeöffnetem Dialog auf der Verbindungslinie angezeigt. Hauptsächlich können hier die Koordinaten in der Punktliste manuell angepasst werden. Wurde die Verbindungslinie ausgehend von einer Bedingung gezeichnet, kann hier außerdem die Bewertung der Verbindungslinie auf "j" für ja oder "n" für nein gesetzt werden.

Abbildung 4.10. Eigenschaften der Verbindungslinie



Kapitel 5. Übersetzen

Über das Menü **Datei>ST erzeugen** rufen Sie die Übersetzungsfunktion auf. Vor der eigentlichen Codegenerierung wird der PRAP nochmals geprüft. Es können hier auch Fehler auftreten, die bei den Graphen- und Variablenprüfungsroutinen nicht gemeldet wurden.

Falls die Prüfung erfolgreich war, wird eine Datei mit dem strukturierten Text erzeugt. Die Datei trägt den gleichen Namen wie die PRAP-Datei mit der Endung '.st'.

Die generierte Datei kann eventuell direkt in ein Zielsystem übernommen werden. Normalerweise wird jedoch ein zusätzlicher Übersetzungslauf mit TeCOM nötig sein um für das jeweilige Zielsystem Code zu generieren.

So wird z.B. mit TeCOM_S7 eine Anweisungsliste für die S7-Steuerungen erzeugt.

Für den Aufruf weiterer Übersetzer wie TeCOM sowie zum betrachten der übersetzten ST-Datei eignet sich TeEdit sehr gut.

Wichtig

Diese Funktion ist in TeCADLT nicht verfügbar

Kapitel 6. Dateieigenschaften

Zur Pflege der Dokumentation und zur Kennzeichnung der Prozessablaufpläne sollten die Projektdaten und Übersetzungseinstellungen (Menüpunkt **Datei > Eigenschaften...**) eingegeben werden. Der Dialog der sich über diesen Menüpunkt öffnen lässt, enthält 5 Registereinträge (**Auftragsdaten**, **Übersetzung**, **Variablennamen**, **Operanden** und **Alle Eigenschaften**) zur Auswahl, die erst durch Klicken auf den jeweiligen Eintrag angezeigt werden. Blaue Einträge in den Eingabefeldern weisen auf eine globale bzw. übergeordnete Eigenschaften hin, wogegen schwarze auf lokal definierte hindeuten. Wenn Sie globale Eigenschaften verwenden wollen, müssen Sie gegebenenfalls die Anwendung Studio über den Menüpunkt **Module > Studio** (Studio muss vorher gestartet worden sein) aufrufen und im Projektextplorer mit der rechten Maustaste den Menüpunkt **Eigenschaften...** aufrufen. Dadurch müssen Sie nicht in jeden untergeordneten Prozess die Eigenschaften neu setzen. Das ist natürlich nur möglich, wenn Sie ein Projekt im Studio angelegt haben.

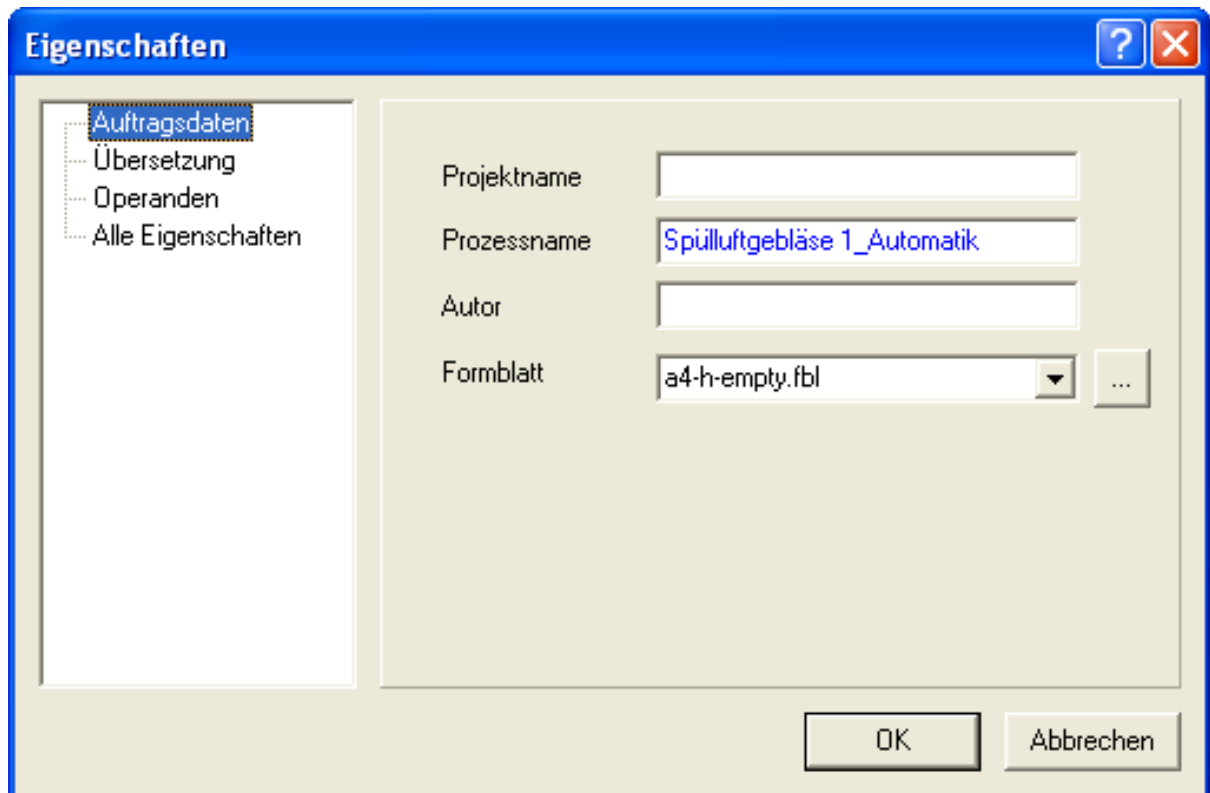
Hinweis:

Sie sollten die "prp-Datei" immer über den Projektextplorer von Studio öffnen, damit Sie über die Menüpunkte **Module > Studio** oder **Module > TeCAD** zwischen den Anwendungen einfach hin- und herschalten können.

6.1. Dialog Eigenschaften Auftragsdaten

Die Auftragsdaten dienen vor allem der Verwaltung und Beschriftung der Formblattfelder beim Drucken:

Abbildung 6.1. Eigenschaften Auftragsdaten



Im Register **Auftragsdaten** können folgende Einstellungen vorgenommen werden:

- Projektname

Der Projektname ist eine hierarchische Eigenschaft und muss nicht in jedem Unterordner neu eingetragen werden.

- Prozessname

Der Prozessname kann nur lokal eingestellt bzw. verändert werden.

- Autor

Der Autor ist eine hierarchische Eigenschaft und muss nicht in jedem Unterordner neu eingetragen werden.

- Formblatt

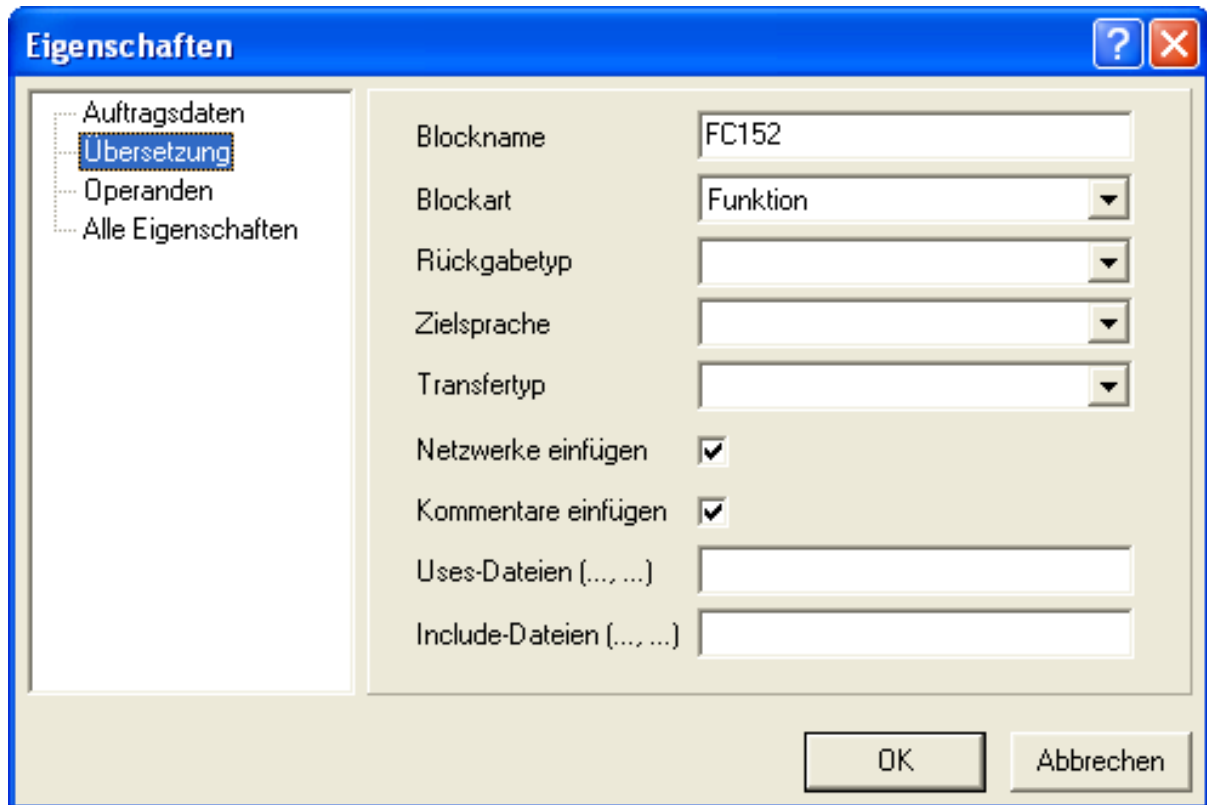
Das Formblatt ist eine hierarchische Eigenschaft und muss nicht in jedem Unterordner neu eingetragen werden. Es stehen nur die Formblätter zur Auswahl, die in Ihrem Projektverzeichnis vorhanden sind. Wenn Sie andere Formblätter verwenden wollen, dann können Sie diese über den rechten Button '...' in Ihr Projektverzeichnis importieren.

6.2. Dialog Eigenschaften Übersetzung

Die Eingabefelder der Übersetzung legen fest, wie und in welche Sprache der Quellcode erzeugt wer-

den soll.

Abbildung 6.2. Eigenschaften Übersetzung



Im Dialog können folgende Einstellungen vorgenommen werden:

- Blockname

Es muss mindestens ein Block angegeben werden. Dieser muss im Zielsystem bekannt sein. Bei STEP-7 kann z.B. direkt ein FC oder FB mit Nummer angegeben oder ein symbolischer Name verwendet werden. Im Blocknamen dürfen keine Leerzeichen o. ä. enthalten sein.

- Blockart

Dient zur Angabe, ob eine Funktion, ein Funktionsblock oder ein Programm generiert werden soll. Der Funktionsblock hat den Vorteil, dass interne statische Variablen deklariert werden können. Auch die Hilfsvariablen für die Automatenzustände u.ä. müssen dann nicht adressiert werden. Funktionsblöcke sind zwischen verschiedenen Zielplattformen gut austauschbar. Nachteile können in der schwierigeren Verwaltung oder schlechterer Performance auf dem Zielsystem bestehen. Für weitere Details zu den Auswahlmöglichkeiten lesen Sie bitte in der Hilfe zu strukturiertem Text (ST) weiter.

- Rückgabetyt

Kann nur bei Angabe einer Funktion in Blockart verwendet werden und legt den Rückgabetyt der Funktion fest. Meist kann hier VOID angegeben werden. Dies entspricht keinem Rückgabewert.

- Zielsprache

Es kann zwischen verschiedenen Zielsprachen gewählt werden. Die Anzahl der möglichen Zielsprachen wird ständig erweitert. Wenn Sie noch nicht wissen in welche Zielsprache übersetzt werden soll oder Ihre Sprache nicht in der Liste auftaucht, wählen Sie "ST". Der von TeCAD generierte strukturierte Text wird dann nicht weiter übersetzt.

- Transfertyp

Mit diesem Transfertyp können Sie festlegen, wohin der Quell-Code übermittelt werden soll.

- Netzwerke einfügen

Die Umsetzung der Optionen zu Netzwerken hängt davon ab, ob das Zielsystem Netzwerke kennt und TeCOM diese umsetzen kann.

- Kommentare einfügen

Wird eine Art von Know-How-Schutz benötigt oder will man einfach die Dateien verkleinern, kann man die Übernahme der Kommentare bei der Übersetzung abschalten.

- Uses-Dateien

Dient zum Einbinden von Datentypen und Funktionen, die in einer anderen ST-Datei deklariert wurden. Die Dateien können entweder vom Projektverzeichnis (Ordner mit der Projekt-Datenbank) oder im aktuellen Ordner der Prap- Datei gesucht werden. Dabei muss der Dateiname mit < > oder respektive mit " " eingeschlossen werden. Wenn sie nichts angegeben haben, dann wird im aktuellen Prap- Verzeichnis gesucht. Bei der Verwendung mehrerer Dateinamen, müssen diese durch Komma voneinander getrennt werden.

- Include-Dateien

Include-Dateien sind nötig, wenn eigene Funktionen oder Datenstrukturen verwendet werden sollen, die in einer anderen ST-Datei deklariert wurden. Dabei wird im Gegensatz zu Uses-Dateien der ST-Code in den bestehenden Quellcode kopiert. Die Angabe der Dateien erfolgt nach dem gleichen Prinzip wie bei Punkt /op0"Uses-Dateien".

6.3. Dialog Eigenschaften Variablenamen

Je nach Zielsystem und Benutzerwünschen können die automatisch generierten Variablen verschiedene Namen tragen. Die Generierung der Variablennamen soll jetzt von Einstellungen abhängig gemacht werden. Dafür kann für jede Variablenart getrennt ein Formatstring gesetzt werden. Als Parameter im Formatstring muss (außer bei _INIT) mindestens die Nummer X für den Platzhalter @num@ vorhanden sein. Weiterhin sind weitere Parameter möglich, die mit Werten von Eigenschaften gefüllt werden. Der Formatstring muss außerdem weitere Zeichen enthalten, die das eindeutige Auffinden der Variablen in der Variablen-tabelle auch bei veränderten Parametern ermöglichen.

Die Formatstrings sind hierarchische Eigenschaften mit einem Standardwert der dem bisherigem Format entspricht. Die hierarchische Gliederung erfolgt mit TeRANiS Studio in einem Verzeichnisbaum. Nähere Informationen zu hierarchischen Eigenschaften ist in der Hilfe von Studio beschrieben. Normalerweise sind diese Strings in den Resouceneigenschaften gespeichert und werden dort beim Anlegen einer neuen Resource in Abhängigkeit von deren Typ angelegt. Da die Formatstrings bei jeder

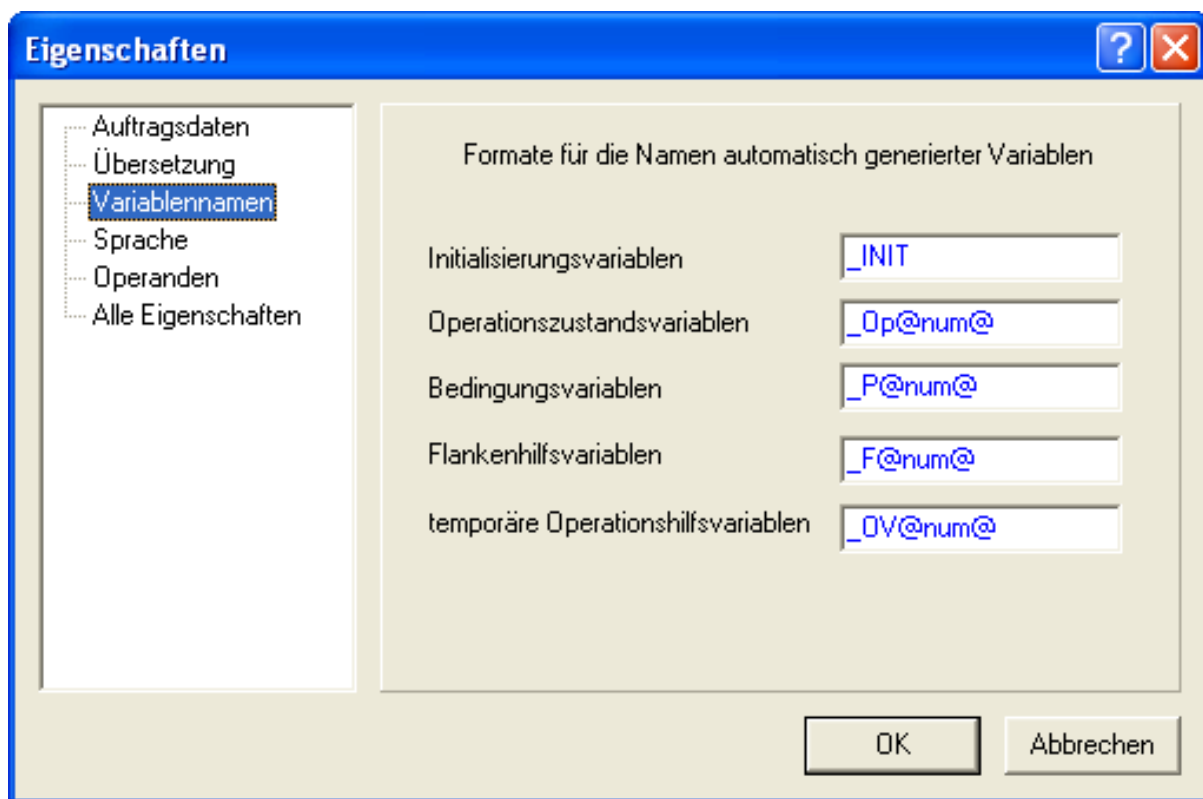
Änderung des Praps benötigt werden, ist ein Cache nötig. Es genügt, wenn die Strings beim Laden eines Prap aus den Eigenschaften gelesen werden und dann unverändert bleiben. Alle nicht lokalen Eigenschaften werden dabei bereits ersetzt, da sonst ständig ini-Dateien gelesen werden müssten.

Die Parameter haben folgende Platzhalter:

@num@ : Nummer der Variablen

@property@ : Wird ersetzt durch den Wert der Eigenschaft 'property'

Abbildung 6.3. Eigenschaften Variablennamen



Im Register **Variablennamen** können folgende Einstellungen vorgenommen werden:

- Initialisierungsvariablen

Die Variable ist normalerweise für mehrere Teilprozesse gültig. Es kann aber pro Resource mehrere Variablen geben um Teilbereiche getrennt initialisieren zu können. Die Variable ist nicht lokal. Der Name dieser Eigenschaft ist 'VARFORMAT_INIT' und kann auch unter dem Register 'Alle Eigenschaften' geändert werden.

- Operationszustandsvariablen

Lokale Variable, die den Zustand des PRAP speichert. Für den Platzhalter '@num@' steht die

Nummer der Operation. Der Name dieser Eigenschaft ist 'VARFORMAT_OP' und kann auch unter dem Register 'Alle Eigenschaften' geändert werden.

- Bedingungsvariablen

Lokale Variable, die das Ergebnis der Berechnung einer Bedingung speichert. Die Speicherung ist nur bei Zeitglied abgelaufen bzw. für die Online-Ansicht notwendig. Der Platzhalter '@num@' wird automatisch durchnummeriert. Der Name dieser Eigenschaft ist 'VARFORMAT_P'.

- Flankenhilfsvariablen

Lokale Variable zur Speicherung des Zustandes der Eingangsvariable zur Flankenbildung. Der Platzhalter '@num@' wird automatisch durchnummeriert. Der Name dieser Eigenschaft ist 'VARFORMAT_F'

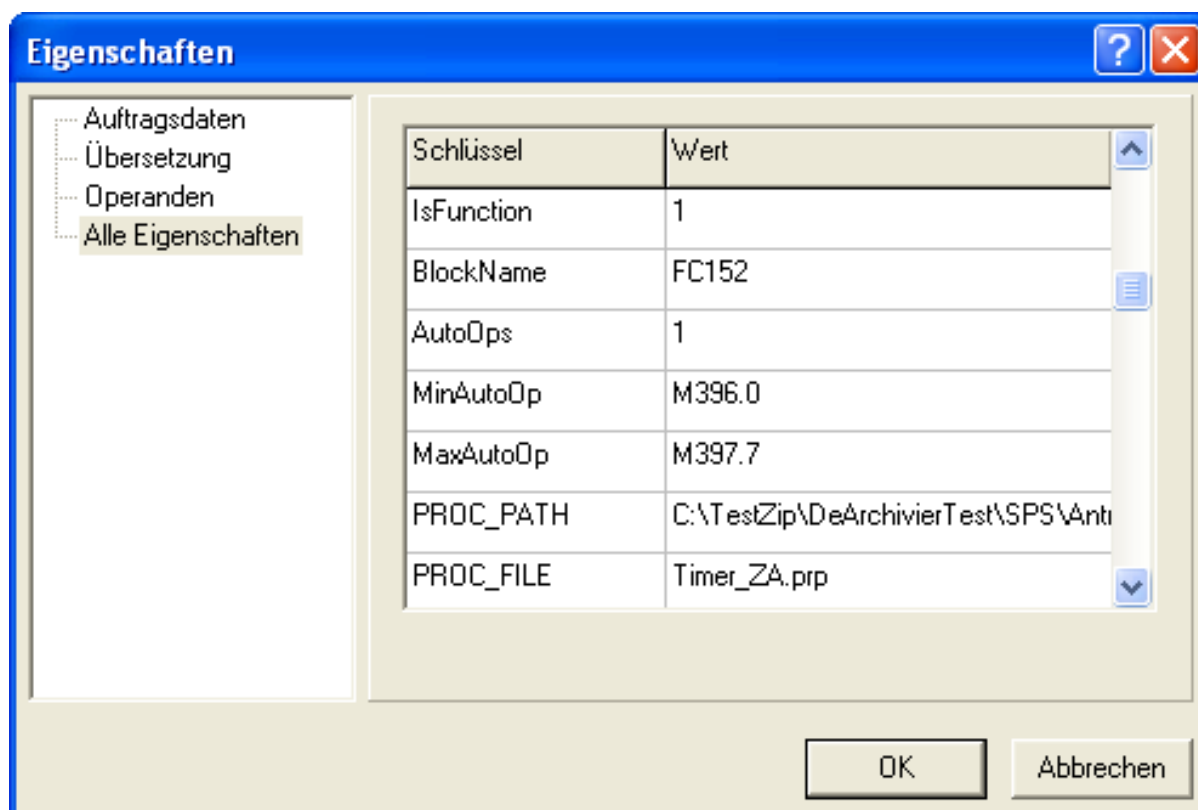
- temporäre Operationshilfsvariablen

Temporäre Variable um den Zustand einer Operationsvariable an die nachfolgende Aktion zu übergeben. Eine Speicherung ist nicht erforderlich. Die Variable ist nicht in der Variablen-tabelle enthalten und wird nur bei der Codegenerierung eingesetzt. Der Platzhalter '@num@' ist die Nummer der Operationsvariable. Der Name dieser Eigenschaft ist 'VARFORMAT_OV'

6.4. Dialog Eigenschaften Alle Eigenschaften

Bei der Auswahl von Alle Eigenschaften werden alle Eigenschaften aufgelistet und können bei Bedarf geändert, hinzugefügt oder gelöscht werden. Hinzufügen oder Löschen von Eigenschaften kann über das Kontextmenü der Tabelle ausgewählt werden.

Abbildung 6.4. Eigenschaften - Alle Eigenschaften



6.5. Dialog Eigenschaften Operanden

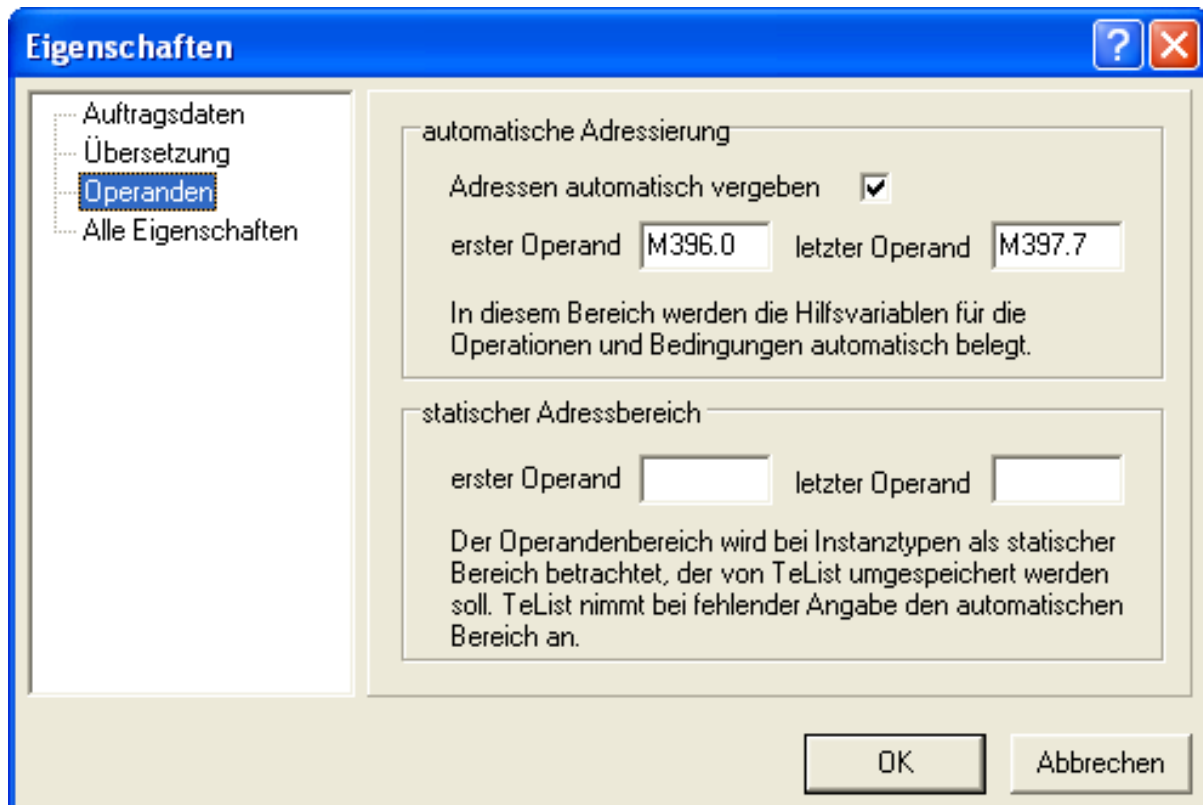
Eine Angabe bei den Operanden ist nur nötig, wenn:

- Sie die Operandenadressen von TeCAD nummerieren lassen wollen oder
- Sie einen Typenteilprozess für TeList erstellen wollen und TeList einen Variablenbereich als Block umspeichern soll.

Bei der automatischen Operandenvergabe kann TeCAD nur Bit-Adressen für BOOL-Variablen vergeben. Diese müssen im Format ...X.Y vorliegen. Y wird dabei immer von 0 bis 7 nummeriert und X dann weitergezählt. Die Syntax davor wird so übernommen wie angegeben. Diese Variante ist für SIMATIC-S7 Systeme getestet.

Die Angabe für TeList wird im entsprechenden Abschnitt genauer besprochen.

Abbildung 6.5. Eigenschaften Operanden



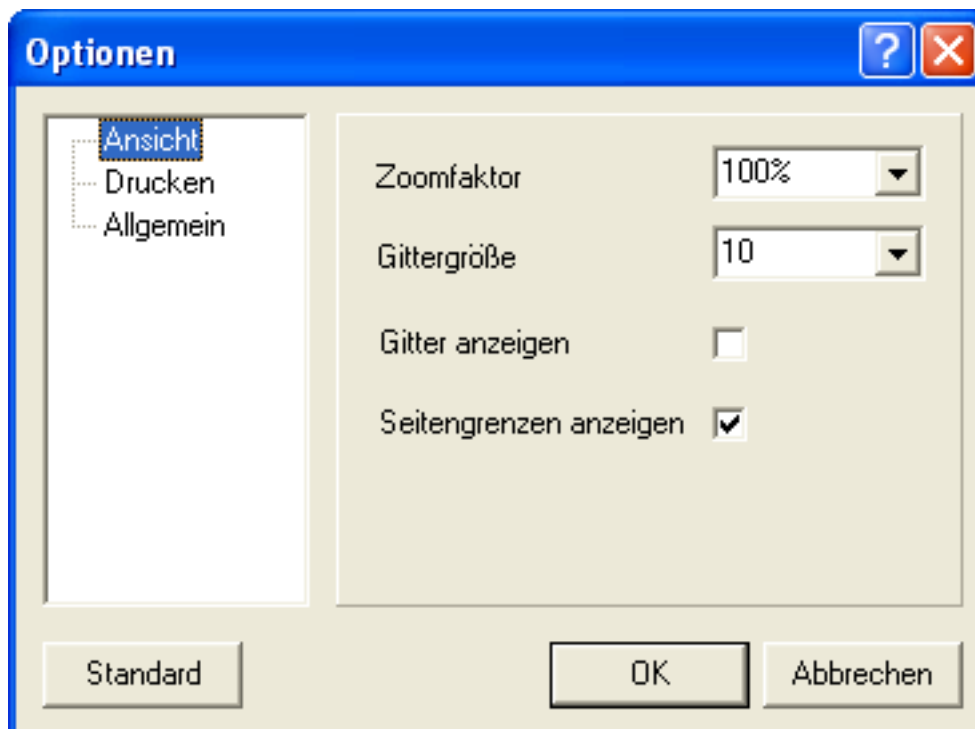
Kapitel 7. Optionen

Zum Einstellen der Optionen müssen Sie auf Menüpunkt **Extras > Optionen...** klicken. In diesem Dialog befinden sich 3 Registereinträge (Ansicht, Drucken und Allgemein), die durch Klicken auf den jeweiligen Eintrag angezeigt werden. Änderungen werden erst beim Verlassen des Dialogs durch die **OK** Schaltfläche übernommen. Im Optionsdialog sind für jedes Register Standardwerte definiert, die bei Betätigung der **Standard** Schaltfläche gesetzt werden. Dabei werden immer nur die Werte von dem aktuell angezeigten Register ersetzt.

7.1. Ansicht

In diesem Register können Eigenschaften der Anzeige eingestellt werden..

Abbildung 7.1. Optionen-Dialog Ansicht



Der Zoomfaktor stellt den Skalierungsfaktor bei der Anzeige der Prap-Grafik dar. Der selbe Wert kann auch über die Werkzengleiste und über +/- Tasten verändert werden.

Das Gitter dient zum sicheren Positionieren von Elementen auf dem Bildschirm. Standardmässig ist das Gitter auf 10 Einheiten eingestellt. Alle Mausbewegungen auf dem Bildschirm werden auf die Gittergröße gerundet. Wenn Elemente mit den Pfeiltasten bewegt werden, geschieht das ebenfalls im Gitterraster.

Es wird nicht empfohlen, die Gittergröße kleiner als 5 zu wählen oder sogar durch Gittergröße 1 ganz auszuschalten, da das Positionieren von Elementen sehr erschwert wird. Wenn das Gitter nach-

träglich vergrößert wird, kann es passieren das Elemente nicht mehr mit der Maus selektiert werden können. Die Grafikelemente können über den Menüpunkt **Ansicht > An Gittergröße anpassen** an das aktuell eingestellte Gitter angepasst werden.

Die Seitengrenzen werden durch gestrichelte Linien in der Grafik angezeigt.

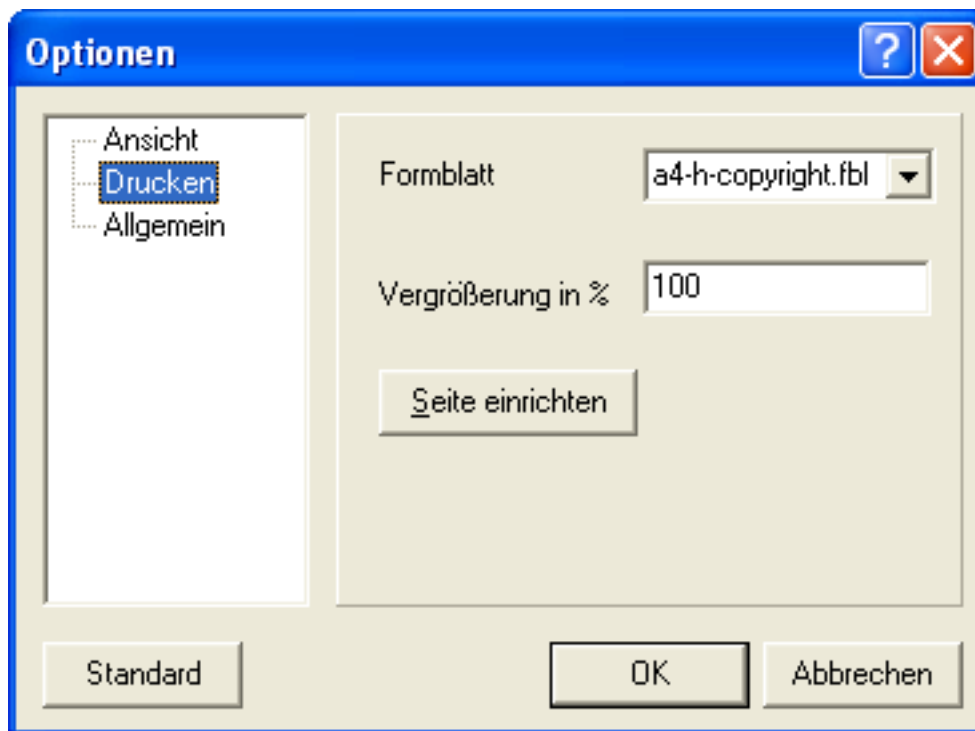
Die Größe einer Seite hängt vom verwendeten **Formblatt** ab, nicht vom eingestellten Drucker. Nur wenn kein Formblatt ausgewählt wurde, werden die Seitengrenzen des Druckers angezeigt.

7.2. Drucken

TeCAD kann Grafik und auch Listen auf Formblätter drucken.

Die Druckeinstellungen können unter **Extras > Optionen** eingestellt werden:

Abbildung 7.2. Optionen-Dialog Drucken



Die *.fbl Dateien müssen sich im gleichen Verzeichnis befinden, wie die TeCAD.exe-Datei. Es können auch eigene Formblätter entworfen werden. Dazu wird am besten ein vorhandenes mit einem Texteditor angepasst.

Die Vergrößerung beim Drucken bezieht sich auf die Plangrafik. Je grösser die Zahl, umso grösser wird der PRAP im Verhältnis zum Formblatt. Die angezeigten Seitengrenzen in der Grafik verschieben sich entsprechend.

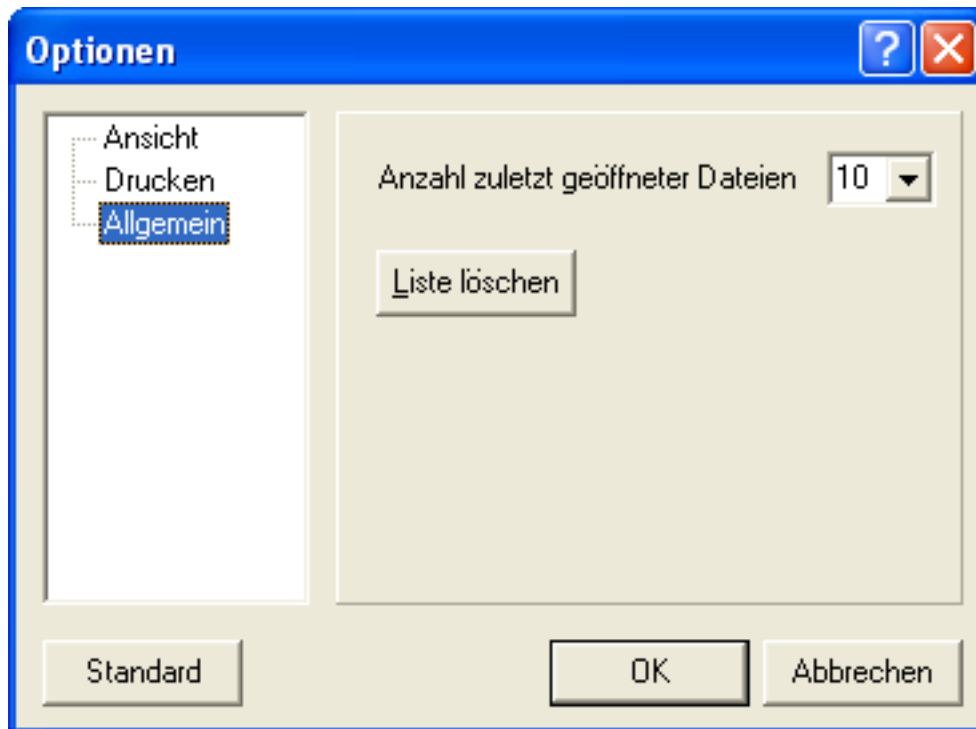
Die Prozessvariablenliste, Operationsvariablenliste und Variablenliste werden als Tabellen auch auf das eingestellte Formblatt gedruckt. Die Vergrößerung ist hier ohne Bedeutung.

Über den Menüpunkt **Datei > Seitenansicht** kann eine Druckvorschau angezeigt werden.

7.3. Allgemein

Der Optionen-Dialog Allgemein ...

Abbildung 7.3. Optionen-Dialog Allgemein



Im Eingabefeld "Anzahl zuletzt geöffneter Dateien" kann die Anzahl der zuletzt geöffneten Dateien eingestellt werden. Mit der darunter liegenden Schaltfläche "Liste löschen" können Sie vorhandene Dateinamen aus der Liste wieder löschen.


Kapitel 8. Drucken

Sowohl [Prozessablaufpläne](#), als auch Variablenlisten können auf von Windows unterstützten Druckern gedruckt werden.

Dazu stehen Ihnen folgende Möglichkeiten zur Auswahl:


- Aktuelle Ansicht drucken

Zuerst müssen Sie über die Tabellensymbolleiste  (siehe [Fensterauswahl](#))

die richtige Ansicht von der Datei aktivieren. Der Ausdruck kann dann durch klicken über Menü **Datei > Drucken...** oder über die Standardsymbolleiste  erfolgen.

- Alles drucken

Bei dieser Variante können alle Ansichten zu einer Datei über Menü **Datei > Alles drucken...** gedruckt werden. Hier müssen Sie nur eine Ansicht von der Datei aktivieren.

Bevor Sie mit dem Ausdruck beginnen, ist es ratsam sich die Druckvorschau über die Standardsymbolleiste  anzusehen. Hier erhält man eine weitestgehend mit dem zukünftigen Druckbild übereinstimmende Darstellung. Vorallem kann man die Darstellung des Formblattes und die Anzahl der Seiten beurteilen. In der Seitenansicht wird immer nur die aktuelle Ansicht angezeigt.

Druckoptionen:

Zum Einstellen der folgenden Optionen müssen Sie dazu den Optionsdialog [Drucken](#) über Menü **Extras > Optionen...** öffnen.

- Formblätter

Der Ausdruck erfolgt über [Formblätter](#), wenn in der Druckeinstellung nicht ausdrücklich 'Kein Formblatt' gewählt wurde. Das gewählte Formblatt muss mit der Druckausrichtung (hoch/quer) übereinstimmen.

- Zoom

In der Druckeinstellung kann außerdem ein Vergrößerungsfaktor für den Ausdruck angegeben werden. Dieser ist völlig unabhängig von der Bildschirmdarstellung und beeinflusst nur die PRAP-Grafik und nicht das Formblatt.

Bekannte Probleme:

Wenn der Druck von Text über den Rand von Grafikelementen hinausragt, sollten Sie die Elemente entsprechend vergrößern.

Wenn die Druckvorschau aber korrekt aussieht, sollten Sie die Einstellungen Ihres Druckertreibers überprüfen. Einige Drucker unterstützen das Drucken von Schrift über Vektordaten. Wenn Sie stattdessen Rasterdaten wählen, wird zwar der Ausdruck langsamer, aber das Ergebnis besser.

siehe auch: Optionsdialog [Drucken](#)

8.1. Formblatt

Formblätter werden zum Ausdruck verwendet.

Das Formblatt hat einen rechteckigen Bereich, indem die Zeichnung ihren Platz findet. Außerdem kann ein Formblatt verschiedene Linien sowie statische und [variable Texte](#) enthalten.

Mit dieser Version werden Formblätter für verschiedene Druckformate als "fbl-Dateien" mitgeliefert, die sich an geltenden DIN-Normen orientieren.

In einer späteren Version wird ein grafischer Formblatteditor enthalten sein. Dennoch können die Formblätter eigenen Wünschen angepasst oder neu erstellt werden. Dazu ist ein Texteditor und etwas Handarbeit nötig.

Siehe dazu: [Erstellen von eigenen Formblättern](#)

8.1.1. Variable Texte in Formblättern

Auf Formblätter können Texte gedruckt werden, die sich in Abhängigkeit von der Zeichnung ändern. Dazu gehören z.B. die Blattnummer, der Bearbeiter oder das Datum.

Folgende Schlüsselbegriffe sind fest vorgegeben:

- PAGE wird durch die aktuelle Blattnummer ersetzt
- MAXPAGE wird durch die maximale Blattnummer ersetzt
- TYPE wird durch die Art des Planes ersetzt (z.B. [Prozessablaufplan](#))

Folgende variable Schlüsselbegriffe (siehe untere Tabelle "vordefinierte Eigenschaften") sind frei definierbar und können über den Menüpunkt **Datei > Eigenschaften...** (siehe [Dateieigenschaften](#)) geändert werden.

Tabelle 8.1. vordefinierte Eigenschaften

AUTHOR	default: leer
PROJECT_NAME	default: Name der Datenbank
PROC_NAME	default: Name der Datei

8.1.2. Erstellen von eigenen Formblättern

Als Vorlage für eigene Entwürfe können die mitgelieferten Formblatt-Dateien genutzt werden. Die Daten sind im Textformat und können mit jedem beliebigen Texteditor (z.B. Notizblock) erstellt werden.

Wenn beim Einlesen des Formblattes Fehler auftreten, wird das Formblatt eventuell nur teilweise dargestellt. In diesem Fall müssen die Fehler in der Datei beseitigt werden.

Beschreibung des Formates

8.1.3. Dokumentation des Formblatt-Dateiformates

Die Formblattdatei ist eine normale Textdatei und kann z.B. mit TeEdit bearbeitet werden.

Alle Einheiten sind in 0,01 mm und müssen ganzzahlig sein.

Rechteckige Bereiche oder Endpunkte von Linien werden in der Form (x1,y1-x2,y2) angegeben.

x1 sollte kleiner als x2 und y1 kleiner als y2 sein.

Zuerst kommt der bedruckbare Bereich. Das ist der Bereich in dem die Nutzdaten (Prap) gezeichnet werden dürfen.

(3000 , 600-18000 , 25600)

Dann kommen Linien (Rahmen usw.). Die Zahl nach dem Komma ist die Liniendicke.

{

(2000 , 600-18000 , 600) , 12

(18000 , 600-18000 , 27500) , 12

(2000 , 27500-18000 , 27500) , 12

(3000 , 600-3000 , 27500) , 12

(2000 , 600-2000 , 27500) , 12

(3000 , 26300-18000 , 26300) , 12

(2000 , 14050-3000 , 14050) , 12

(3000 , 27200-18000 , 27200) , 8

(3000 , 26900-9800 , 26900) , 8

(3000 , 26600-18000 , 26600) , 8

(16800 , 26900-18000 , 26900) , 8

(16800 , 26300-16800 , 27500) , 8

(13300 , 27200-13300 , 27500) , 8

(4000 , 26300-4000 , 27500) , 8

(5300 , 26300-5300 , 27500) , 8

}

Jetzt kommen Texte. Texte werden in einen rechteckigen Bereich geschrieben. Dann kommt ein Schlüsselbegriff, der bei statischen Texten leer sein kann. Danach der Text. Dieser wird bei Schlüsselbegriffen bei der Ausgabe ersetzt.

Zum Schluss die Fontbeschreibung:

Format: "Facename",höhe,ausrichtung,dicke,breite,modifikation,drehung

"Facename"

z.B. "Arial"

höhe

in 1/10 mm

ausrichtung

Buchstabenkombination aus tlrbcvw

t Top l Left r Right b Bottom (Text einzeilig) c Center (horizontal) v vertikal zentriert (Text einzeilig) w Wortumbruch

dicke

von 100-400 (dünn bis dick)

breite

durchschnittliche Breite in 1/10 mm

modifikation

Buchstabenkombination aus ius

i Italic - Kursiv u Underline - Unterstrichen s StrikeOut - Durchgestrichen

drehung

in 1/10 Grad

{

(3100 , 26200-16000 , 26200) , " " , "Erstellt mit TeCAD © AVI-GmbH" , "Arial" , 200 , lb

(16900 , 26360-18000 , 26360) , "BlockName" , " " , "Arial" , 220 , l

(16800 , 27180-18000 , 27180) , " " , "Blatt" , "Arial" , 220 , cb

(16850 , 27250-18000 , 27250) , " " , "von" , "Arial" , 220 , c

(16900 , 27250-17900 , 27250) , "PAGE" , "##" , "Arial" , 220 , l

(16900 , 27250-17900 , 27250) , "MAXPAGE" , "##" , "Arial" , 220 , r

(9850 , 26580-16800 , 26580) , " " , "Zeichnungs-Nr.:" , "Arial" , 220 , lb

(11800 , 26580-16800 , 26580) , "DRAWING_NO" , " " , "Arial" , 220 , lb

```
(13350,27480-16800,27480),"TYPE","","Arial",220,lb  
(9850,27480-16800,27480),"PROJECT_NAME","","Arial",220,lb  
(9800,27200-16800,26600),"PROC_NAME","","Arial",400,cv  
}
```

Kapitel 9. Syntaxprüfung

Die Syntaxprüfung dient dazu, die logische Korrektheit und Vollständigkeit des **Prozessablaufplanes** sicherzustellen.

Es gibt zwei Stufen der Syntaxprüfung. Zuerst wird die Designprüfung durchgeführt und danach die Variablenprüfung. Die beiden Stufen können auch getrennt ausgeführt werden.

Um die Korrektheit des PRAP nachzuweisen, genügt es meistens die Designprüfung durchzuführen. Um den PRAP in ein Steuerungsprogramm zu übersetzen, ist es notwendig beide Prüfungen erfolgreich abzuschließen.

Designfehler

Variablenfehler

9.1. Designfehler

Folgende Fehler können bei der Designprüfung auftreten:

9.1.1. PRAP muß mindestens eine Operation enthalten

Sie haben einen **PRAP** erstellt, der keine **Operation** enthält. Ein PRAP benötigt mindestens eine Operation, da sonst keine Ausgaben erfolgen und der PRAP damit sinnlos ist.

Abhilfe: Operation einfügen

9.1.2. Keine Initialoperation festgelegt

Eine **Operation** in ihrem **PRAP** muss der Startpunkt für alle Aktionen sein. Diese Operation muss über das Menü **PRAP > Startoperation** festlegen markiert werden. Gehört diese Operation zu einer kombinatorischen Struktur, ist es gleich, welche Operation Sie aus der Struktur markieren.

Denn sofort nach dem Start wird die richtige Operation ausgeführt.

Aus Gründen der Vollständigkeit müssen Sie auch in einem PRAP, der nur eine kombinatorische Struktur enthält, eine Initialoperation markieren.

9.1.3. Ungültiger Knotentyp

Es wurde eine Verbindung zu einem Knoten gezeichnet, der weder Operation noch Bedingung oder Konnektor ist.

Abhilfe: Verbindung löschen

9.1.4. Verbindung zeigt ins leere

Dieser Fehler tritt auf, wenn eine Verbindungslinie nicht an einem Verbindungspunkt eines Knotens endet.

Abhilfe: Endpunkt der Verbindung an einen Verbindungspunkt führen.

Falls das nicht hilft, Verbindung löschen und neu zeichnen.

9.1.5. Linienanfangspunkt ist nicht belegt

Der Anfangspunkt der Linie sollte am Verbindungspunkt eines anderen Elements liegen.

9.1.6. Knoten ist doppelt im Pfad

Diese **Bedingung** kommt wiederholt in einem Weg vor. Der Weg muss zu einer **Operation** führen.

Abhilfe: Suchen der fehlerhaften Stelle, an der der Zyklus zustande kommt und Ändern der Verbindungen.

9.1.7. Bedingung ist in keinem Pfad enthalten

Eine **Bedingung** muss immer an eine **Operation** gekoppelt sein.

Außerdem muss über diese Bedingung ein Weg von einer Operation zu einer anderen führen.

Abhilfe: Wahrscheinlich ist diese Bedingung überflüssig.

9.1.8. Pfad enthält keine Bedingungen

Sie haben zwei **Operationen** direkt oder über **Konnektoren** direkt verbunden.

Abhilfe: **Bedingung** einfügen.

9.1.9. Ungültige Bewertung des Ausganges

Eine **Bedingung** muss zwei komplementär bewertete Ausgänge haben.

An einer Bedingung muss es zwei abgehende Verbindungslinien geben. Davon muss eine mit 'ja' und eine mit 'nein' bewertet sein.

Abhilfe: Eine Verbindung durch Doppelklick ändern

9.1.10. Bedingung muß zwei komplementär bewertete Ausgänge haben

Die Ausgänge einer **Bedingung** müssen unterschiedlich bewertet werden.

An einer Bedingung muss es zwei abgehende Verbindungslinien geben. Davon muss eine mit 'ja' und eine mit 'nein' bewertet sein.

Abhilfe: Fehlende Verbindung zeichnen oder eine Verbindung durch Doppelklick ändern

9.1.11. Es sind nur 2 Ausgänge bei einer Bedingung erlaubt

An einer **Bedingung** kann es nur zwei abgehende Verbindungslinien geben. Davon muss eine mit 'ja' und eine mit 'nein' bewertet sein.

Abhilfe: überflüssige Verbindung entfernen.

Hinweis: Vielleicht wollten Sie zwei **Operationen** gleichzeitig ausführen ?

Dann müssen Sie einen zweiten **PRAP** erstellen, da in einem PRAP nur eine Operation aktiv sein

9.1.12. Ausgänge müssen unterschiedlich bewertet werden

Eine **Bedingung** muss zwei komplementär bewertete Ausgänge haben.

An einer Bedingung muss es zwei abgehende Verbindungslinien geben. Davon muss eine mit 'ja' und eine mit 'nein' bewertet sein.

Abhilfe: Die Bewertung einer **Verbindung** ändern

9.1.13. Bedingung ist nicht mit einer Prozessvariable verknüpft

Diese **Bedingung** hat keine zugehörige **Prozessvariable**. So kann keine Auswertung stattfinden.

Abhilfe: Prozessvariable auswählen oder neu erzeugen.

9.1.14. Konnektor darf nur für ankommende Pfeile benutzt werden

Ein **Konnektor** kann entweder nur ankommende oder nur abgehende Verbindungen aufnehmen.

Dieser Konnektor hat schon eine ankommende Verbindung.

Abhilfe: Für die abgehende Verbindung muss ein zweiter Konnektor eingefügt werden.

9.1.15. Konnektor hat keinen passenden Partner

Zu diesem **Konnektor** gibt es kein Gegenstück, der eine abgehende Verbindung hat. Beide Konnektoren müssen den gleichen Text enthalten.

Abhilfe: Text ändern oder abgehende Verbindung zeichnen.

9.1.16. Es darf nur einen Ziel-Konnektor geben

In diesem **PRAP** gibt es mehrere **Konnektoren** mit abgehender Verbindung, die den gleichen Text enthalten, wie dieser Konnektor.

Abhilfe: Text ändern oder einen Konnektor löschen.

9.1.17. Konnektor darf nur einen Ausgang haben

Dieser **Konnektor** darf nur eine abgehende Verbindung haben.

Abhilfe: Eine Verbindung löschen.

9.1.18. Konnektor hat keine Verbindung

Dieser **Konnektor** hat keine Verbindungslinie.

Abhilfe: Verbindung zeichnen.

9.1.19. Konnektor ist in keinem Pfad enthalten

Über diesen **Konnektor** führt kein Weg.

Abhilfe: Konnektor löschen

9.1.20. Operation hat keine Übergangspfade

Jede Operation in einem PRAP muss einen Übergangspfad zu einer anderen Operation besitzen.

Nur, wenn der PRAP aus einer einzelnen Operation besteht, hat diese keinen Übergangspfad.

Die markierte Operation hat keinen Übergangspfad.

9.1.21. PRAP muß stark zusammenhängend sein

In einem **PRAP** muss jede **Operation** immer wieder erreichbar sein.

In ihrem PRAP gibt es eine oder mehrere Operationen, die nicht von jeder anderen Operation mittelbar oder unmittelbar erreichbar sind.

Abhilfen:

Haben Sie versucht mehrere PRAP in einer Datei zu zeichnen? Dann kopieren Sie jeden PRAP in eine extra Datei.

Gibt es in dem PRAP Verbindungen, die nur von einer Operation wegführen und keine, die zurückführen?

Dann müssen zusätzliche **Bedingungen** eingefügt werden, die eine Rückführung des Prozesses erlauben.

Tauchen die gleichen Bedingungen in einem Pfad doppelt auf, wird eine der Verbindungen nie durchlaufen.

Es existiert dann zwar eine Verbindungslinie, die aber nicht für einen Zustandsübergang genutzt werden kann.

Für den betreffenden Übergang müssen zusätzliche Bedingungen eingefügt werden, die einen Übergang ermöglichen.

9.1.22. Programmfehler

Dieser Fehler sollte nicht auftreten. Falls doch, waren wir doch nicht unfehlbar und es ist ein Fehler in der Datenstruktur aufgetreten. Wenn Sie uns helfen wollen, den Fehler zu finden, schicken Sie uns doch bitte diesen PRAP.

9.2. Variablenfehler

Folgende Fehler können bei der Variablenprüfung auftreten:

9.2.1. Leere Parameter

Für die gewählte Funktion fehlt ein notwendiger Parameter. Beispielsweise muss bei der Funktion 'Zeitglied erzeugen' ein Zeitglied und ein Zeitwert angegeben werden.

9.2.2. Variable nicht definiert

Die angegebene Variable ist nicht in der Variablenliste enthalten. Eventuell meinten Sie eine andere Variable? Falls nicht sollten Sie in der Variablenliste die Funktion **Extras>Variablen automatisch hinzufügen** wählen. Dies fügt alle fehlenden Variablen in der Liste ein. Sie müssen nur noch den Typ und evtl. die Adresse hinzufügen.

9.2.3. Variable hat nicht erforderlichen Typ

Einige Variablen müssen einen bestimmten Typ haben. Meist passiert dies bei der Funktion 'direkt', da die angegebene Variable den Typ BOOL haben muss.

9.2.4. Variable mehrfach beschrieben

Wenn Sie Variablen mehrmals in Operationsvariablen verwenden, darf die Variable nicht zeitgleich beschrieben werden. Im Moment wird nur geprüft, ob eine direkte Binärvariable nochmals beschrieben wird. Dies würde im erzeugten Code zu einer Fehlfunktion führen, da nur die letzte Operation den Wert der Variable setzen würde.

9.2.5. Variable sollte beschreibbar sein

Sie haben eventuell eine Konstante an einer Stelle angegeben, wo eine Variable als Ausgang verwendet werden soll.

9.2.6. Gleiche Adressen in Variablen

Alle Variablen in der Variablenliste müssen unterschiedliche Adressen haben. Wenn Sie ausserdem die Liste mit TeVAR abgleichen, wird auch die Überdeckung der Variablen geprüft.

9.2.7. Zeitglieder starten

Zu einem Zeitglied gehört immer eine Operations- und eine Prozessvariable, die das selbe Zeitglied verwenden. Sie dürfen kein Zeitglied abfragen welches in diesem Prap nicht gestartet wurde.

Kapitel 10. System Informationen

10.1. Technische Voraussetzungen

- Betriebssysteme: Windows 2000, Windows XP, Windows 2003

für TeList ist ein installiertes Microsoft-Excel ab Version 97 Voraussetzung, Microsoft-Word für die Dokumentation

(Prinzipiell ist die TeRANiS-Gruppe auf Systemen ab Windows 98 oder Windows NT lauffähig, ggf. müssen Treiber für die Datenbank (ADO-Jet) nachinstalliert werden. Es wird jedoch keine Garantie für die uneingeschränkte Benutzbarkeit übernommen.)
- PC, auf denen die angegebenen Betriebssysteme arbeiten, Bildschirm/Grafikkarte ab 800x600 Punkte
- Für die Online-Darstellung ist eventuell weitere Hard- und Software notwendig, die abhängig von der Zielplattform ist

Weitere Empfehlungen:

- möglichst hohe Bildschirmauflösung
- Microsoft -Excel und -Access können zur Erweiterung der Funktionalität von TeVar verwendet werden.
- Insbesondere für grössere Projekte sollte genügend Arbeitsspeicher vorhanden sein. (>128MB)

10.2. Installation und Deinstallation

Sofern Ihr System CDs automatisch abspielt, brauchen Sie nur die TeRANiS CD einzulegen.

Die Installation erfolgt durch Starten der Teranis.msi-Datei, die sich auf der TeRANiS-CD befindet, oder die Sie auf anderem Weg erhalten haben.

Um Platz auf der Festplatte zu sparen, können Sie bei der benutzerdefinierten Installation die Module auswählen die Sie benutzen möchten bzw. für die Sie eine Lizenz erworben haben. Funktionen für die keine Lizenz vorhanden ist, laufen im DEMO-Modus.

Die Deinstallation kann über den entsprechenden Eintrag im Startmenü oder in der Systemsteuerung unter der Rubrik Software erfolgen.

10.3. Hardwaremäßiger Softwareschutz

Das Produkt TeRANiS wird zusammen mit einem HASP-HL USB-Dongle ausgeliefert. Auf diesem Dongle sind die Funktionen entsprechend Ihrer erworbenen Lizenz freigeschaltet.

Besitzen Sie keinen Dongle oder ist eine Funktion nicht freigeschaltet, kann die Funktion nur eingeschränkt im Demomodus verwendet werden. Nicht freigeschaltete Funktionen dürfen nur zum Testen des Programmes und nicht für produktive Projekte verwendet werden.

Achtung ! Der Dongle muss die gesamte Zeit, während die Anwendung gestartet ist, eingesteckt bleiben, da er zu verschiedenen Zeitpunkten abgefragt wird.

10.3.1. Umfang der Lizenzen

Ihre erworbene Lizenz bezieht sich auf freigeschaltete Funktionen, die maximale TeRANiS-Version und die maximale Betriebssystem-Version auf der TeRANiS eingesetzt werden kann. Sie können mit dieser Lizenz also auch jede frühere TeRANiS-Version verwenden. Auch die Verwendung auf einer früheren Betriebssystem-Version ist möglich. Beachten Sie jedoch die Einsatzvoraussetzungen für TeRANiS.

10.3.2. Treiber

Die Treiber für den HASP-HL-Dongle werden bei der Installation von TeRANiS mit installiert, sofern Sie die Option nicht abgewählt haben. Die Treiber sind von Microsoft zertifiziert und sind Bestandteil von Windows-Update. Damit können die Treiber auch direkt aus dem Internet installiert werden.

10.3.3. Hilfsprogramme

Auf der Installations-CD finden Sie verschiedene Diagnose-Tools für den Donglebetrieb. Diese können auch von der Website des Dongle-Herstellers heruntergeladen werden: <http://www.alladin.de>

10.3.4. Updates

Um weitere Funktionen freizuschalten, eine höhere TeRANiS Version zu verwenden oder es auf einem neuen Betriebssystem einzusetzen müssen Sie ein Update erwerben. Dabei erhalten Sie normalerweise keinen neuen Dongle, sondern eine E-Mail, mit der Ihr Dongle umprogrammiert wird.

10.4. Einschränkungen der Demo-Version

Wenn keine Lizenz auf einem Dongle für eine bestimmte Funktion gefunden werden kann, schaltet das betreffende Modul in den Demomodus. Dieser Zustand wird beim Starten des Moduls in einer Meldung angezeigt. Um den Demomodus zu beenden muss das Modul neu gestartet werden. Ein nachträgliches Einstecken des Dongles hat keine Wirkung.

Im Demomodus haben die einzelnen Module unterschiedliche Einschränkungen. Alle Funktionen der Module sind zugänglich und können somit getestet werden.

Allgemeines

- Beim Starten bestimmter Funktionen erscheint eine Dialogbox, die auf den Demomodus hin-

weist. Diese muss von Hand quittiert werden.

- Beim Drucken erscheint ein Text der auf die Demoversion hinweist auf dem Ausdruck

TeCAD

- Die Variablenanzahl ist auf 30 beschränkt. Da Variablen teilweise automatisch angelegt werden ist damit auch der Planungsmodus begrenzt.
- In den übersetzten ST-Code ist ein Text eingefügt, der vor der Weiterverarbeitung manuell entfernt werden muss

TeList

- Die Anzahl von übersetzbaren Instanzen ist auf 2 beschränkt

TeVar

- Die Variablenanzahl ist auf 100 beschränkt. (Dies gilt nur für das Einfügen und Ändern von Variablen)
- Es wird nur ein frei deklariertes Datentyp übernommen.

TeCOM

- Die maximale Anzahl generierter Codezeilen ist auf 500 beschränkt

TeRANiS Studio

- Die Anzahl durchführbarer Transaktionen ist auf 10 beschränkt. Danach muss das Programm neu gestartet werden.

10.5. Bezugsquellen

TeRANiS wird von der ViDEC-GmbH vertrieben. Nähere Informationen erhalten Sie dort.













ViDEC Data Engineering GmbH Osterdeich 108 D-28205 Bremen Telefon: 0421 - 33 950 - 0 Telefax: 0421 - 33 795 - 61 E-Mail: Info@videc.de



















<http://www.videc.de>





Kapitel 11. Referenz

11.1. Menü und Schaltflächen

Tabelle 11.1. Menü und Schaltflächen

Menü	Tastenkürzel	Schaltfläche	Beschreibung
Datei > Neu	Strg+N		Erstellen einer neuen Datei an
Datei > Öffnen...	Strg+O		Öffnen einer Datei
Datei > Letzte Dateien öffnen			Öffnen einer zuvor geöffneten Datei
Datei > Speichern	Strg+S		Speichern der aktuellen Datei
Datei > Speichern unter ...			Speichern der aktiven Datei unter einem neuen Namen
Datei > Schliessen			Schliessen der aktiven Datei
Datei > Eigenschaften...			
Datei > ST erzeugen	F9		
Datei > Seitenansicht			
Datei > Druckereinstellungen...			
Datei > Drucken ...	Strg+P		Drucken der aktiven Datei
Datei > Alles drucken...			
Datei > Beenden	Alt+F4		Beenden von TeEdit
Bearbeiten > Rückgängig	Strg+Z		Letzte Aktion wieder rückgängig machen
Bearbeiten > Wiederherstellen	Strg+Y		Rückgängig gemachte Aktion wiederherstellen
Bearbeiten > Ausschneiden	Strg+X		Ausschneiden des markierten Textes
Bearbeiten > Kopieren	Strg+C		Kopieren des markierten Textes
Bearbeiten > Einfügen	Strg+V		Einfügen des markierten Textes
Bearbeiten > Löschen	Strg+Entf		Löschen des markierten Textes

Menü	Tastenkürzel	Schaltfläche	Beschreibung
			
Bearbeiten > Alles auswählen	Strg+A		Gesamten Text markieren
Ansicht > Neu zeichnen			
Ansicht > An Fenstergröße anpassen			
Ansicht > Onlineansicht			
Ansicht > An Gittergröße anpassen			
Ansicht > Symbolleisten			
Ansicht > Prapansicht			
Ansicht > Operationsvariablenliste			
Ansicht > Prozessvariablenliste			
Ansicht > Variablenliste			
Einfügen > Operation	Strg+O		
Einfügen > Bedingung	Strg+P		
Einfügen > Konnektor	Strg+K		
Einfügen > Verbindungslinie	Strg+L		
Einfügen > Kommentar	Strg+M		
Extras > Graph prüfen	Umsch+F8		
Extras > Variablen prüfen	Strg+F8		
Extras > Alles prüfen	F8		
Extras > Plan als Bild speichern...			
Extras > Optionen...			
Module > Studio	Umsch+F1		Öffnen der Anwendung Studio
Module > TeVar	Umsch+F2		Öffnen der Anwendung TeVar
Module > TeCAD	Umsch+F3		Öffnen der Anwendung TeCAD
Module > TeList	Umsch+F4		Öffnen der Anwendung TeList

Menü	Tastenkürzel	Schaltfläche	Beschreibung
Module > TeEdit	Umsch+F5		Öffnen der Anwendung TeEdit
Fenster > Überlappend			Anordnen der geöffneten Fenster überlappend
Fenster > Horizontal anordnen			Anordnen der geöffneten Fenster horizontal
Fenster > Vertikal anordnen			Anordnen der geöffneten Fenster vertikal
Fenster > Alle verkleinern			Minimieren der geöffneten Fenster
Fenster > Alle anordnen			
Hilfe > Hilfe zu TeCAD			Aufrufen der TeCAD-Hilfe
Hilfe > Hilfe zu Online-Funktionen			Aufrufen der Online-Hilfe
Hilfe > Hilfe zu ST			Aufrufen der ST-Hilfe
Hilfe > Info...			Aufrufen von Info

Stichwortverzeichnis

A

Allgemein, 50
Anweisungen, 18
Anweisungsfunktion, 13
Auftragsdaten, 40
Ausdrucksfunktion, 13 , 17

B

Bedingung, 5 , 33
Bedingungsstabilität, 10
Bezugsquellen, 65

C

CSV Datei, 29

D

Darstellung, 22
Dateieigenschaften, 40
Designfehler, 56
Drucken, 49 , 51

E

Eigenschaftsdialog, 40 , 41 , 43 , 45 , 46
Eigenschaftsfenster, 31
Elemente, 22 , 23
Entwurfsprozess, 19
Export, 24 , 29

F

Fehler
 Designfehler, 56 - 60
 Variablenfehler, 61 - 62
Fenster, 25 , 31
Flankenerkennung, 13 , 18
Formblatt, 51 , 52 , 52 , 52

G

Gittergröße, 49
Grafikelemente, 3 , 22
Grafikfenster, 25

I

Import, 30
Informationsflussmodell, 19
Installation und Deinstallation, 63

K

Kombinatorische Strukturen, 9
Kommentar, 6
Konnektoren, 6 , 33

M

Menü, 66
Modell, 20

N

Nichtbinäre Funktionen, 13

O

Operanden, 46
Operation, 3 , 34
Operationsvariablen, 7 , 36
Operationsvariablenliste, 26
Optionen, 48 , 50

P

Prozessablaufplan, 2 , 2 , 3 , 9 , 18 , 22
Prozessgrößen, 19
Prozessvariable, 7 , 35
Prozessvariablenliste, 26
Prüfroutinen, 12

R

Referenz, 66

S

Schaltfläche, 66
Softwareschutz, 63
Strukturen, 9
Strukturstabilität, 11
Syntaxprüfung, 56
System Information, 63

T

TeCAD, 1 , 1
Technische Voraussetzungen, 63
Text, 6

U

Uebersetzen, 39
Uebersetzung, 41
Undo, 24

V

Variable, 27 , 28 , 28 , 29 , 30 , 30
Variablen, 7 , 37
Variablenamen, 43
Variablenfehler, 56
Variablenliste, 26
Verbindungen, 23
Verbindungslinien, 37
Verbindungspunkte, 24

Z

Zeichnen, 3 , 23

Zeitdauerwerte, 17
Zeitglieder, 13
Ziehmarken, 23
Zwischenablage, 24