

**Auszug aus dem Buch**  
**"SPS-Programmierung mit IEC 1131-3"**  
© Copyright 1997 John/Tiegelkamp mit freundlicher  
Genehmigung der Autoren und des Springer-Verlags

## 4.1 Kontaktplan KOP

Die Sprache Kontaktplan (KOP) kommt aus dem Bereich der elektromechanischen Relaissysteme und beschreibt den Stromfluß durch die einzelnen Netzwerke der POE. Diese Programmiersprache beschränkt sich im wesentlichen auf die Bearbeitung boolescher Signale (1  $\equiv$  TRUE oder 0  $\equiv$  FALSE).

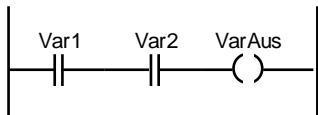
### 4.1.1 Netzwerke, grafische Elemente mit Verbindungen (KOP)

Wie in FBS gibt es auch bei KOP Netzwerke. Die Beschreibung der Grundelemente und deren Verbindungen kann dem Abschn. 1 entnommen werden.

### 4.1.2 Netzwerkaufbau in KOP

Ein KOP- Netzwerk wird an seiner linken und rechten Seite durch sogenannte Stromschienen begrenzt. Von der linken Schiene, sie besitzt den logischen Zustand 1 („Strom fließt“), führen Verbindungen den „Strom“ zu Elementen (Variablen), die abhängig von ihrem logischen Zustand die Weiterführung des Stroms zu den Ausgängen gestatten oder unterbinden.

0001 StartNetzwerk:  
(\* Netzwerk - Kommentar \*)



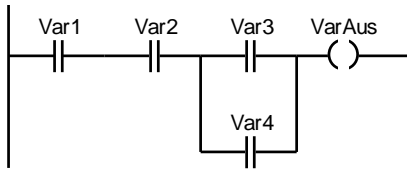
*Anweisungsteil mit linker  
und rechter Stromschiene*

**Bsp. 4.1.** Grafikelemente eines KOP- Netzwerks

### 4.1.3 Grafikobjekte in KOP

In KOP ergibt sich das Berechnungsergebnis durch die Anordnung der Einzel-elemente (inklusive Variablenamen), sowie die Art ihrer Verbindung.

Diese Elemente können hintereinander („in Serie“) oder parallel verbunden sein.



**Bsp. 4.2.** Die Elemente Var1, Var2, sind mit der Parallelschaltung aus Var3, Var4 in Serie geschaltet. Var1 bis Var4 werden als „Kontakte“, VarAus als „Spule“ bezeichnet.

Die in Bsp. 4.2 gezeigte Schaltung entspricht dem logischen Ausdruck:

$VarAus := Var1 \wedge Var2 \wedge (Var3 \vee Var4)$ .


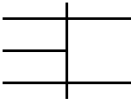
KOP-Netzwerke bestehen aus folgenden grafischen Objekten:

- 1) Verbindungen,
- 2) Kontakte und Spulen,
- 3) Grafikelemente zur Ausführungssteuerung (Sprünge),
- 4) Grafikelement zum Aufrufe von Funktionen und Funktionsbausteinen (Standard und Anwender-definiert), sowie die in Abschn. 1 genannten
- 5) Konnektoren.

Diese Grafikobjekte werden im folgenden erläutert.

### Verbindungen.

KOP kennt ähnlich wie FBS horizontale und vertikale Linien als Verbindungselemente sowie deren Kreuzungspunkte.

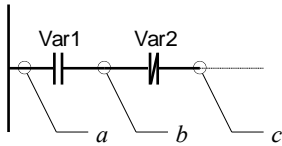
Grafikobjekt	Bezeichnung	Erklärung	Bsp. <sup>a</sup>
	Horizontale Verbindung	Die horizontale Verbindung reicht den links stehenden Wert (FALSE oder TRUE) nach rechts weiter.	
	Vertikale Verbindung mit horizontalen Verbindungen	Eine vertikale Verbindung verknüpft alle linken horizontalen Linien mit einem logischen OR- und reicht diesen Wert an alle rechten horizontalen Verbindungen weiter.	[0005]

<sup>a</sup> Die in [...] stehende Nummer bezieht sich auf Netzwerknummern in Bsp. 4.11

**Tab. 4.1.** Verbindungen in KOP

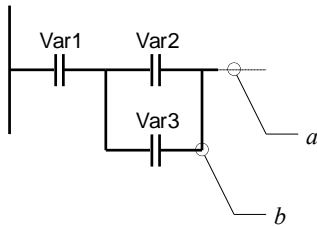
### Kontakte und Spulen.

Ein Kontakt verknüpft den Wert einer eingehenden Verbindung mit dem Wert der zugewiesenen Variablen. Die Art der Verknüpfung hängt von der Variante des Kontakts ab. Der errechnete Wert wird an die rechte Verbindung weitergegeben.



Var1	Var2	Not Var2	Wert von b	Wert von c
0	0	1	0	0
1	0	1	1	1
0	1	0	0	0
1	1	0	1	0

**Bsp. 4.3.** Ein Schließer und ein Öffner als UND-Verknüpfung: Die Verbindung a besitzt den Wert TRUE (Stromschiene immer auf 1). b berechnet sich aus „Var1 AND 1“, c aus „b AND NOT Var2“.







Var1	Var2	Var3	Wert von a	Wert von b
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	1	0
0	0	1	0	0
1	0	1	1	1
0	1	1	0	0
1	1	1	1	1

**Bsp. 4.4.** Drei Schließer als UND/ODER-Verknüpfung: Die Verbindung a besitzt den Wert „Var1 AND (Var2 OR Var3)“, b den Wert „Var1 AND Var3“.

Kontakte (Öffner und Schließer) regeln, abhängig vom Wert ihrer Variablen sowie ihrer Lage im Netzwerk, die Weitergabe von Werten in einem Netzwerk (die Variablen werden nicht geändert!). Die Zuweisung von Werten an die Ausgangsvariablen der Netzwerke übernehmen Spulen (den Variablen wird ein Wert zugewiesen).

Der Name der zum Kontakt gehörenden Variablen ist jeweils über dem Grafikelement zu finden. Die Variablen sind vom Datentyp BOOL (TRUE / FALSE). Die dazugehörigen Verbindungen leiten ebenfalls nur boolesche Werte weiter.

Kontakte werden unterschieden in *Öffner* und *Schließer*. Einen Schließer kann man sich als schließenden elektromechanischen Kontakt vorstellen: wird der in Ruhestellung (FALSE) offene Kontakt „aktiviert“ (TRUE), so schließt er sich und Strom kann fließen. Ein Öffner verhält sich genau umgekehrt: er ist normalerweise (TRUE) geschlossen und öffnet den Kontakt bei der Aktivierung (FALSE).

Grafikobjekt	Bezeichnung	Erklärung	Bsp. <sup>a</sup>
VarName 	Schließer	RechteVerbindung := LinkeVerbindung AND VarName, (kopiert den Wert von LinkeVerbindung nach RechteVerbindung, wenn der Wert der Variablen gleich TRUE; sonst FALSE).	[0007], [0009]
VarName 	Öffner	RechteVerbindung := LinkeVerbindung AND NOT VarName, (kopiert den Wert von LinkeVerbindung nach RechteVerbindung, wenn der Wert der Variablen gleich FALSE; sonst FALSE).	[0007], [0008], [0010]
VarName 	Kontakt mit positiver Flanken-erkennung	RechteVerbindung := TRUE, wenn * die Variable bei der letzten Auswertung den Wert FALSE besaß <b>und</b> * die Variable bei der aktuellen Auswertung den Wert TRUE besitzt <b>und</b> * LinkeVerbindung den Wert TRUE besitzt, := FALSE in allen anderen Fällen, (liefert genau (und nur dann) bei einem FALSE → TRUE Übergang der Variablen den Wert von LinkeVerbindung; sonst FALSE).	[0001], [0005], [0006]
VarName 	Kontakt mit negativer Flanken-erkennung	RechteVerbindung := TRUE, wenn * die Variable bei der letzten Auswertung den Wert TRUE besaß <b>und</b> * die Variable bei der aktuellen Auswertung den Wert FALSE besitzt <b>und</b> * LinkeVerbindung den Wert TRUE besitzt := FALSE in allen anderen Fällen, (liefert genau (und nur dann) bei einem TRUE → FALSE Übergang der Variablen den Wert von LinkeVerbindung; sonst FALSE).	

<sup>a</sup> Die in [...] stehende Nummer bezieht sich auf Netzwerknummern in Bsp. 4.11

**Tab. 4.2.** Kontakte in KOP

Spulen dienen dem Zuweisen von Werten an Variablen:

Grafikobjekt	Bezeichnung	Erklärung	Bsp. <sup>a</sup>
VarName —( )—	Spule	Variable <sup>b</sup> := LinkeVerbindung, (kopiert den Wert von LinkeVerbindung in die Variable).	[0007]
VarName —(/)—	Negative Spule	Variable <sup>b</sup> := NOT LinkeVerbindung, (kopiert den invertierten (Tausch FALSE <-> TRUE) Wert von LinkeVerbindung in die Variable).	
VarName —(S)—	Setze Spule	Variable <sup>b</sup> := TRUE, wenn LinkeVerbindung gleich TRUE, bleibt unverändert in allen anderen Fällen.	[0010]
VarName —(R)—	Rücksetze Spule	Variable <sup>b</sup> := FALSE, wenn LinkeVerbindung gleich TRUE, bleibt unverändert in allen anderen Fällen.	[0005]
VarName —(M)—	Gepufferte Spule	wie „Spule“. Die Variable liegt im gepufferten Bereich.	
VarName —(SM)—	Setze gepufferte Spule	wie „Setze Spule“. Die Variable liegt im gepufferten Bereich <sup>b</sup> .	[0008]
VarName —(RM)—	Rücksetze gepufferte Spule	wie „Rücksetze Spule“. Die Variable liegt im gepufferten Bereich <sup>b</sup> .	[0001]
VarName —(P)—	Spule mit positiver Flanken-erkennung	Variable <sup>b</sup> := TRUE, wenn * LinkeVerbindung bei der letzten Auswertung den Wert FALSE besaß <b>und</b> * LinkeVerbindung bei der aktuellen Auswertung den Wert TRUE besitzt, := FALSE in allen anderen Fällen, (speichert genau (und nur dann) bei einem FALSE → TRUE Übergang von LinkeVerbindung in die Variable TRUE, sonst FALSE).	

<sup>a</sup> Die in [...] stehende Nummer bezieht sich auf Netzwerknummern in Bsp. 4.11

<sup>b</sup> RechteVerbindung erhält stets den Wert von LinkeVerbindung

**Tab. 4.3.** Spulen in KOP (wird fortgesetzt)

	Spule mit negativer Flanken-erkennung	Variable <sup>b</sup> := TRUE, wenn * LinkeVerbindung bei der letzten Auswertung den Wert TRUE besaß <b>und</b> * LinkeVerbindung bei der aktuellen Auswertung den Wert FALSE besitzt, := FALSE in allen anderen Fällen, (speichert genau (und nur dann) bei einem TRUE → FALSE Übergang von LinkeVerbindung in die Variable TRUE, sonst FALSE).	[0009]
--	---------------------------------------	--	--------

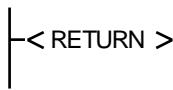
a Die in [...] stehende Nummer bezieht sich auf Netzwerknummern in Bsp. 4.11

b RechteVerbindung erhält stets den Wert von LinkeVerbindung

**Tab. 4.3.** (Fortsetzung)

### Ausführungssteuerung.

Um die Programmausführung zu beeinflussen, stehen sowohl Aufrufe zum Verlassen der POE, als auch zur gezielten Abarbeitung einer Netzwerk- Reihenfolge (Sprünge) zur Verfügung.

Grafikelement	Bezeichnung	Erklärung	Bsp. <sup>a</sup>
	(Unbedingter) Rücksprung	Verlasse POE und kehre zur aufrufenden POE zurück. Bei Erreichen des POE-Ende (Nach Ende des letzten Netzwerks) erfolgt ebenfalls ein Rücksprung.	[0011]
	Bedingter Rücksprung	LinkeVerbindung des RETURN-Elements gleich TRUE: Verlasse POE und kehre zur aufrufenden POE zurück, sonst: ohne Bedeutung.	

a Die in [...] stehende Nummer bezieht sich auf Netzwerknummern in Bsp. 4.11

b tnw steht für Teilnetzwerk und liefert der Anweisung einen booleschen Wert

**Tab. 4.4.** Grafikelemente zur Ausführungssteuerung in KOP (wird fortgesetzt)

	(Unbedingter) Sprung	Springe zu dem mit dem Bezeichner „Marke“ gekennzeichneten Netzwerk.	[0002], [0004]
$tnw^b \rightarrow \gg$ Marke	Bedingter Sprung	Linke Verbindung des Sprung Elements gleich TRUE: Springe zu dem mit dem Bezeichner „Marke“ gekennzeichneten Netzwerk, sonst: ohne Bedeutung.	[0001], [0008]

- a Die in [...] stehende Nummer bezieht sich auf Netzwerknummern in Bsp. 4.11
- b  $tnw$  steht für Teilnetzwerk und liefert der Anweisung einen booleschen Wert

**Tab. 4.4.** (Fortsetzung)

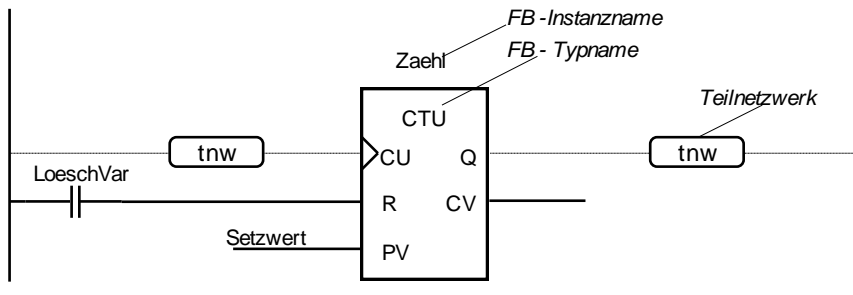
**Aufruf von Funktionen und Funktionsbausteinen.**

Die POE-Typen FUN und FB können von einem in KOP dargestellten Programm aufgerufen werden.

FBs können mehrere Ein- und Ausgangsparameter besitzen, Funktionen mehrere Eingangswerte und einen Ausgang (Funktionswert). Diese Ein- und Ausgänge (Formalparameter), sie müssen nicht alle vom Typ BOOL sein, sind innerhalb eines Rechtecks aufgeführt. Die zugeordneten Linien werden außen mit Aktualparametern (Variablennamen, Konstanten) versehen oder mit anderen Aus- bzw. Eingängen verbunden. Erweiterbare Funktionen besitzen keine Formalparameter. Dies entspricht der Verwendung wie in FBS.

Die Negation von Ein-/Ausgängen bzw. die Kennzeichnung von flankengetriggerten Eingängen bei FBs erfolgt wie in FBS.

*Funktionsbausteine* müssen mindestens einen booleschen Ein- und Ausgang besitzen, die eine direkte oder indirekte Verbindung zu den Stromschiene besitzen. Bei Standard-FBs sind dies der Ausgang Q und einer der booleschen Eingänge.



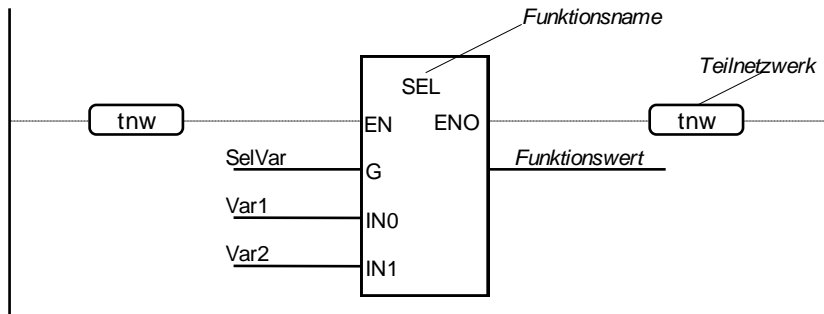
**Bsp. 4.5.** Aufruf des Standard-FBs CTU in einem KOP- Netzwerk



Der Parameter PV erhält als Aktualparameter die Variable Setzwert zugewiesen. Der Eingang R ist vorbeschaltet mit dem Kontakt LoeschVar. Ausgang CV ist unbeschaltet. „tnw“ kennzeichnet ein Teilnetzwerk, das ebenfalls zur Vor- und Nachbeschaltung von Ein- und Ausgängen zulässig ist.

Das linke tnw in Bsp. 4.5 könnte entfallen (Zaehl.R ist bereits mit der Stromschiene verbunden); das rechte tnw muß dagegen vorhanden sein, da CV vom Typ INT ist und nicht mit der Stromschiene verbunden werden kann.

*Funktionen* müssen einen EN / ENO Ein- und Ausgang besitzen, der die Ausführung der Funktion kontrolliert. Ist EN FALSE, erfolgt keine Ausführung der Funktion und ENO wird auf FALSE gesetzt. ENO kann auch als Fehlerstatus der Funktion verwendet werden. Diese beiden Parameter müssen direkt oder über ein Teilnetzwerk mit den Stromschiene verbunden sein. Inwieweit die übrigen, vor allem nicht-booleschen Parameter (z.B. IN0 in Bsp. 4.6) vorbeschaltet oder nur über Variablennamen versorgt werden, ist wie bei den FBs implementierungsabhängig.



**Bsp. 4.6.** Aufruf der Standard-Funktion SEL in ein KOP- Netzwerk

Der Funktionswert einer benutzerdefinierten Funktion wird wie bei den übrigen Sprachen durch eine Zuweisung festgelegt; d.h. mindestens eine Spule oder ein Ausgang eines FBs besitzt den Namen der (Funktions-) POE.

#### 4.1.4 Programmieretechnik in KOP

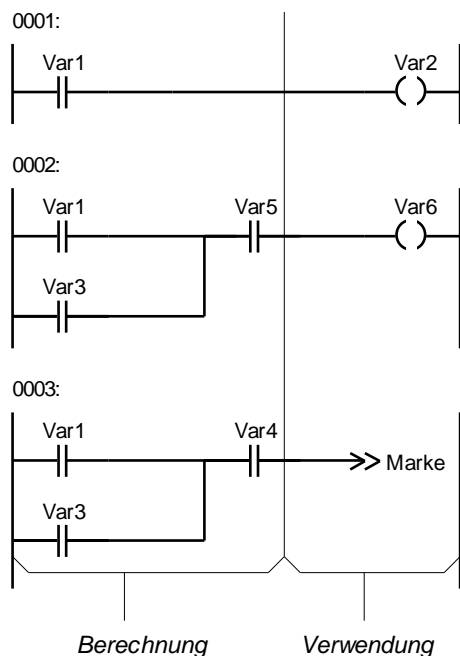
##### **Werteberechnung.**

Netzwerke werden der Reihe nach von oben nach unten abgearbeitet. Soll davon abgewichen werden, kann dies der Anwender über (Rück-) Sprunganweisungen programmieren.

Die in Funktionsplan angesprochene Netzwerk-Auswertung gilt auch für KOP.

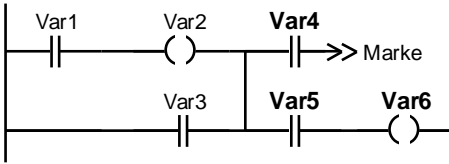
Die IEC 1131-3 erlaubt in einem Netzwerk die Weiterverarbeitung eines Wertes nach einer (Zwischen-) Zuweisung, wie in Bsp. 4.8 dargestellt. Bei den meisten Programmiersystemen besteht ein Netzwerk jedoch aus:

- einem linken Teil, in dem aus Variablen, Konstanten und Funktionen ein gewünschter Wert berechnet wird (Berechnung),
- einem rechten Teil, der das Ergebnis verwertet (Verwendung).



**Bsp. 4.7.** Struktur eines KOP- Netzwerks

0000:



**Bsp. 4.8.** Erlaubtes, aber unübliches KOP- Netzwerk mit Zwischenzuweisung, das die gleiche Funktion wie die drei Netzwerke von Bsp. 4.7 besitzt.

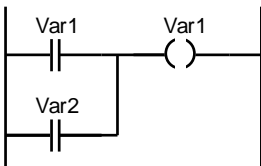
Wie man sieht, ist das in Bsp. 4.8 vorgestellte Netzwerk zwar kompakter als die 3 Netzwerke von Bsp. 4.7, seine Funktionalität jedoch wesentlich schwieriger zu interpretieren.

KOP wurde zur Verarbeitung von booleschen Werten konzipiert. Beinhaltet die POE viel Integer-Verarbeitung u.ä., sollte man möglichst eine andere Sprache verwenden. Solche Netzwerke sind zwar grundsätzlich in KOP programmierbar, aber sehr unübersichtlich (Verbindung zu Stromlinien, EN/ENO, ...).

Manche Systeme bieten deshalb Editoren an, die sowohl KOP als auch FBS-Netzwerke darstellen können. Sie stellen boolesche Netzwerke in KOP, die restlichen Netzwerke einer POE in FBS dar.

#### **Rückkopplungsvariable.**

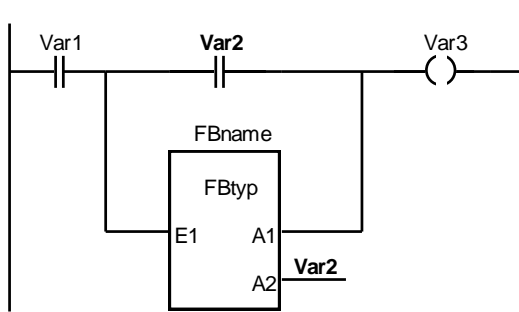
Soll der Wert einer Variablen in der nächsten Netzwerkberechnung als Eingang verwendet werden, spricht man wie in FBS von einer Rückkopplungsvariablen.



**Bsp. 4.9.** Rückkopplungsvariable (Var1) in KOP

Im Gegensatz zu FBS sind in KOP nur implizite Rückkopplungen, also keine expliziten Verbindungen von rechts nach links erlaubt.

Werden FBs und Rückkopplungsvariablen in einem Netzwerk verwendet, kann dies je nach SPS-System zu unterschiedlichem Verhalten führen:



**Bsp. 4.10.** Netzwerk mit FB und Rückkopplungsvariable (Var2) mit undefinierter Berechnungsreihenfolge; siehe dazu nachfolgenden Text.

In Bsp. 4.10 ist unklar, ob der Schließer Var2 den Parameterwert FBname.A2 der letzten oder aktuellen Netzwerkberechnung erhält. Eine vergleichbare Situation tritt auf, wenn die Variable Var4 in Bsp. 4.8 durch Var6 ersetzt wird. Auch dann ist unklar, ob der neue Schließer Var6 den aktuellen Wert der Spule Var6 erhält oder den der letzten Netzwerkauswertung.

Die IEC 1131-3 macht hierüber keine weiteren Aussagen. Um solche Unterschiede auszuschließen, sollte ein FB als eigenes Netzwerk angesehen werden. Sind Vor- und Nachverknüpfungen der Parameter notwendig, so dürfen die Variablen jeweils in den Vor- und Nachverknüpfungen nur als Kontakte vorkommen oder, falls sie als Spulen (schreibender Zugriff) auftreten, nur einmal im gesamten Netzwerk verwendet werden. Netzwerke wie in Bsp. 4.8 sollten entfallen.

Wie sich leicht ersehen läßt, kann das in Bsp. 4.10 beschriebene Reihenfolgeproblem auch für globale Daten zutreffen. Ist Var2 in FBname als EXTERNAL deklariert und wird innerhalb von FBname beschrieben, führt dies zu Berechnungsproblemen beim Schließer Var2. Beeinflußt ein FB durch EXTERNAL-Variablen seine Aufrufumgebung, so spricht man von einem FB *mit Seiteneffekten* (im Gegensatz zur *Seiteneffekt-Freiheit*).

Ist kein „Gedächtnis“ erforderlich und kann deshalb auf FBs verzichtet werden, sollten seiteneffektfreie Funktionen an Stelle der FBs verwendet werden. Ist dennoch ein FB erforderlich, sollte auf eine vorsichtige Verwendung von EXTERNAL-Variablen geachtet werden.

#### 4.1.5 Beispiel KOP: Bergbahn

Das folgende Beispiel gibt das Bergbahn- Beispiel aus Abschn. 4.1.5 (AWL) in der Darstellung KOP wieder. Dort ist auch die Funktionalität, die der FB leistet, beschrieben.

Die Netzwerke werden im Anschluß an ihre grafische Darstellung kommentiert.

##### FUNCTION\_BLOCK **BBSteuer**

###### VAR\_INPUT

BahnAn:        BOOL;   (\* Schalter zum Bergbahn Start \*)  
BahnAus:        BOOL;   (\* Schalter zum Einleiten des Betriebsendes \*)  
S1, S2, S3:     BOOL;   (\* Sensor fuer Stationsaufenthalt \*)  
TuerAuf:        BOOL;   (\* Schalter fuer Tueroeffnen. 1: Tuer oeffnen; 0: zu! \*)

###### END\_VAR

###### VAR\_IN\_OUT

StartStopp:     BOOL;   (\* Kabine faehrt: 1; steht: 0 \*)

###### END\_VAR

###### VAR\_OUTPUT

TuerOeffnen:    BOOL;   (\* Motor zur Tuer Oeffnung \*)  
TuerSchliessen: BOOL;   (\* Motor zum Tuerverschluss \*)  
EndeSignal:     BOOL;   (\* Warnsignal fuer Power Off \*)

###### END\_VAR

###### VAR

StationStopp:   CTU;    (\* Standard-FB (Zaehler) fuer Stationseinfahrten \*)  
TuerZeit:        TON;    (\* Standard-FB (Zeitverzoeger.) fuer Kabinenstart \*)  
RichtWeich:     SR;     (\* Standard-FB (FlipFlop.) zur Richtungsbestimmung \*)

###### END\_VAR

###### VAR\_RETAIN

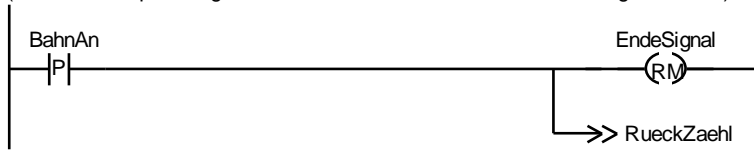
Richtung:        BOOL;   (\* aktuelle Richt: auf- oder abwaerts \*)

###### END\_VAR

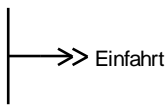
0001:

(\* System gestartet ? Das beim letzten Beenden aktivierte Endesignal \*)

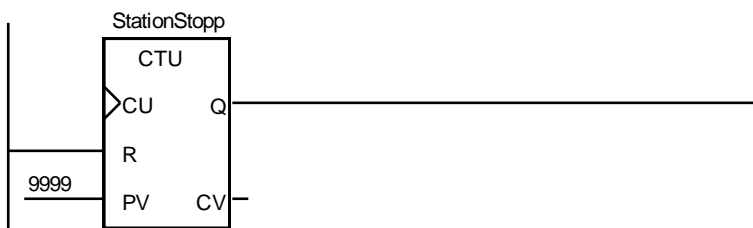
(\* wird nach Spannungswiederkehr und Aufnahme der Aktivitaet geloescht \*)



0002:



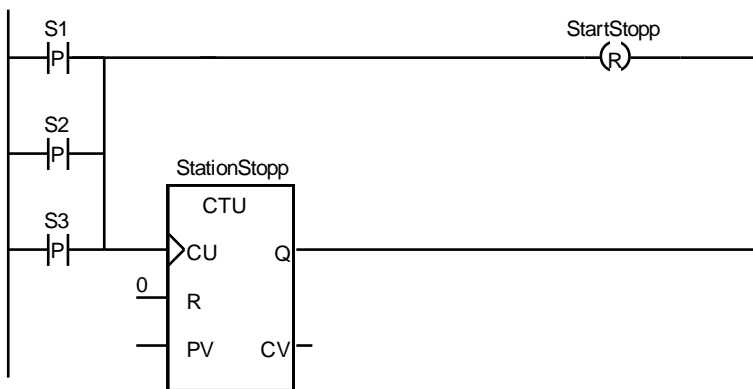
0003 RueckZaehl:  
(\* Ruecksetzen Zaehler \*)



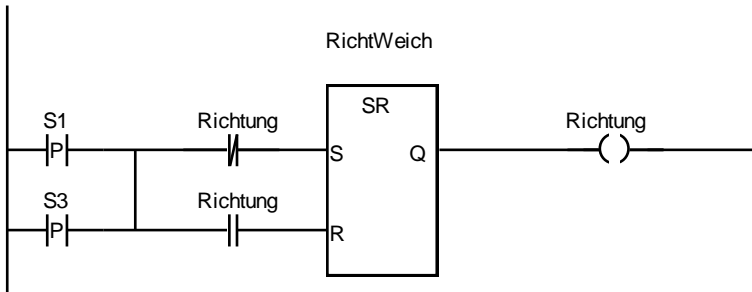
0004:



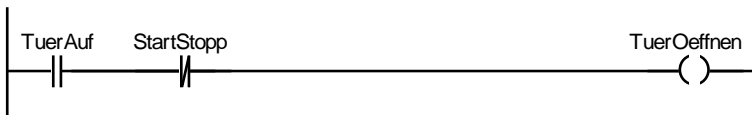
0005 Einfahrt:  
(\* Beim Einfahren der Kabine in eine Station wird \*)  
(\* der Zaehler hochgezaehlt und gestoppt \*)



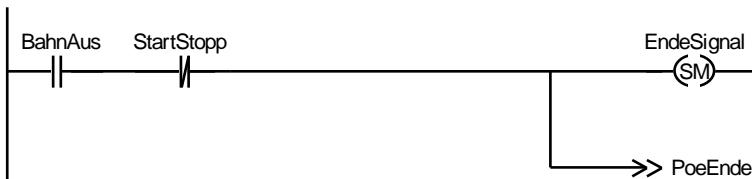
0006:  
 (\* Evtl. Richtungswechsel \*)



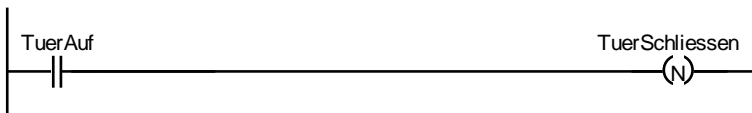
0007:  
 (\* Tueroeffnen nur nach Kabinenstopp und \*)  
 (\* nach Betaetigen des Tuerschalters \*)



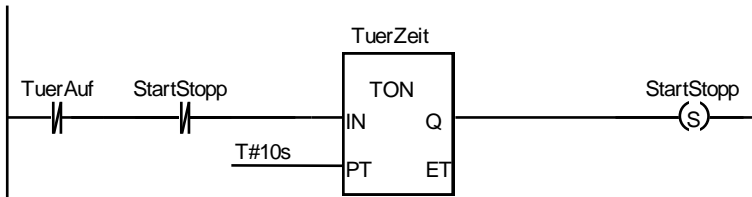
0008:  
 (\* Bahnende und in Station -> POE-Ende \*)



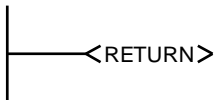
0009 KabineVerschliessen:  
 (\* Tuerschliessen nach Betaetigen des Tuerschalters \*)



0010:  
(\* Weiterfahrt 10 Sek. nach Betaetigen des Tuerschalters \*)



0011:  
(\* Kehre zur aufrufenden POE zurueck \*)



END\_FUNCTION\_BLOCK

**Bsp. 4.11.** KOP Programm zur Bergbahn- Steuerung, vgl. auch Bsp. 4.12.

### **Kommentierung der Bergbahn- Netzwerke.**

Netzwerk- Nummer	Kommentierung
0001	Bei einer positiven Flanke an der Variablen BahnAn (und nur dann), fließt TRUE. Dadurch wird EndeSignal auf FALSE gesetzt (in einem remanenten Bereich gepuffert); anschließend wird zum Netzwerk RueckZaehl gesprungen. Wurde keine Flanke erkannt, wird mit dem nächsten Netzwerk 0002 fortgefahren.
0002	Springe zu Netzwerk Einfahrt (Netzwerk 0005).
0003	Bei Erreichen dieses Netzwerks erhält der Eingang R fest TRUE. Damit wird die FB-Instanz StationStopp vom Typ CTU rückgesetzt. Die Überlaufvariable PV wird auf 9999 gesetzt. Ausgang Q wird nicht weiter abgeprüft. Fahre mit nächstem Netzwerk 0004 fort.
0004	Springe zu Netzwerk KabineVerschliessen (Netzwerk 0009).



0005      Sobald die Kabine in eine der drei Stationen einfährt, ändert sich eines der entsprechenden Signale S1, S2 oder S3 von FALSE nach TRUE (positive Flanke). Da eine OR- Verknüpfung vorliegt, wird genau in einem solchen Durchgang (und nur dann) CU auf TRUE gesetzt und zählt StationStopp um 1 hoch. Die 0 des Reset-Eingangs könnte auch weggelassen werden, da dieser Eingang in diesem Netzwerk nicht benötigt wird. Im Fall einer positiven Flanke wird die Kabine über StartStopp angehalten (durch „Rücksetze Spule“ erhält die Variable FALSE zugewiesen).

Liegt bei keinem der S1-S3 ein FALSE→TRUE- Übergang vor, erhält der CU Eingang des Zählers eine 0; StartStopp bleibt unverändert.

0006      Sobald die Kabine in die Tal- oder Bergstation einfährt, ändert sich eines der entsprechenden Signale S1 oder S3 von FALSE→TRUE. Da eine OR- Verknüpfung vorliegt, wird genau in einem solchen Durchgang (und nur dann) TRUE weitergeleitet. Besitzt Richtung den Wert 1, erhält der RESET-Eingang eine 1 und damit Richtung über den Flipflop-FB RichtWeich eine 0.

Im anderen Fall ändert Richtung über den SET-Parameter seinen Wert.

0007      Logische AND-Verknüpfung von TuerAuf und dem negierten Wert von StartStopp.

TuerAuf	StartStopp	Not StartStopp	TuerOeffnen
0	0	1	0
1	0	1	1
0	1	0	0
1	1	0	0

Damit wird die Tür nur geöffnet, wenn der Türschalter auf 1 betätigt wurde und der Aufzugsmotor steht.

0008      Logische AND-Verknüpfung von BahnAus mit dem negierten Werte von StartStopp.

BahnAus	StartStopp	Not StartStopp	EndeSignal
0	0	1	0
1	0	1	1
0	1	0	0
1	1	0	0

Solange der BahnAus Schalter auf TRUE (ein) geschaltet ist und die Kabine in einer Station steht, ertönt das EndeSignal. Dann wird auch nach POE-Ende gesprungen.

In allen anderen Fällen wird mit dem nächsten Netzwerk 0009

fortgefahren.

- 0009      Sobald der Schalter TuerAuf auf FALSE geschaltet wird, erhält TuerSchliessen einen TRUE→FALSE- Übergang. In genau dieser Situation wird TuerSchliessen auf TRUE gesetzt, ansonsten immer auf FALSE.
- 0010      Sobald der Türschalter auf „zu“ (TuerAuf auf FALSE) geschaltet wird und die Kabine sich dabei in einer Station (StartStopp auf FALSE) befindet, wird IN auf 1 gesetzt; damit startet die FB Instanz TuerZeit. Sind 10 Sekunden abgelaufen, ändert sich Q von FALSE auf TRUE. Ist die Tür immer noch geschlossen (TuerAuf auf FALSE), startet die Kabine (StartStopp auf TRUE); wurde die Tür vorher wieder geöffnet, wird TuerZeit rückgesetzt.  
Ist das Zeitglied nicht gestartet (Q-Ausgang ist FALSE) oder TuerAuf auf TRUE, bleibt StartStopp unverändert.  
Bei StartStopp handelt es sich um eine Rückkopplungs-Variable. Der im Öffner verwendete Wert stammt von einer Initialisierung oder einem vorher zugewiesenen Wert. Nach der Spulenzuweisung wird mit dem neuen Wert weitergearbeitet.
- 0011      Rückkehr zum aufrufenden Programm. Eine Schleifenprogrammierung zum POE-Anfang wäre falsch, da dieser FB (soweit parametrierbar) zyklisch mit neuen Eingangsparametern aufgerufen wird.