



# SCE Ausbildungsunterlage für die durchgängige Automatisierungslösung Totally Integrated Automation (TIA)

## TIA Portal Modul 040-010 Schrittkettenprogrammierung mit S7-GRAPH und SIMATIC S7-300

## Passende SCE Trainer Pakete zu diesen Unterlagen

Eine Übersicht über die aktuell verfügbaren SCE Pakete finden Sie unter: [siemens.de/sce/tp](https://www.siemens.de/sce/tp)

### Fortbildungen

Für regionale Siemens SCE Fortbildungen kontaktieren Sie ihren regionalen SCE Kontaktpartner [siemens.de/sce/contact](https://www.siemens.de/sce/contact)

### Weitere Informationen rund um SCE

[siemens.de/sce](https://www.siemens.de/sce)

### Verwendungshinweis

Ausbildungsunterlage für die durchgängige Automatisierungslösung Totally Integrated Automation (TIA) wurde für das Programm „Siemens Automation Cooperates with Education (SCE)“ speziell zu Ausbildungszwecken für öffentliche Bildungs- und F&E-Einrichtungen erstellt. Die Siemens AG übernimmt bezüglich des Inhalts keine Gewähr.

Diese Unterlage darf nur für die Erstausbildung an Siemens Produkten/Systemen verwendet werden. D.h. sie kann ganz oder teilweise kopiert und an die Auszubildenden zur Nutzung im Rahmen deren Ausbildung aushändigt werden. Weitergabe sowie Vervielfältigung dieser Unterlage und Mitteilung ihres Inhalts ist innerhalb öffentlicher Aus- und Weiterbildungsstätten die Zwecke der Ausbildung gestattet.

Ausnahmen bedürfen der schriftlichen Genehmigung durch die Siemens AG Ansprechpartner: Herr Roland Scheuerer [roland.scheuerer@siemens.com](mailto:roland.scheuerer@siemens.com).

Zu widerhandlungen verpflichten zu Schadensersatz. Alle Rechte auch der Übersetzung sind vorbehalten, insbesondere für den Fall der Patentierung oder GM-Eintragung.

Der Einsatz für Industriekunden-Kurse ist explizit nicht erlaubt. Einer kommerziellen Nutzung der Unterlagen stimmen wir nicht zu.

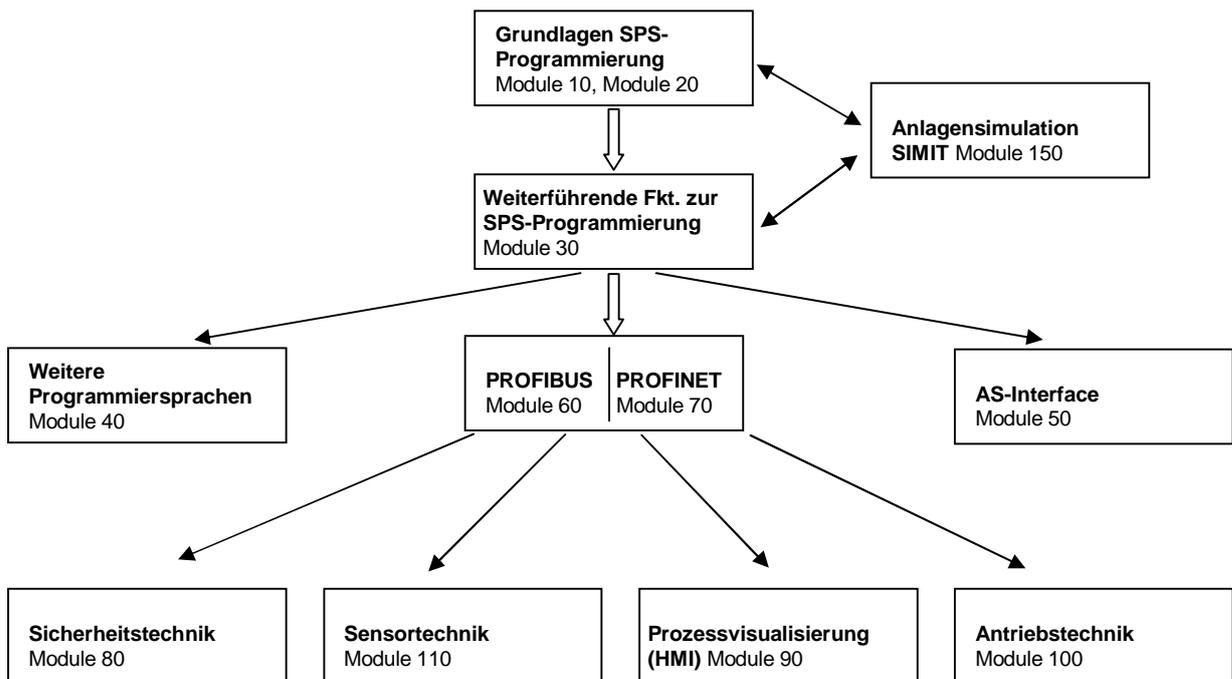
Wir danken der Fa. Michael Dziallas Engineering und allen Personen für die Unterstützung bei der Erstellung der Unterlage.

**SEITE:**

1.	Vorwort.....	4
2.	Hinweis zur Programmiersprache S7-GRAPH.....	6
3.	Ablaufsteuerungen .....	7
3.1	Zeitgeführte Ablaufsteuerung .....	7
3.2	Prozessabhängige Ablaufsteuerung.....	8
4.	Darstellungsmöglichkeiten von Bewegungsabläufen und Signalzuständen .....	9
4.1	Beschreibung der Steuerungsaufgabe.....	10
4.2	Aufschrieb in chronologischer Reihenfolge .....	11
4.3	Tabellenform.....	11
4.4	Kurzschreibweise .....	11
4.5	Bewegungsdiagramme .....	12
4.5.1	Weg- Schritt- Diagramm .....	12
4.5.2	Weg- Zeit- Diagramm .....	12
4.6	GRAFCET nach DIN EN 60848.....	13
5.0	Projektierung der Abschervorrichtung.....	15
5.1	Neues TIA Projekt erstellen und Hardware anlegen.....	15
5.2	PLC-Variablen eingeben.....	20
5.2	GRAPH Schrittkettenbaustein hinzufügen .....	21
5.3	Lokale Variablen der GRAPH Schrittkette .....	23
5.4	Aufbau der Schrittkette.....	27
5.4.1	Das Prinzip der Ablaufkette.....	28
5.5	Grafische Darstellung der Abschervorrichtung.....	29
5.6	Aktionen und Transitionen zuweisen.....	35
5.7	Schrittkette [FB1] im Main [OB1] aufrufen und beschalten.....	43
5.8	Projekt Abschervorrichtung_S7-300 in die Steuerung laden .....	45
6.	Testfunktionen.....	46
6.1	Beobachten der Schrittkette .....	46
6.2	Schrittkette im Testbetrieb.....	48
6.3	Synchronisation der Schrittkette .....	49

## 1. Vorwort

Das Modul SCE\_DE\_040-010 ist inhaltlich der Lehrinheit ‚**Weitere Programmiersprachen**‘ zugeordnet und stellt einen **Schnelleinstieg** in die Programmierung der SIMATIC S7 300 mit der Programmiersprache **S7-GRAPH** mit dem TIA-Portal dar.



### Lernziel:

Der Leser soll in diesem Modul das programmieren einer Schrittkettensteuerung mit dem grafischen Programmierwerkzeug S7-GRAPH erlernen. Das Modul zeigt in den folgenden Schritten die Vorgehensweise anhand eines ausführlichen Beispiels.

- Vorstellung der Arten von Ablaufsteuerungen und die Darstellungsmöglichkeiten von Bewegungsabläufen, Schaltzuständen, Signalfluss und Bewegungsdiagrammen.
- Erstellen eines einfachen Bewegungsablaufs dargestellt als Weg-Schritt-Diagramm und als Grafcet anhand eines Programmierbeispiels.
- Die dazugehörige Ablaufsteuerung wird als Schrittkettenprogramm in S7-GRAPH erstellt. Mit Hilfe der Test- und Diagnosefunktionen wird die Funktionsweise des erstellten Programms überprüft.

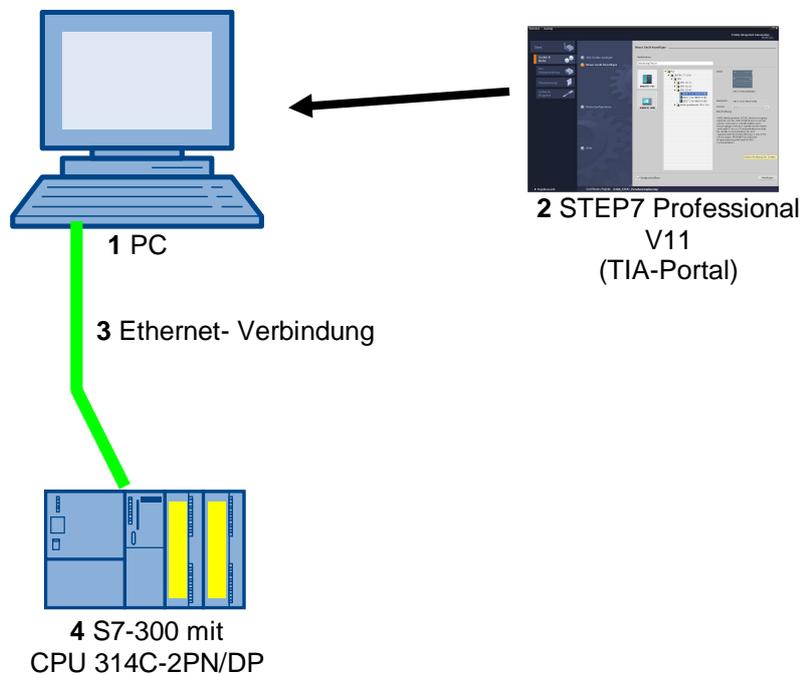
### Voraussetzungen:

Für die erfolgreiche Bearbeitung dieses Moduls wird folgendes Wissen vorausgesetzt:

- Kenntnisse in der Handhabung von Windows
- Grundlagen der SPS- Programmierung mit STEP 7 Professional V11 (z.B. Module 020- bis 030- )

## Benötigte Hardware und Software

- 1 PC Pentium 4, 1.7 GHz 1 (XP) – 2 (Vista) GB RAM, freier Plattenspeicher ca. 2 GB  
Betriebssystem Windows XP Professional SP3 / Windows 7 Professional / Windows 7 Enterprise / Windows 7 Ultimate / Windows 2003 Server R2 / Windows Server 2008 Premium SP1, Business SP1, Ultimate SP1
- 2 Software STEP7 Professional V11 SP1 (Totally Integrated Automation (TIA-) Portal V11)
- 3 Ethernet- Verbindung zwischen PC und CPU 314C-2 PN/DP
- 4 SPS SIMATIC S7-300, z. B. CPU 314C-2PN/DP mit 24DI/16DO.  
Die Eingänge müssen auf ein Schaltfeld herausgeführt sein.



## 2. Hinweis zur Programmiersprache S7-GRAPH

### S7-GRAPH

- ist seit November 2001 gemäß IEC 61131-3 und PLCopen Base Level zertifiziert.
- Siemens ist damit der erste Hersteller, der für die Schrittkettenprogrammierung S7-GRAPH (SFC - Sequential Function Chart) das PLCopen-Zertifikat erhalten hat.
- Programmierer die S7-GRAPH einsetzen, erstellen ihre Programme damit konsequent nach dem internationalen Standard IEC 61131-3.



### **Datenformate, Sprachelemente und grafische Darstellung entsprechen damit durchgehend der Norm IEC 61131-3.**

Mit der Programmiersprache S7-GRAPH wird der Funktionsumfang von STEP 7 um eine grafische Programmiermöglichkeit für Ablaufsteuerungen erweitert.

Mit S7-GRAPH können Sie Ablaufsteuerungen übersichtlich und schnell programmieren. Der Prozess wird dabei in Einzelschritte zerlegt und der Ablauf graphisch dargestellt.

In den Einzelschritten werden die auszuführenden Aktionen festgelegt.

Die Weiterschaltbedingungen zu den jeweiligen nächsten Schritt (Transitionen) können in der Programmiersprache KOP oder FUP erstellt werden.

## 3. Ablaufsteuerungen

Eine Ablaufsteuerung ist eine Steuerung mit zwangsläufig schrittweisem Ablauf, bei der das Weiterschalten von einem Schritt auf den programmgemäß nächsten Schritt abhängig von Weiterschaltbedingungen erfolgt.

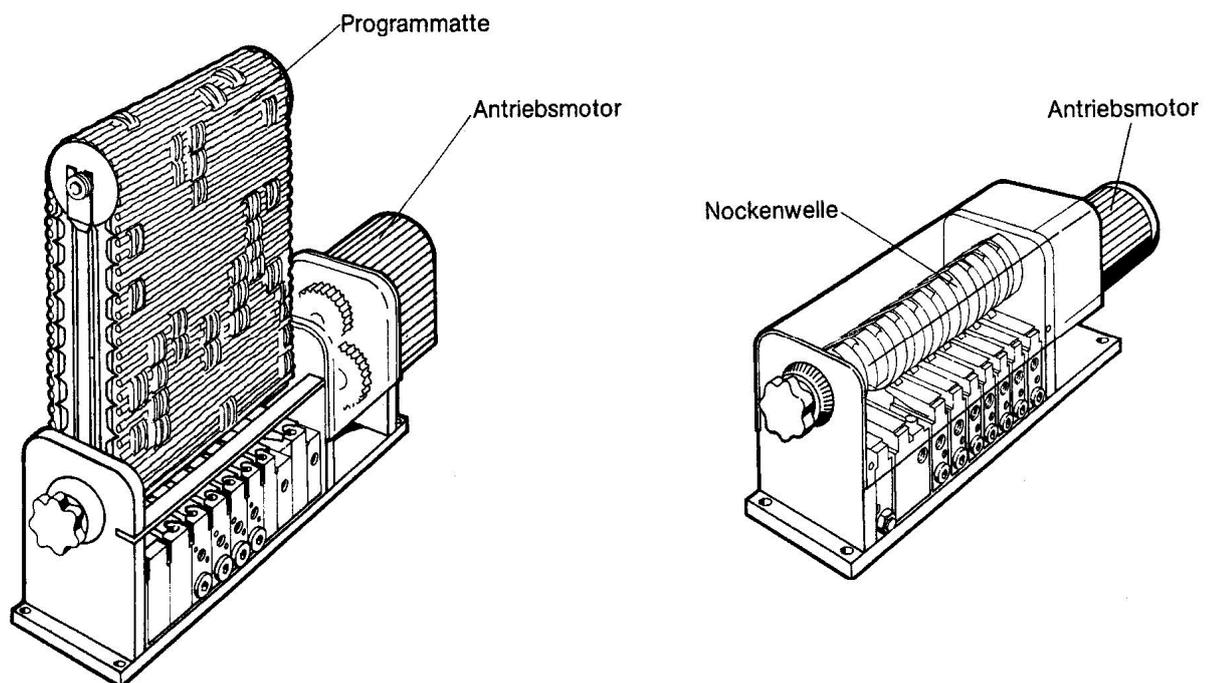
Die Schrittfolge kann in besonderer Weise programmiert sein, z. B. mit Sprüngen, Schleifen, Verzweigungen.

Mit S7-GRAPH können Ablaufsteuerungen programmiert werden, da der schrittweise Ablauf sehr einfach und schnell graphisch dargestellt werden kann.

Es gibt zwei Arten von Ablaufsteuerungen:

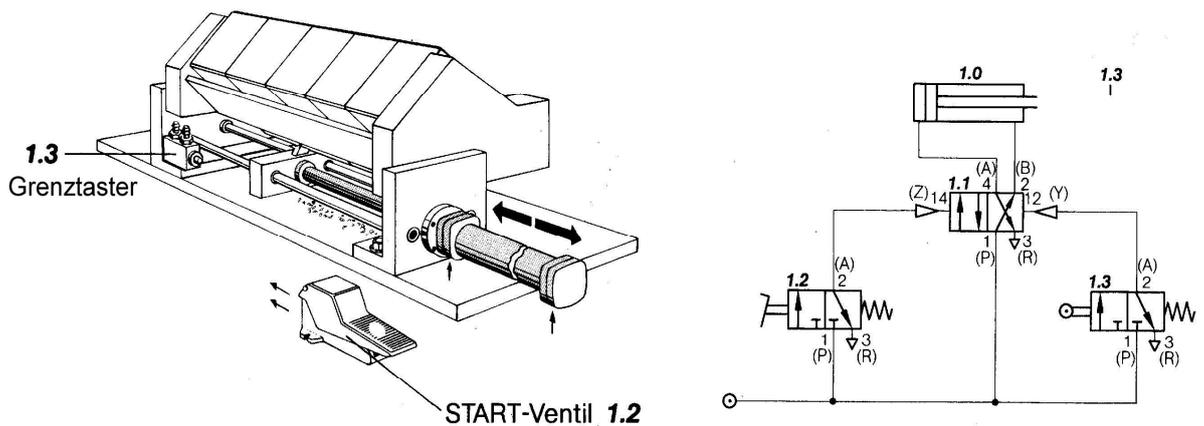
### 3.1 Zeitgeführte Ablaufsteuerung

Bei der zeitgeführten Ablaufsteuerung sind die Weiterschaltbedingungen nur von der Zeit abhängig. Zum Erzeugen der Weiterschaltbedingungen können z. B. Zeitglieder, Zeitzähler, Schaltwalzen oder Nockenmatten mit gleichbleibender Drehzahl benutzt werden.



## 3.2 Prozessabhängige Ablaufsteuerung

Bei der prozessabhängigen Ablaufsteuerung sind die Weiterschaltbedingungen nur von Signalen der gesteuerten Anlage abhängig. Zum Erzeugen der Signale werden Signalglieder wie Grenzta-ster, Schalter, Taster oder Sensoren verwendet. Die erfassten Signale können auch mit Zeitfunktionen verknüpft werden.



### Schneidvorrichtung

Bei Betätigung des Start-Ventils fährt der Zylinder der Schneidvorrichtung aus. Nach Erreichen der vorderen Endlage wird der Grenzta-ster betätigt und der Zylinder fährt selbsttätig wieder ein.

## 4. Darstellungsmöglichkeiten von Bewegungsabläufen und Signalzuständen

Das Zusammenwirken von Arbeits- und Steuerungselementen kann durch geeignete Darstellungsmöglichkeiten übersichtlich aufgezeigt werden. Selbst bei anspruchsvollen Aufgabenstellungen lassen sich die Zusammenhänge noch schnell und sicher erkennen. Zudem ermöglicht eine einfache Darstellung von Bewegungsabläufen und Schaltzuständen eine Verständigung verschiedenster Fachleute in größerem Rahmen.

### mögliche Darstellungsformen von Bewegungsabläufen und Signalzuständen

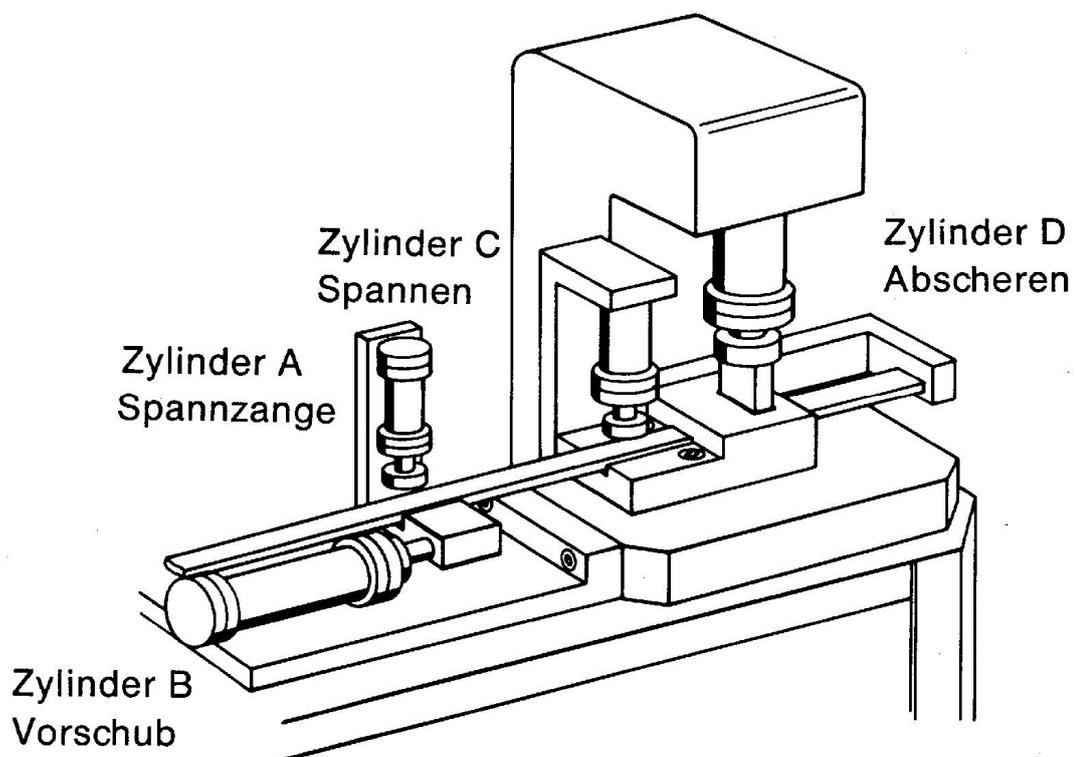
- **Beschreibung der Steuerungsaufgabe**  
In Form eines Textes wird der Steuerungsablauf beschrieben.
- **Aufschrieb in chronologischer Reihenfolge**  
In Kurzen Zeilen wird der Bewegungsablauf dargestellt.
- **Tabellenform**  
Der schrittweise Ablauf wird in eine Tabelle eingetragen.
- **Kurzschreibweise**  
Durch eine vereinfachte Darstellung der Bewegungen kann der Ablauf schnell und einfach dargestellt werden.
- **Bewegungsdiagramme**  
Mit Hilfe von Weg-Schritt bzw. Weg-Zeit Diagrammen wird der Bewegungsablauf graphisch dargestellt, zudem ist eine bessere Übersicht der Zusammenhänge gegeben.
- **Grafcet**  
Die Abläufe werden in Schritte und Transitionen unterteilt.  
Im Schrittfeld steht die alphanumerische Kennzeichnung.  
Kommentare dürfen in „Anführungszeichen“ daneben stehen.  
Die Weberschaltbedingung steht auf der rechten Seite der Transition.  
Der Punkt bzw. der dafür verwendete Stern beschreibt eine UND- Verknüpfung, das Plus-Zeichen beschreibt eine ODER- Verknüpfung.  
Negationen werden mit einem Strich über dem Variablennamen beschrieben.

Anhand eines Programmbeispiels sollen die verschiedenen Darstellungsmöglichkeiten erläutert werden.

#### 4.1 Beschreibung der Steuerungsaufgabe

Es soll eine Steuerung für eine Abschervorrichtung entworfen werden. Durch das Zusammenwirken einer Zuführeinheit und einer Schervorrichtung soll Stangenmaterial abgeschnitten werden. Die Zuführung erfolgt durch den Vorschubzylinder (Zylinder B), der beim Vor- bzw. Rückhub die pneumatische Spannzange (Zylinder A) mitbewegt. Ist das Material gegen einen Festanschlag eingeschoben, wird es durch die Spannvorrichtung (Zylinder C) festgehalten. Danach kann der Abschervorgang beginnen (Zylinder D) und gleichzeitig das Öffnen der Spannzange (Zylinder A) erfolgen. Ist die Spannzange (Zylinder A) geöffnet so erfolgt das Zurückfahren (Zylinder B) in Ausgangsstellung. Ist der Abschervorgang beendet (Zylinder D) und hat die Zuführeinheit die Ausgangsstellung erreicht, so wird die Spannvorrichtung (Zylinder C) geöffnet und es kann ein neuer Arbeitsablauf begonnen werden.

Der Start wird durch die Betätigung der Start-Taste ausgelöst, wenn sich alle Zylinder in der hinteren Endlage befinden.



## 4.2 Aufschrieb in chronologischer Reihenfolge

Zylinder A fährt aus und schließt die Spannzange,  
 Zylinder B fährt aus und schiebt das Material bis zum Anschlag,  
 Zylinder C fährt aus und spannt das Stangenmaterial in der Schervorrichtung,  
 Zylinder A fährt ein (die Spannzange geöffnet) und Zylinder D fährt aus (Abscheren),  
 Zylinder B fährt ein (die Vorschubeinheit fährt zurück) und Zylinder D fährt ein,  
 Zylinder C fährt ein und öffnet die Spannvorrichtung.

## 4.3 Tabellenform

Schritt	Zylinder A	Zylinder B	Zylinder C	Zylinder D
1	ausfahren	-	-	-
2	-	ausfahren	-	-
3	-	-	ausfahren	-
4	einfahren	-	-	ausfahren
5	-	einfahren	-	einfahren
6	-	-	einfahren	-

## 4.4 Kurzschreibweise

Für den Bewegungsablauf ist es oft unerheblich welche Aufgaben mit einer Bewegung erfüllt werden, so kann ein Bewegungsablauf auch für unterschiedlichste Steuerungen verwendet werden. Bei umfangreicheren Steuerungen sollte der Bewegungsablauf zuerst in der Kurzschreibweise beschrieben werden, da hier eine schnelle Übersicht der Bewegungen gegeben wird.

Bei der Kurzschreibweise werden den Bewegungen Bezeichnungen zugeordnet.

- Bezeichnung für das Ausfahren oder den Vorhub eines Zylinders: +
- Bezeichnung für das Einfahren oder den Rückhub eines Zylinders: -
- Für Motoren können M+ für Rechtslauf, M- für Linkslauf und M\* für Stopp verwendet werden. Parallelbewegungen werden in der Kurzschreibweise übereinander geschrieben.

Für unser Programmbeispiel sieht die Kurzschreibweise folgendermaßen aus.

A-      B-  
 A+    B+    C+    D+    D-    C-

**Aus der Kurzschreibweise lesen Sie folgende Informationen:**

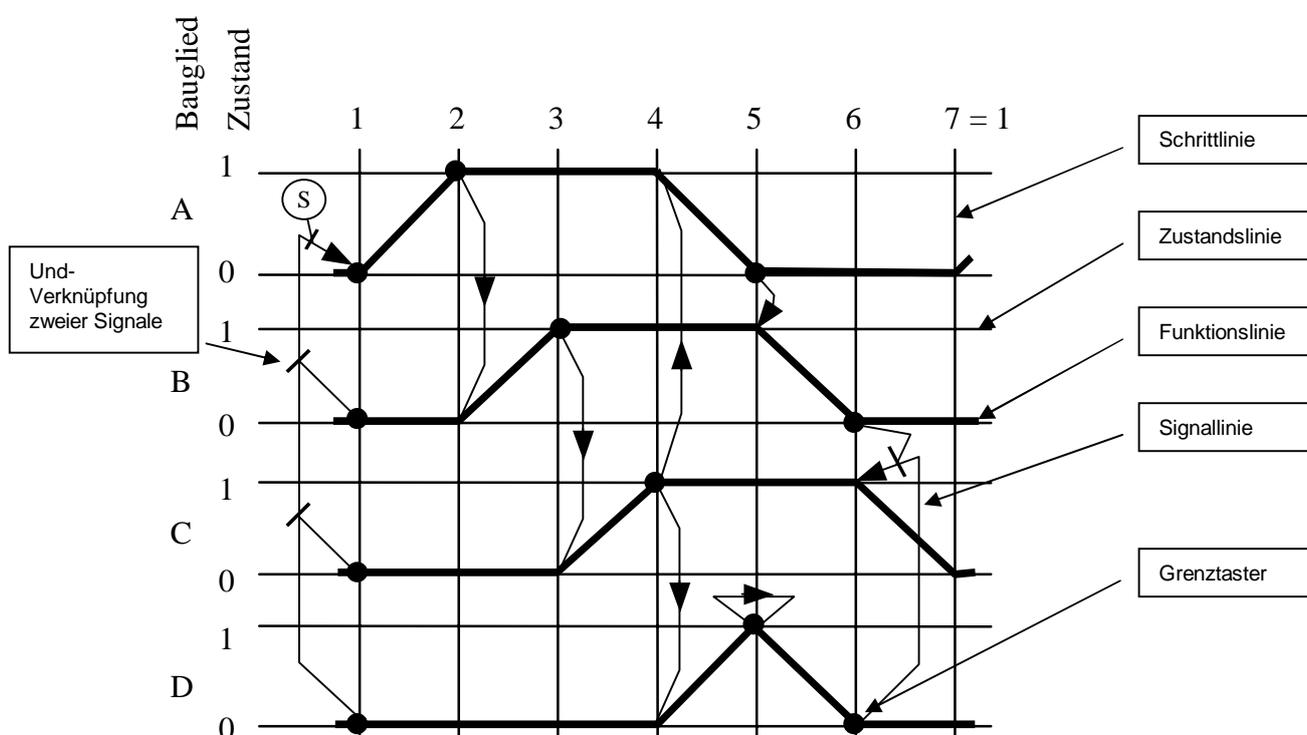
- Anzahl der Arbeitselemente und der notwendigen Schritte (4 Arbeitselemente und 6 Schritte)
- Parallel- und Mehrfachbewegungen eines Arbeitselements (Parallelbewegung in Schritt 4 und 5)
- Grundstellung der Anlage (A- B- C- D-)
- Signalzustände der Signalgeber (a0 b0 c0 d0 Grundstellung)(b0 c0 d0 A+)(a1 b0 c0 d0 1.Schritt)
- Signalfluss der Signalgeber (Start – A+, a1 – B+, b1 – C+, c1 – D+, a0 & d1 – D-, b0 & d0 – C-, c0)

## 4.5 Bewegungsdiagramme

### 4.5.1 Weg- Schritt- Diagramm

Hier wird der Arbeitsablauf eines Arbeitselements dargestellt und zwar wird in Abhängigkeit von den jeweiligen Schritten (Schritt: Änderung des Zustands irgendeiner Baueinheit) der zurückgelegte Weg aufgetragen. Sind für eine Steuerung mehrere Arbeitsglieder vorhanden, so werden diese in derselben Weise dargestellt und untereinander gezeichnet. Der Zusammenhang wird durch die Schritte hergestellt. Beim Weg-Schritt-Diagramm ist der Abstand der Schrittlinien immer gleich. Zusätzlich können in das Weg-Schritt-Diagramm die Signallinien mit eingetragen werden.

Für unser Programmbeispiel sieht das Weg-Schritt-Diagramm folgendermaßen aus.



### 4.5.2 Weg- Zeit- Diagramm

Das Weg-Zeit-Diagramm ist im Wesentlichen ein Weg-Schritt-Diagramm bei dem zusätzlich der zeitliche Verlauf der Bewegungen aufgezeigt wird durch eine Zeitleiste am unteren Ende des Diagramms kann die Dauer einer Bewegung abgelesen werden. Der Abstand der Schrittlinien verändert sich je nach benötigter Zeit. Die Anzahl der Schritte und die Art der Bewegungen bleiben unverändert.

## 4.6 GRAFCET nach DIN EN 60848

Der Grafcet ist eine prozessorientierte Darstellung einer Steuerungsaufgabe, unabhängig von deren Realisierung, z.B. der verwendeten Betriebsmittel. Er erleichtert das Zusammenwirken verschiedener Fachdisziplinen, z.B. Maschinenbau, Pneumatik, Hydraulik, Verfahrenstechnik, Elektrik, Elektronik usw. Eine Steuerungsaufgabe wird mit ihren wesentlichen Eigenschaften in einer Grobstruktur (Schrittfeld) und mit den für die jeweilige Anwendung erforderlichen Details in einer Feinstruktur (Befehlsfeld) übersichtlich dargestellt.

Der Ablauf der Abschervorrichtung kann nach einer Zuordnung der Signalglieder und der Arbeitselemente in einem GRAFCET nach DIN EN 60848 dargestellt werden.

Zuordnung der Signalglieder

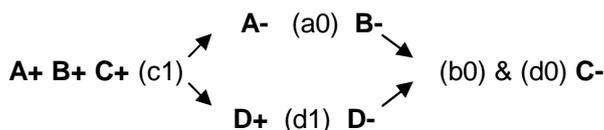
S1_START	Start-Taster
B1_a0	Sensor Zylinder A eingefahren
B2_a1	Sensor Zylinder A ausgefahren
B3_b0	Sensor Zylinder B eingefahren
B4_b1	Sensor Zylinder B ausgefahren
B5_c0	Sensor Zylinder C eingefahren
B6_c1	Sensor Zylinder C ausgefahren
B7_d0	Sensor Zylinder D eingefahren
B8_d1	Sensor Zylinder D ausgefahren

Zuordnung der Arbeitselemente

M1_A+/-	Magnetventil für Zylinder A aus/einfahren
M2_B+/-	Magnetventil für Zylinder B aus/einfahren
M3_C+/-	Magnetventil für Zylinder C aus/einfahren
M4_D+/-	Magnetventil für Zylinder D aus/einfahren

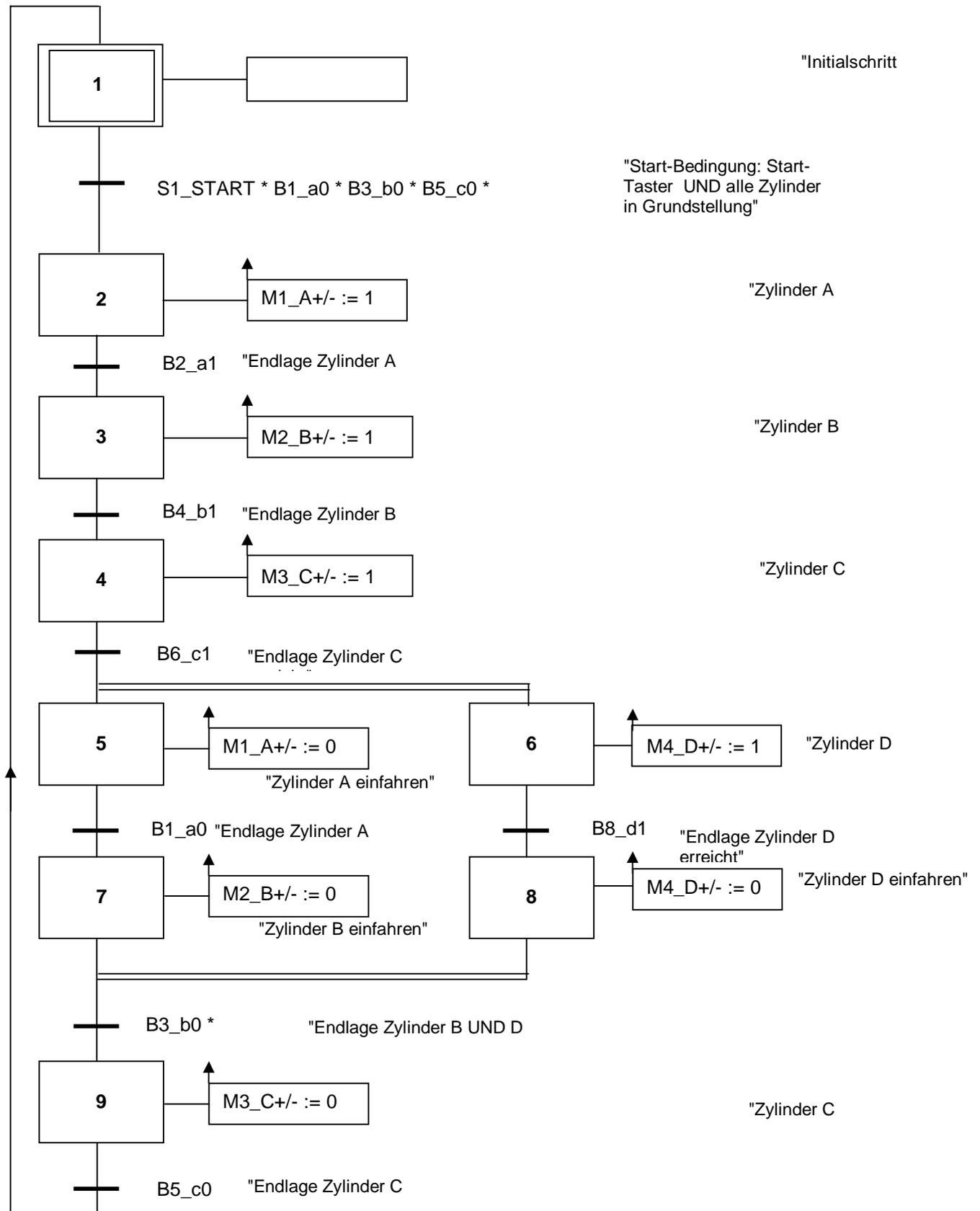
### Hinweis

Da es sich bei der Abschervorrichtung um zwei getrennte Arbeitsstationen (Zuführstation und Schervorrichtung) handelt, wird der Grafcet mit einer Simultanverzweigung erstellt.



Bei einer Simultanverzweigung teilt sich die Schrittkette auf und die Schritte laufen parallel.

## GRAFSET der Abschervorrichtung nach DIN EN 60848



## 5.0 Projektierung der Abschervorrichtung

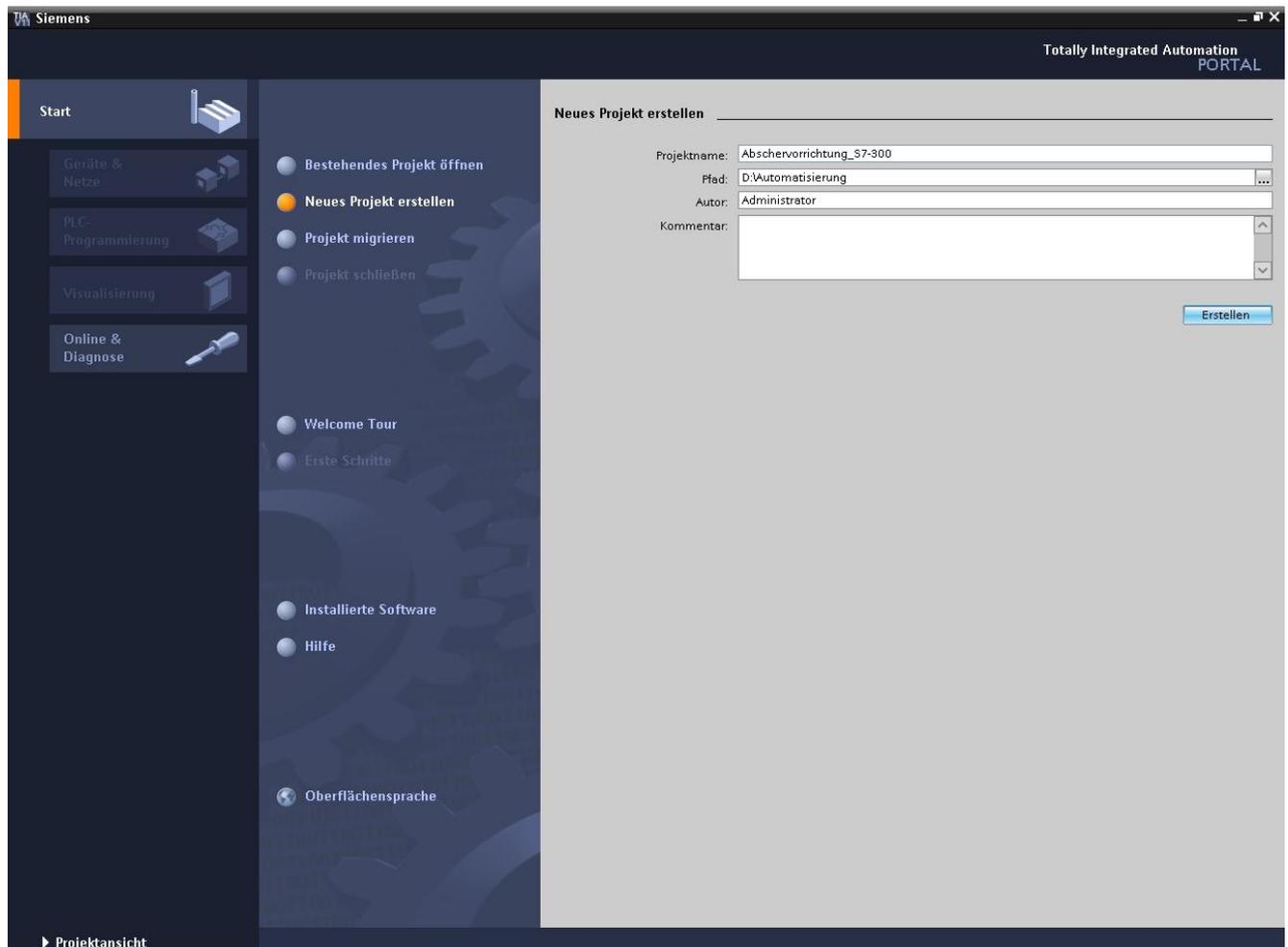
Aus dem GRAFCET soll nun ein S7-GRAPH Programm erstellt werden.

### 5.1 Neues TIA Projekt erstellen und Hardware anlegen

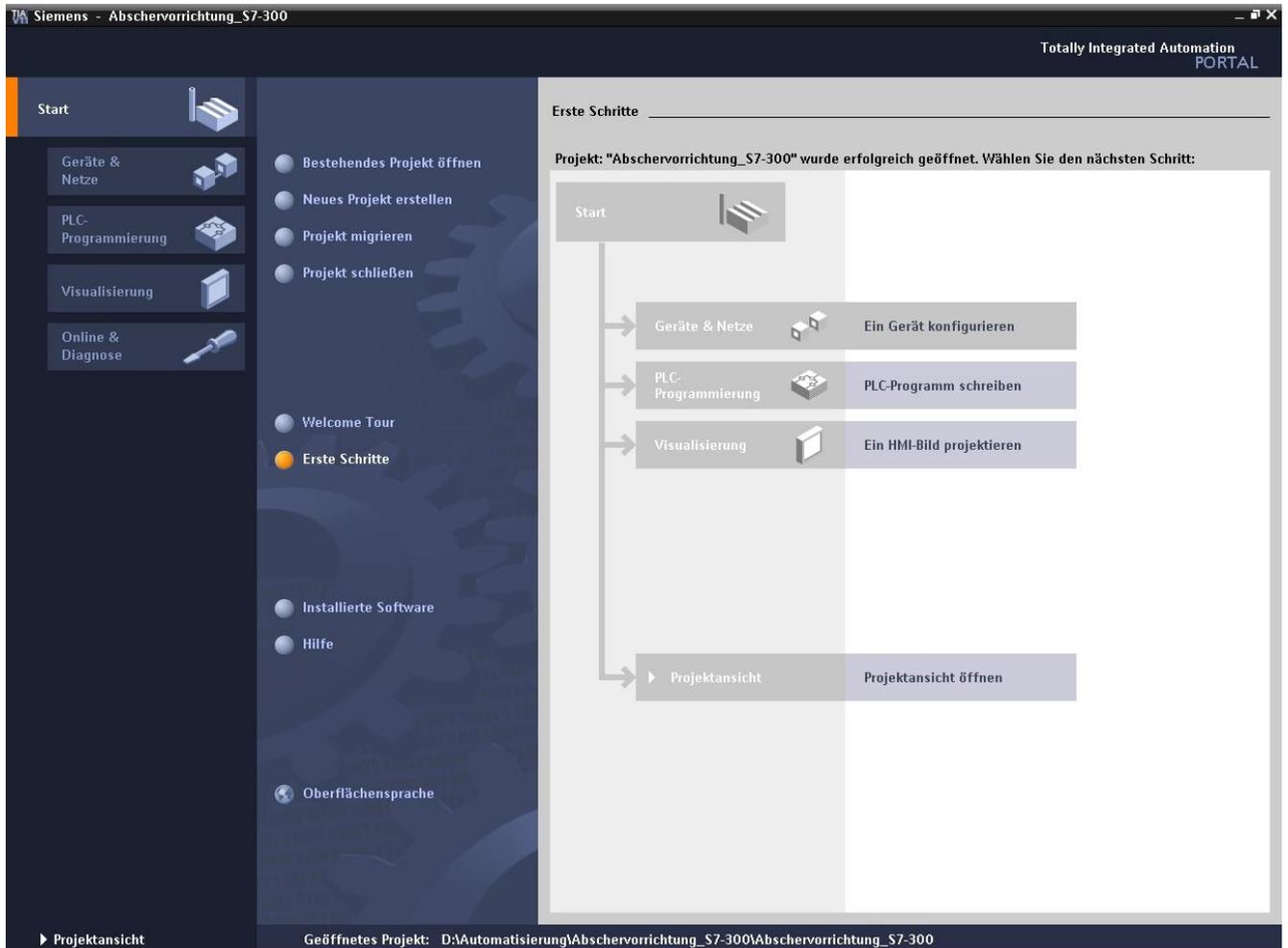
1. Das zentrale Werkzeug ist das **‚Totally Integrated Automation Portal‘**, das hier mit einem Doppelklick aufgerufen wird. ( → TIA-Portal V11)



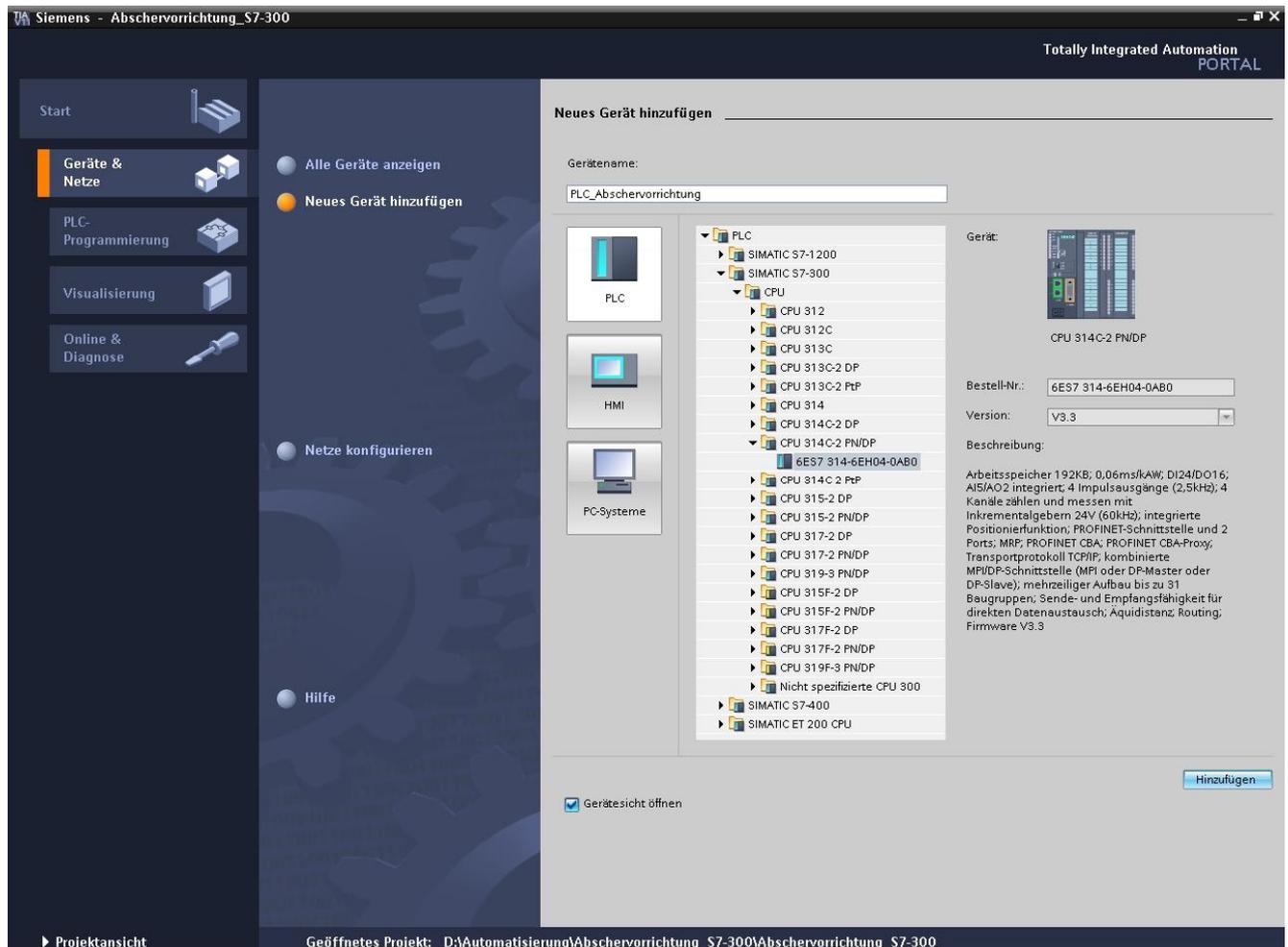
2. Programme für die SIMATIC S7-300 werden in Projekten verwaltet. Ein solches Projekt wird nun in der Portalansicht angelegt ( → Neues Projekt erstellen → Abschervorrichtung\_S7-300 → Erstellen)



3. Nun werden ‚**Erste Schritte**‘ zur Projektierung vorgeschlagen. Wir wollen zuerst ‚**ein Gerät konfigurieren**‘. ( → Erste Schritte → ein Gerät konfigurieren)



4. Danach klicken wir auf ‚**Neues Gerät hinzufügen**‘. Als Gerätenamen vergeben wir ‚**PLC\_Abschervorrichtung**‘ und wählen die ‚**CPU314C-2 PN/DP**‘ mit passender Bestell- und Versionsnummer aus.  
( → neues Gerät hinzufügen → PLC\_Abschervorrichtung → PLC → SIMATIC S7-300 → CPU → CPU314C-2 PN/DP → 6ES7 314-6EH04-0AB0 → V3.3 → Hinzufügen)



- Nun wechselt die Software automatisch zur Projektansicht mit der geöffneten Hardwarekonfiguration in der Gerätesicht.  
Hier können nun weitere Module aus dem Hardware-Katalog (rechts!) hinzugefügt werden.  
Wir wählen die interne DI24/DO16 Baugruppe mit 24 digitalen Eingängen und 16 digitalen Ausgängen und ändern die E/A- Anfangsadresse auf 0.  
( → Hardware-Katalog → DI24/DO16 → 0 → 0)

The screenshot shows the Siemens SIMATIC Manager interface. The main window displays the hardware configuration for a PLC system. The rack configuration shows a CPU 314C-2 PN/DP and a DI24/DO16 module. The 'Geräteübersicht' table lists the modules, and the 'E/A-Adressen' section shows the start address set to 0.

Baugruppe	Baugr...	Steck...	E-Adresse	A-Adresse	Typ	Bestell-Nr.
PLC_Abschervorrichtung	0	2			CPU 314C-2 PN/DP	6ES7 314-6EH04-0AB0
MPIDP-Schnittstelle_1	0	2 X1	2047*		MPIDP-Schnittstelle	
PROFINET-Schnittstelle_1	0	2 X2	2046*		PROFINET-Schnittst...	
DI24/DO16_1	0	2 5	0...2	0...1	DI24/DO16	
AIS/AO2_1	0	2 6	800...809	800...803	AIS/AO2	
Zählen_1	n	2 7	816...831	816...831	Zählen	

**E/A-Adressen**

**Eingangsadressen**

Anfangsadresse: 0  
Endadresse: 2  
Prozessabbild: OB1-PA  
Alarm-OB Nummer: 40

6. Damit die Software später auf die richtige CPU zugreift muss deren IP-Adresse und die Subnetzmaske eingestellt werden.  
 (→ Eigenschaften → Allgemein → PROFINET- Schnittstelle → Ethernet-Adressen → IP-Adresse im Projekt einstellen → IP-Adresse: 192.168.0.1 → Subnetzmaske: 255.255.255.0)

The screenshot shows the Siemens TIA Portal interface for configuring a SIMATIC 314C-2 PN/DP PLC. The main window displays the rack configuration with a 'PROFINET-Schnittstelle\_1' highlighted. Below it, a table lists the hardware components:

Baugruppe	Baugr...	Steck...	E-Adresse	A-Adresse	Typ	Bestell-Nr.
PLC_Abschervorrichtung	0	2			CPU 314C-2 PN/DP	6ES7 314-6EH04-0AB0
MP/DI/Schnittstelle_1	0	2 X1	2047*		MP/DI-Schnittstelle	
PROFINET-Schnittstelle_1	0	2 X2	2046*		PROFINET-Schnittst...	
DI24/DO16_1	0	2 5	0..2	0...1	DI24/DO16	
AIS/AO2_1	0	2 6	800..809	800..803	AIS/AO2	
Zähler_1	0	2 7	816..831	816..831	Zähler	

The 'Eigenschaften' (Properties) window for the selected 'PROFINET-Schnittstelle\_1 [PN-IO]' is open, showing the 'IP-Protokoll' (IP Protocol) settings:

- IP-Adresse im Projekt einstellen
- IP-Adresse: 192 . 168 . 0 . 1
- Subnetzmaske: 255 . 255 . 255 . 0
- Router verwenden
- Router-Adresse: 0 . 0 . 0 . 0

## 5.2 PLC-Variablen eingeben

Da bei moderner Programmierung nicht mit absoluten Adressen, sondern mit Variablen programmiert wird, müssen hier die **globalen PLC-Variablen** festgelegt werden.

Diese globalen PLC-Variablen sind beschreibende Namen mit Kommentar für jene Eingänge und Ausgänge, die im Programm Verwendung finden. Später kann bei der Programmierung über diesen Namen auf die globalen PLC-Variablen zugegriffen werden.

Diese globalen Variablen sind im gesamten Programm in allen Bausteinen verwendbar.

Wählen Sie hierzu in der Projektnavigation die ‚**PLC\_Abschervorrichtung [CPU314C-2 PN/DP]**‘ und dann ‚**PLC-Variablen**‘. Öffnen Sie die ‚**Standard-Variablen-tabelle**‘ mit einem Doppelklick und tragen dort wie unten gezeigt die Namen für die Ein- und Ausgänge ein. ( → PLC\_Abschervorrichtung [CPU314C-2 PN/DP] → PLC-Variablen → Standard-Variablen-tabelle)

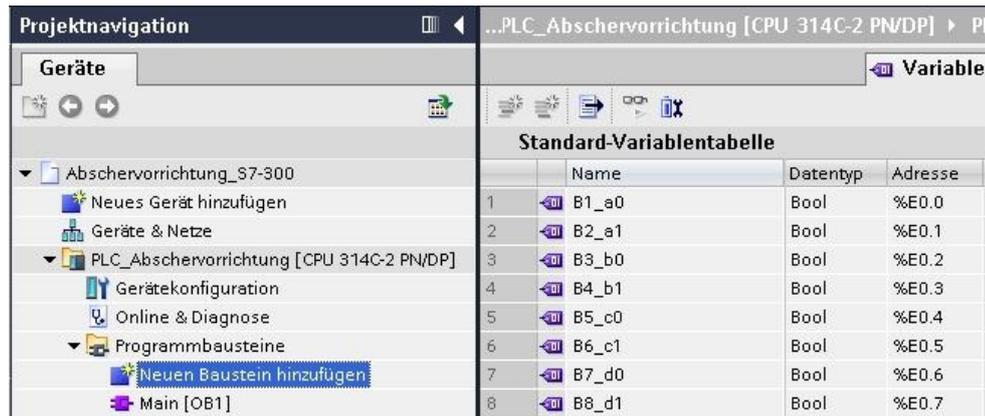
	Name	Datentyp	Adresse	Rema...	Sichtb..	Erreic...	Kommentar
1	B1_a0	Bool	%E0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor Zylinder A eingefahren NO
2	B2_a1	Bool	%E0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor Zylinder A ausgefahren NO
3	B3_b0	Bool	%E0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor Zylinder B eingefahren NO
4	B4_b1	Bool	%E0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor Zylinder B ausgefahren NO
5	B5_c0	Bool	%E0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor Zylinder C eingefahren NO
6	B6_c1	Bool	%E0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor Zylinder C ausgefahren NO
7	B7_d0	Bool	%E0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor Zylinder D eingefahren NO
8	B8_d1	Bool	%E0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor Zylinder D ausgefahren NO
9	S1_START	Bool	%E1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Start-Taster NO
10	M1_A+/-	Bool	%A0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Magnetventil Zylinder A aus_einfahren
11	M2_B+/-	Bool	%A0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Magnetventil Zylinder B aus_einfahren
12	M3_C+/-	Bool	%A0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Magnetventil Zylinder C aus_einfahren
13	M4_D+/-	Bool	%A0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Magnetventil Zylinder D aus_einfahren
14	Hinzufügen...				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Allgemein	
Variable	M4_D+/-
Allgemein	
Name:	M4_D+/-
Datentyp:	Bool
Adresse:	%A0.3
Kommentar:	Magnetventil Zylinder D aus_einfahren

## 5.2 GRAPH Schrittkettenbaustein hinzufügen

- Um die GRAPH-Schrittkette zu erstellen wählen Sie in der Projektnavigation die ,**PLC\_Abschervorrichtung [CPU314C-2 PN/DP]**' und dann ,**Programmbausteine**'. Dann führen Sie einen Doppelklick auf ,**Neuen Baustein hinzufügen**' aus.  
( → PLC\_Abschervorrichtung [CPU314C-2 PN/DP] → Programmbausteine → Neuen Baustein hinzufügen)



- Wählen Sie in der Auswahl ,**Funktionsbaustein (FB)**' und vergeben den Namen ,**Schrittkette**'. Als Programmiersprache wird ,**GRAPH**' vorgegeben. Die Nummerierung erfolgt automatisch. Da dieser FB1 später sowieso über den symbolischen Namen aufgerufen wird, spielt die Nummer keine so große Rolle mehr. Übernehmen Sie die Eingaben mit ,**OK**'.  
( → Funktionsbaustein (FB) → Schrittkette → GRAPH → OK)

Neuen Baustein hinzufügen
✕

Name:

  
 Organisations-  
baustein

  
 Funktions-  
baustein

  
 Funktion

  
 Daten-  
baustein

Sprache:

Nummer:

Manuell  
 Automatisch

Beschreibung:

Funktionsbausteine sind Codebausteine, die ihre Werte dauerhaft in Instanz-Datenbausteinen ablegen, sodass sie auch nach der Bausteinbearbeitung zur Verfügung stehen.

[mehr...](#)

► **Weitere Informationen**

Neu hinzufügen und öffnen

## 5.3 Lokale Variablen der GRAPH Schrittkette

Der Baustein ‚**Schrittkette [FB1]**‘ wird dann automatisch geöffnet. Bevor die Schrittkette graphisch erstellt wird, sollten in der Schnittstelle des Bausteins die zusätzlich benötigten Variablen deklariert werden. Bei der Deklaration der Schnittstelle werden die, nur in diesem Baustein bekannten, lokalen Variablen festgelegt.

### Die Variablen unterteilen sich in zwei Gruppen:

- Bausteinparameter, die die Schnittstelle des Bausteins für den Aufruf im Programm bilden.

Typ	Bezeichnung	Funktion	Verfügbar in
Eingangsparameter	Input	Parameter, deren Werte der Baustein liest.	Funktionen, Funktionsbausteinen und einigen Arten von Organisationsbausteinen
Ausgangsparameter	Output / Return	Parameter, deren Werte der Baustein schreibt.	Funktionen und Funktionsbausteinen
Durchgangsparameter	InOut	Parameter, deren Wert der Baustein beim Aufruf liest und nach der Bearbeitung wieder in denselben Parameter schreibt.	Funktionen und Funktionsbausteinen

- Lokaldaten, die zum Speichern von Zwischenergebnissen dienen.

Typ	Bezeichnung	Funktion	Verfügbar in
Temporäre Lokaldaten	Temp	Variablen, die zum Speichern von temporären Zwischenergebnissen dienen. Temporäre Daten bleiben nur für einen Zyklus erhalten.	Funktionen, Funktionsbausteinen und Organisationsbausteinen
Statische Lokaldaten	Static	Variablen, die zum Speichern von statischen Zwischenergebnissen im Instanz-Datenbaustein dienen. Statische Daten bleiben so lange erhalten, bis sie neu geschrieben werden, auch über mehrere Zyklen hinweg.	Funktionsbausteinen

Bei der Deklaration der lokalen Variablen werden in unserem Beispiel zusätzlich noch folgende Variablen benötigt.

**Input:**

start	Bool	Startbefehl
zyl_A_eingefahren	Bool	Rückmeldung Zylinder A ist eingefahren
zyl_A_ausgefahren	Bool	Rückmeldung Zylinder A ist ausgefahren
zyl_B_eingefahren	Bool	Rückmeldung Zylinder B ist eingefahren
zyl_B_ausgefahren	Bool	Rückmeldung Zylinder B ist ausgefahren
zyl_C_eingefahren	Bool	Rückmeldung Zylinder C ist eingefahren
zyl_C_ausgefahren	Bool	Rückmeldung Zylinder C ist ausgefahren
zyl_D_eingefahren	Bool	Rückmeldung Zylinder D ist eingefahren
zyl_D_ausgefahren	Bool	Rückmeldung Zylinder D ist ausgefahren

**Output:**

zyl_A_aus_einfahren	Bool	Zylinder A aus- bzw. einfahren
zyl_B_aus_einfahren	Bool	Zylinder B aus- bzw. einfahren
zyl_C_aus_einfahren	Bool	Zylinder C aus- bzw. einfahren
zyl_D_aus_einfahren	Bool	Zylinder D aus- bzw. einfahren

Dabei sind sämtliche Variablen vom Typ ‚Bool‘ , das heißt binäre Variablen die nur den Zustand ‚0‘ (false) oder ‚1‘ (true) haben können.

Sämtliche lokalen Variablen sollten zum besseren Verständnis auch mit einem ausreichenden Kommentar versehen werden.

1. Klicken Sie mit der rechten Maustaste in der Schnittstelle des Bausteins ,**Schritt**kette [FB1] auf die Zeile der letzten **Input** Variable und wählen Sie **Zeile hinzufügen**.  
Legen Sie die zusätzlichen **Input** Variablen an.

Schnittstelle						
	Name	Datentyp	Offset	Defaultwert	Sichtbar i...	Kommentar
9	SW_TOP	Bool	...	false	<input type="checkbox"/>	Halbautomatisch/Ignorie
10	SW_MAN	Bool	...	false	<input type="checkbox"/>	Handbetrieb
11	S_SEL	Int	...	0	<input type="checkbox"/>	Schritt auswählen, der a
12	S_ON	Bool	...	false	<input type="checkbox"/>	Schritt aktivieren, der an
13	S_OFF	Bool	...	false	<input type="checkbox"/>	Schritt deaktivieren, der
14	T_PUSH	Bool	...	false	<input type="checkbox"/>	Schaltfreigabe für Transi
15						
16				0	<input type="checkbox"/>	Schrittnummer

14	T_PUSH	Bool	...	false	<input type="checkbox"/>	Schaltfreigabe für Transition im halbautomat
15	start	Bool	...	false	<input type="checkbox"/>	Startbefehl
16	zyl_A_eingefahren	Bool	...	false	<input type="checkbox"/>	Rückmeldung Zylinder A ist eingefahren
17	zyl_A_ausgefahren	Bool	...	false	<input type="checkbox"/>	Rückmeldung Zylinder A ist ausgefahren
18	zyl_B_eingefahren	Bool	...	false	<input type="checkbox"/>	Rückmeldung Zylinder B ist eingefahren
19	zyl_B_ausgefahren	Bool	...	false	<input type="checkbox"/>	Rückmeldung Zylinder B ist ausgefahren
20	zyl_C_eingefahren	Bool	...	false	<input type="checkbox"/>	Rückmeldung Zylinder C ist eingefahren
21	zyl_C_ausgefahren	Bool	...	false	<input type="checkbox"/>	Rückmeldung Zylinder C ist ausgefahren
22	zyl_D_eingefahren	Bool	...	false	<input type="checkbox"/>	Rückmeldung Zylinder D ist eingefahren
23	zyl_D_ausgefahren	Bool	...	false	<input type="checkbox"/>	Rückmeldung Zylinder D ist ausgefahren

2. Klicken Sie nun auf die Zeile der letzten **Output** Variable und wählen Sie **Zeile hinzufügen**.  
Legen Sie die zusätzlichen **Output** Variablen an.

32	MAN_ON	Bool	...	false	<input type="checkbox"/>	Handbetrieb eingeschaltet
33	zyl_A_aus_einfahren	Bool	...	false	<input type="checkbox"/>	Zylinder A aus- bzw. einfahren
34	zyl_B_aus_einfahren	Bool	...	false	<input type="checkbox"/>	Zylinder B aus- bzw. einfahren
35	zyl_C_aus_einfahren	Bool	...	false	<input type="checkbox"/>	Zylinder C aus- bzw. einfahren
36	zyl_D_aus_einfahr...	Bool	...	false	<input type="checkbox"/>	Zylinder D aus- bzw. einfahren

## Liste aller Input und Output Variablen des FB1

Schnittstelle						
	Name	Datentyp	Offset	Defa...	Sichtbar i...	Kommentar
1	▼ Input					
2	OFF_SQ	Bool	...	false	<input checked="" type="checkbox"/>	Kette ausschalten
3	INIT_SQ	Bool	...	false	<input checked="" type="checkbox"/>	Kette in Initialzustand versetzen
4	ACK_EF	Bool	...	false	<input checked="" type="checkbox"/>	Alle Fehler und Störungen quittieren
5	S_PREV	Bool	...	false	<input checked="" type="checkbox"/>	Vorherigen Schritt am Parameter S_NO ausgeben
6	S_NEXT	Bool	...	false	<input checked="" type="checkbox"/>	Nächsten Schritt am Parameter S_NO anzeigen
7	SW_AUTO	Bool	...	false	<input checked="" type="checkbox"/>	Automatischer Betrieb
8	SW_TAP	Bool	...	false	<input checked="" type="checkbox"/>	Halbautomatisch/Weiterschalten mit Transition
9	SW_TOP	Bool	...	false	<input checked="" type="checkbox"/>	Halbautomatisch/Ignoriere Transition
10	SW_MAN	Bool	...	false	<input checked="" type="checkbox"/>	Handbetrieb
11	S_SEL	Int	...	0	<input checked="" type="checkbox"/>	Schritt auswählen, der an S_NO ausgegeben werden soll
12	S_ON	Bool	...	false	<input checked="" type="checkbox"/>	Schritt aktivieren, der an Parameter S_NO anliegt
13	S_OFF	Bool	...	false	<input checked="" type="checkbox"/>	Schritt deaktivieren, der an Parameter S_NO anliegt
14	T_PUSH	Bool	...	false	<input checked="" type="checkbox"/>	Schaltfreigabe für Transition im halbautomatischen Betrieb
15	start	Bool	...	false	<input checked="" type="checkbox"/>	Startbefehl
16	zyl_A_eingefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder A ist eingefahren
17	zyl_A_ausgefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder A ist ausgefahren
18	zyl_B_eingefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder B ist eingefahren
19	zyl_B_ausgefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder B ist ausgefahren
20	zyl_C_eingefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder C ist eingefahren
21	zyl_C_ausgefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder C ist ausgefahren
22	zyl_D_eingefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder D ist eingefahren
23	zyl_D_ausgefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder D ist ausgefahren
24	▼ Output					
25	S_NO	Int	...	0	<input checked="" type="checkbox"/>	Schrittnummer
26	S_MORE	Bool	...	false	<input checked="" type="checkbox"/>	Weitere Schritte sind vorhanden und können in S_NO angezeigt werden.
27	S_ACTIVE	Bool	...	false	<input checked="" type="checkbox"/>	Schritt, der an Parameter S_NO angezeigt wird, ist aktiv
28	ERR_FLT	Bool	...	false	<input checked="" type="checkbox"/>	Sammelfehler Interlock oder Supervision
29	AUTO_ON	Bool	...	false	<input checked="" type="checkbox"/>	Automatischer Betrieb eingeschaltet
30	TAP_ON	Bool	...	false	<input checked="" type="checkbox"/>	Halbautomatischer Betrieb/Schritt mit Transition eingeschaltet
31	TOP_ON	Bool	...	false	<input checked="" type="checkbox"/>	Halbautomatischer Betrieb/Ignoriere Transition eingeschaltet
32	MAN_ON	Bool	...	false	<input checked="" type="checkbox"/>	Handbetrieb eingeschaltet
33	zyl_A_aus_einfahren	Bool	...	false	<input checked="" type="checkbox"/>	Zylinder A aus- bzw. einfahren
34	zyl_B_aus_einfahren	Bool	...	false	<input checked="" type="checkbox"/>	Zylinder B aus- bzw. einfahren
35	zyl_C_aus_einfahren	Bool	...	false	<input checked="" type="checkbox"/>	Zylinder C aus- bzw. einfahren
36	zyl_D_aus_einfahr...	Bool	...	false	<input checked="" type="checkbox"/>	Zylinder D aus- bzw. einfahren
37	▼ InOut					

### Hinweis:

Um eine Verwechslung mit den PLC-Variablen und den GRAPH Baustein-Variablen zu vermeiden, sollten die selbstdefinierten lokalen Variablen klein geschrieben werden.

## 5.4 Aufbau der Schrittkette

Nachdem die lokalen Variablen deklariert wurden kann nun mit der Erstellung der Schrittkette begonnen werden.

**Schnittstelle**

Name	Datentyp	Offset	Defa...	Sichtbar i...	Kommentar
13 S_OFF	Bool	...	false	<input checked="" type="checkbox"/>	Schritt deaktivieren, der an Parameter S_NO anlieg
14 T_PUSH	Bool	...	false	<input checked="" type="checkbox"/>	Schaltfreigabe für Transition im halbautomatischer
15 start	Bool	...	false	<input checked="" type="checkbox"/>	Startbefehl
16 zyl_A_eingefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder A ist eingefahren

**Navigation**

Vorausgeschaltete perma...

**Ketten (1)**

1: <neue Kette>

Schrittfeld

Transition bzw. Weiterschaltbedingung

Der erste Schritt der Ablaufkette wird in den Baustein automatisch eingefügt. Dieser Schritt wird als Initialschritt gekennzeichnet und ist beim Start der Ablaufkette aktiv.

## 5.4.1 Das Prinzip der Ablaufkette

Eine Ablaufkette besteht aus einer Folge von Schritten, die abhängig von den Bedingungen zum Weiterschalten in einer festgelegten Reihenfolge aktiviert werden.

Die Bearbeitung einer Ablaufkette beginnt immer mit einem Initialschritt oder mit mehreren Initialschritten die an beliebiger Stelle in der Ablaufkette stehen. Solange die Aktionen eines Schrittes ausgeführt werden, ist dieser Schritt aktiv. Bei Ausführung von mehreren Schritten gleichzeitig sind alle diese Schritte aktiv.

Ein Schritt wird verlassen, wenn alle eventuell anstehenden Störungen behoben bzw. bestätigt sind und die dem Schritt folgende Transition erfüllt ist.

Der nächste Schritt, der der erfüllten Transition folgt, wird aktiv.

Am Ende einer Ablaufkette steht ein Sprung zu einem beliebigen Schritt dieser Ablaufkette oder einer anderen Ablaufkette des FB. Dadurch ist ein zyklischer Betrieb der Ablaufkette möglich. Am Ende der Ablaufkette kann auch ein Kettenende stehen. Der Ablauf endet mit Erreichen des Kettenendes.

### Aktiver Schritt

Ein aktiver Schritt ist ein Schritt, dessen Aktionen gerade bearbeitet werden. Der Schritt wird aktiv

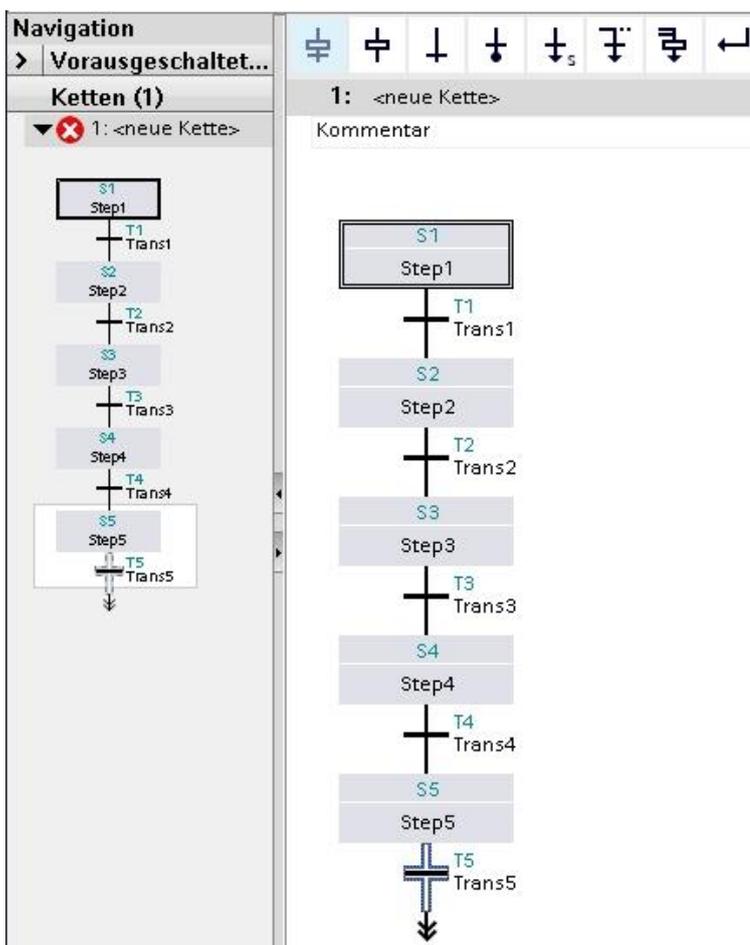
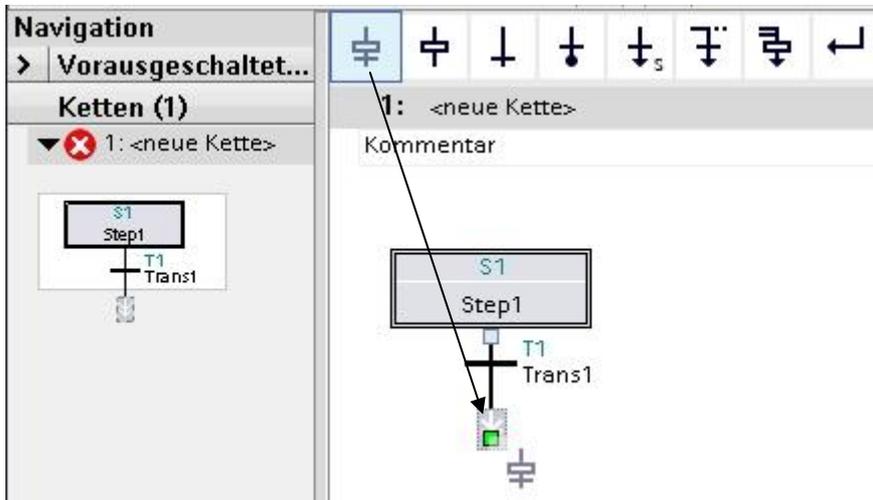
wenn die Bedingungen der vorhergehenden Transition erfüllt sind oder wenn er als Initialschritt definiert ist und die Ablaufkette initialisiert wurde oder wenn er durch eine ereignisabhängige Aktion aufgerufen wird.

### Elemente einer Ablaufkette

Favoriten	
Einfache Anweisungen	
Name	Beschreibung
<ul style="list-style-type: none"> <li>Graph Struktur           <ul style="list-style-type: none"> <li> Schritt und Transition Schritt und Transition [Shift+F5]</li> <li> Schritt Schritt</li> <li> Transition Transition</li> <li> Kettenende Kettenende [Shift+F7]</li> <li> Sprung zu Schritt Sprung [Shift+F12]</li> <li> Alternativzweig Alternativzweig öffnen [Shift+F9]</li> <li> Simultanzweig Simultanzweig öffnen [Shift+F9]</li> <li> Verzweigung schließen Verzweigung schließen [Shift+F11]</li> </ul> </li> </ul>	

## 5.5 Grafische Darstellung der Abschervorrichtung

1. Zum Erstellen unserer Schrittkette ziehen wir zunächst vier weitere Schritte mit Transitionen ans Ende der Kette.



## Die nächsten Schritte werden mit einer Verzweigung eingefügt

Es gibt zwei Arten von Verzweigungen



### Die Alternativ-Verzweigung,

sie wird nach dem angewählten Schritt eingefügt und beginnt mit einer Transition  
Die Schritte einer Alternativ-Verzweigung werden nur bearbeitet wenn die Transition erfüllt ist.  
Die Verzweigung kann, entweder nach links zu einer Transition geschlossen oder mit Kettenende beendet werden.



Verzweigung schließen.



Kettenende einfügen.



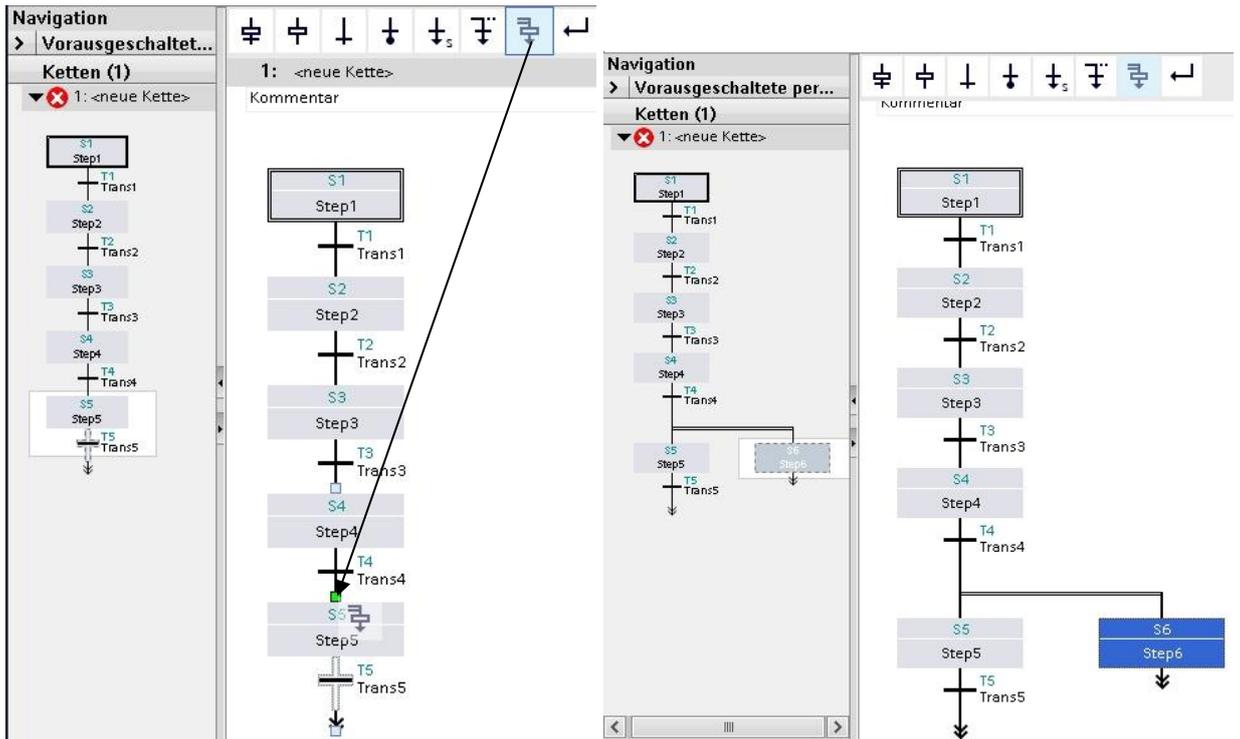
### Die Simultan-Verzweigung,

sie wird nach der angewählten Transition eingefügt und beginnt mit einem Schritt.  
Die Schritte einer Simultan-Verzweigung müssen bearbeitet werden, da sie parallel zu den Grundschrinen durchlaufen werden.  
Die Verzweigung muss nach links zu einem Schritt geschlossen werden.



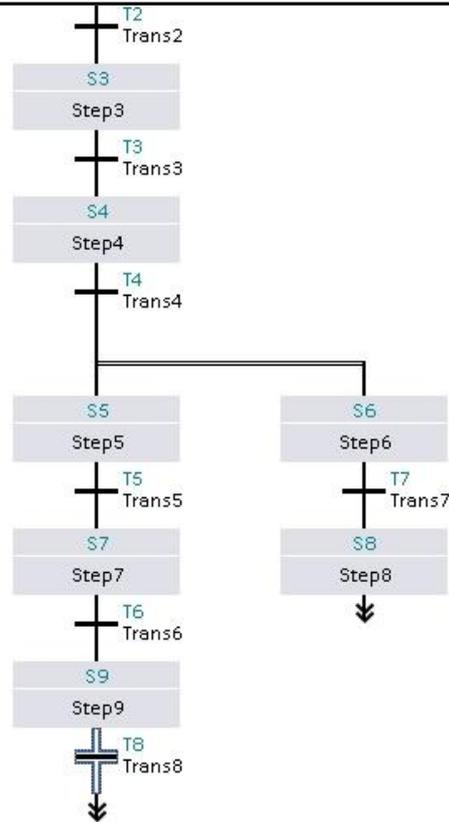
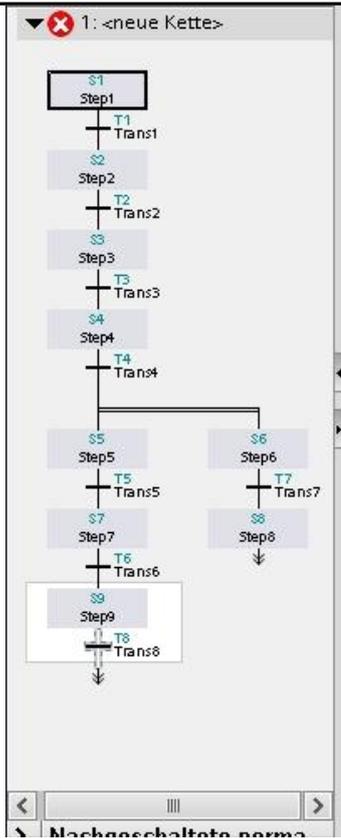
Verzweigung schließen.

2. Danach ziehen wir **Simultanzweig öffnen** ans Ende der Transition 4.

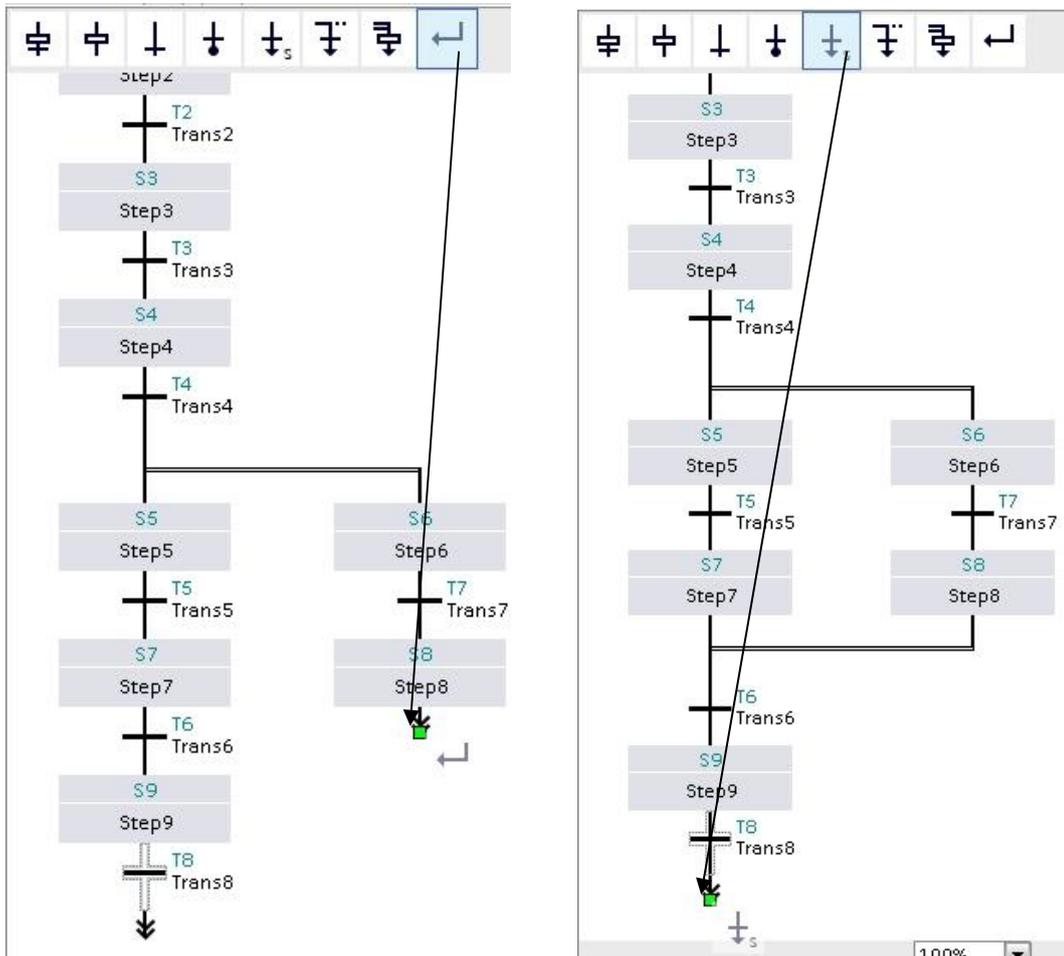


Schritt 6 wird zusammen mit der Simultan-Verzweigung eingefügt.

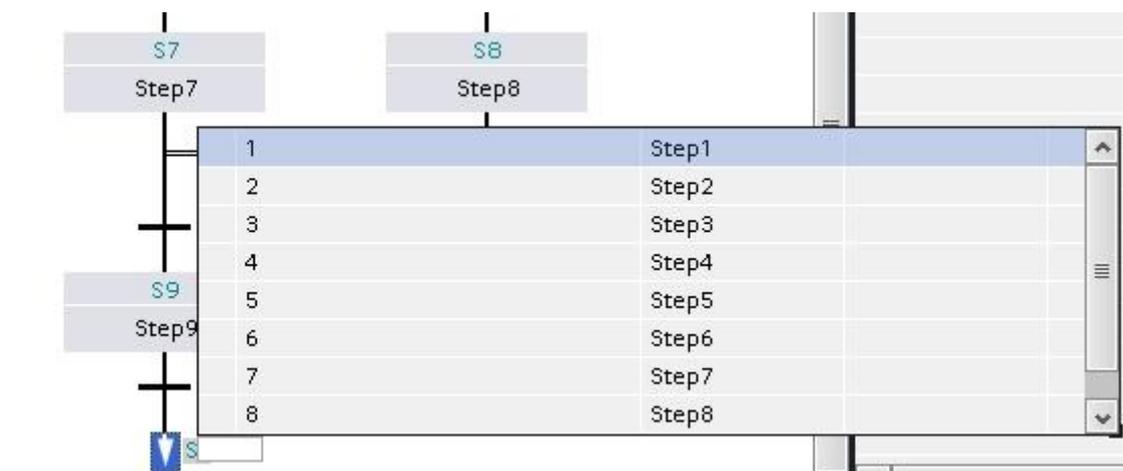
3. Jetzt werden noch drei weitere Schritte mit Transitionen angehängt.



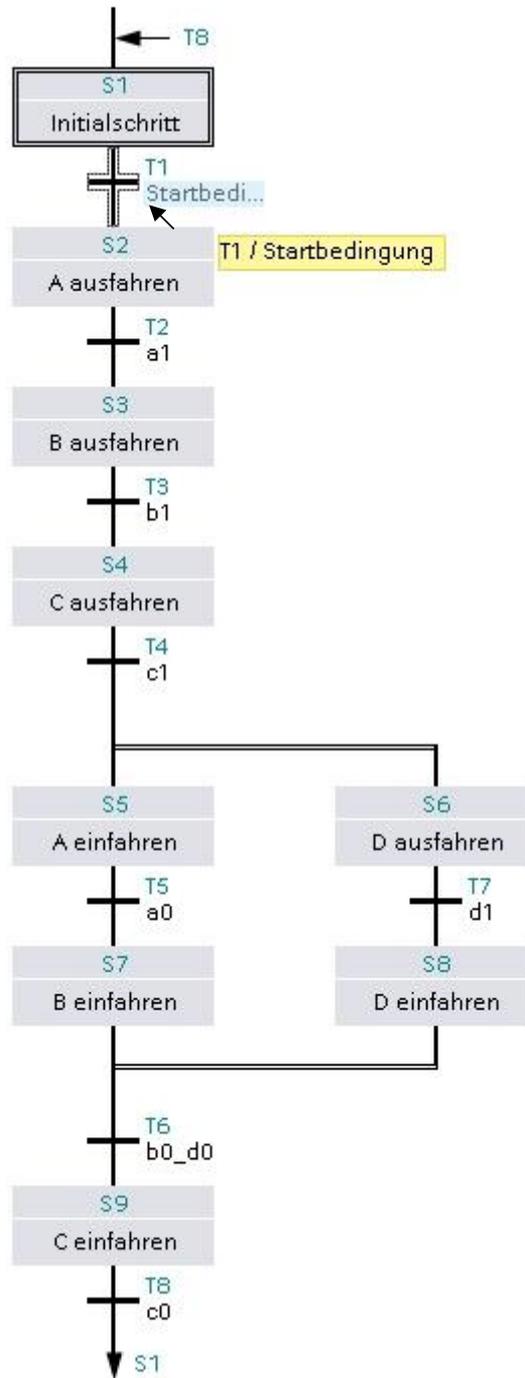
- Nun muss nur noch die Simultan-Verzweigung geschlossen und ein Sprung zum ersten Schritt eingefügt werden.



- Schritt 1 anwählen.



6. Die Schrittkette wird jetzt noch bezeichnet



## 5.6 Aktionen und Transitionen zuweisen

In der Einzelschrittansicht werden die Aktionen und die Transitionen eines Schrittes zugewiesen.

- Um in die Einzelschrittansicht zu gelangen, müssen wir entweder den Initialschritt S1 doppelklicken oder den Schritt markieren und die **Schaltfläche für Einzelschrittansicht** anklicken.

Schaltfläche für Kettenansicht

The screenshot displays the 'Bausteinschnittstelle' (Component Interface) for step S1. The left pane shows a ladder logic diagram with steps S1 to S9 and transitions T1 to T8. The right pane shows the configuration for S1, including interlocks, supervision, and actions. A table is visible for assigning actions to interlocks and events.

Interlock	Ereignis	Kennung	Aktion
		<hinzufügen>	

## Schritt S1 „Initialschritt“ und Transition T1 „Startbedingung“

Im Initialschritt S1 werden zunächst keine Aktionen zugewiesen.

Als Weberschaltbedingung zum nächsten Schritt programmieren wir die Startbedingung.

„Wenn alle Arbeitselemente in eingefahrener Position sind und der Startbefehl gegeben wird“.

2. Zuerst einen UND Baustein an den Eingang der Transition ziehen und drei weitere Eingänge anlegen. Danach die Eingänge mit den Variablen beschalten.

**Schnittstelle**

Name	Datentyp	Offset	Default..	Sichtbar i...	Kommentar
15 start	Bool	...	false	<input checked="" type="checkbox"/>	Startbefehl
16 zyl_A_eingefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder A ist eingefahren
17 zyl_A_ausgefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder A ist ausgefahren
18 zyl_B_eingefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder B ist eingefahren
19 zyl_B_ausgefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder B ist ausgefahren
20 zyl_C_eingefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder C ist eingefahren
21 zyl_C_ausgefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder C ist ausgefahren
22 zyl_D_eingefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder D ist eingefahren
23 zyl_D_ausgefahren	Bool	...	false	<input checked="" type="checkbox"/>	Rückmeldung Zylinder D ist ausgefahren

**Navigation**

Vorausgeschaltete perm...

**Ketten (1)**

1: <neue Kette>

S1 Initialschritt

T1 Startbedingung

S2 Ausfahren

T2 a1

S3 Bausfahren

T3 b1

S4 Causfahren

T4 ct

S5 Aeinfahren

T5 a0

S6 Dausfahren

T7 d1

S7 Beinfahren

S8 Deinfahren

T6 b0\_d0

S9 Ceinfahren

Nachgeschaltete perma...

**S1: Initialschritt**

Kommentar

Interlock -(c)-: .....

Supervision -(v)-: .....

Aktionen: .....

**T1 - Startbedingung: .....**

#start

#zyl\_A\_eingefahren

#zyl\_B\_eingefahren

#zyl\_C\_eingefahren

#zyl\_D\_eingefahren

## Schritt S2 „A ausfahren“ und Transition T2 „a1“

Im zweiten Schritt soll der Zylinder A ausfahren und auch über die nachfolgenden Schritte ausgefahren bleiben, deswegen muss hier der Operand (Output-Variable) speichernd auf 1 gesetzt werden. Der Operand bleibt auch nach verlassen oder deaktivieren des Schrittes, so lange auf 1 gesetzt, bis er wieder auf 0 zurückgesetzt wird.

**Navigation**  
 > Vorausgeschaltete perm...  
**Ketten (1)**  
 1: <neue Kette>

**S2: A ausfahren**  
 Kommentar  
 Interlock -(c)-: .....  
 Supervision -(v)-: .....  
 Aktionen: .....

Interlock	Ereignis	Kennung	Aktion
		- Keine Operation	
		CD - Rückwärts zählen	
		CR - Zähler rücksetzen	
		CS - Zähleranfangswert setzen	
		CU - Vorwärts zählen	
		D - Einschaltverzögerung	
		L - Setzen für limitierte Zeitdauer	
		N - Setzen solange Schritt aktiv	
		ON - Schritt aktivieren	
		OFF - Schritt deaktivieren	
		R - Auf 0 setzen	
		S - Auf 1 setzen	

T2 - a1: .....

3. Die Aktion zu Schritt S2 und die Weberschaltbedingung zu Transition T2 eingeben.

**Navigation**  
 > Vorausgeschaltete perm...  
**Ketten (1)**  
 1: <neue Kette>

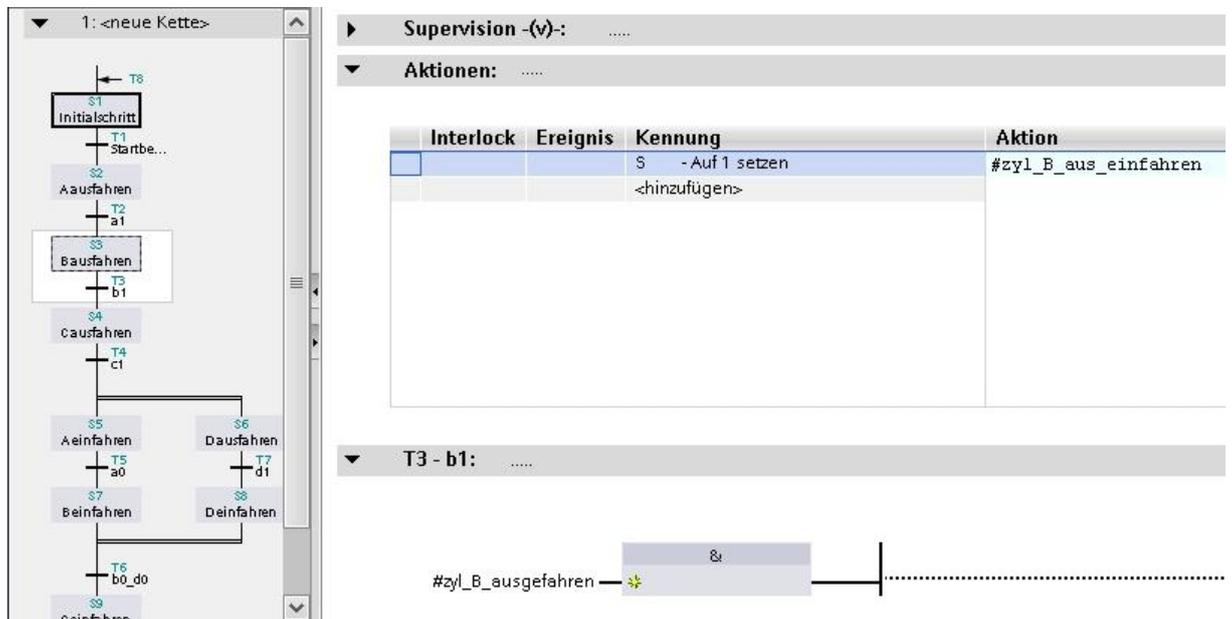
**T2 - a1: .....**

Interlock	Ereignis	Kennung	Aktion
		S - Auf 1 setzen	#zyl_A_aus_einfahren
		<hinzufügen>	

#zyl\_A\_ausgefahren — S — | .....  
 &

## Schritt S3 „B ausfahren“ und Transition T3 „b1“

- Die Aktion zu Schritt S3 und die Weberschaltbedingung zu Transition T3 eingeben.



**Supervision -(v)-:** .....

**Aktionen:** .....

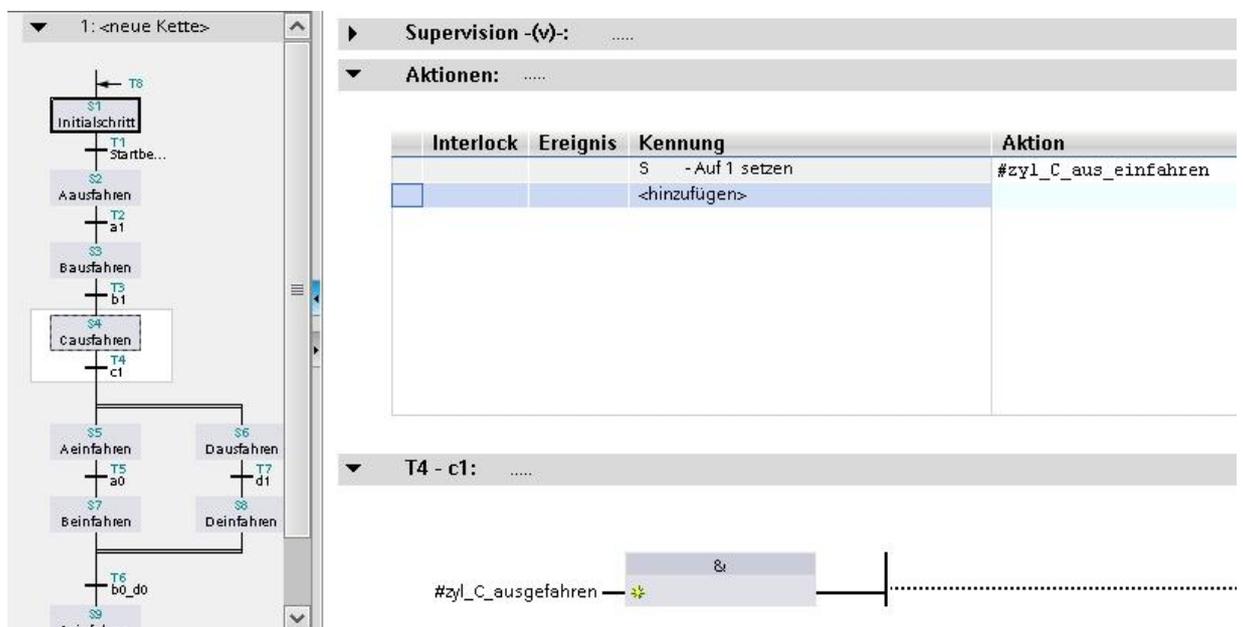
Interlock	Ereignis	Kennung	Aktion
		S - Auf 1 setzen	#zyl_B_aus_einfahren
		<hinzufügen>	

**T3 - b1:** .....

#zyl\_B\_ausgefahren → &

## Schritt S4 „C ausfahren“ und Transition T4 „c1“

- Die Aktion zu Schritt S4 und die Weberschaltbedingung zu Transition T4 eingeben.



**Supervision -(v)-:** .....

**Aktionen:** .....

Interlock	Ereignis	Kennung	Aktion
		S - Auf 1 setzen	#zyl_C_aus_einfahren
		<hinzufügen>	

**T4 - c1:** .....

#zyl\_C\_ausgefahren → &

## Schritt S5 „A einfahren“ und Transition T5 „a0“

Im fünften Schritt soll der Zylinder A wieder einfahren und auch über die nachfolgenden Schritte eingefahren bleiben, deswegen muss hier der Operand speichernd auf 0 rückgesetzt werden.

6. Die Aktion zu Schritt S5 und die Weberschaltbedingung zu Transition T5 eingeben.

The screenshot shows the SIMATIC Manager interface. On the left, a ladder logic diagram highlights step S5 'A einfahren' and transition T5 'a0'. On the right, the configuration window for transition T5 - a0 is open. It shows the 'Aktionen:' section with a table for defining actions and interlocks.

Interlock	Ereignis	Kennung	Aktion
		R - Auf 0 setzen	#zyl_A_aus_einfahren
		<hinzufügen>	

Below the table, the configuration for the transition T5 - a0 is shown, including the interlock condition: #zyl\_A\_eingefahren &.

## Schritt S6 „D ausfahren“ und Transition T7 „d1“

7. Die Aktion zu Schritt S6 und die Weberschaltbedingung zu Transition T7 eingeben.

The screenshot shows the SIMATIC Manager interface. On the left, a ladder logic diagram highlights step S6 'D ausfahren' and transition T7 'd1'. On the right, the configuration window for transition T7 - d1 is open. It shows the 'Aktionen:' section with a table for defining actions and interlocks.

Interlock	Ereignis	Kennung	Aktion
		S - Auf 1 setzen	#zyl_D_aus_einfahren
		<hinzufügen>	

Below the table, the configuration for the transition T7 - d1 is shown, including the interlock condition: #zyl\_D\_ausgefahren &.

Alternativ könnte hier auch die Aktion mit „N – setzen solange Schritt aktiv“ programmiert werden, da im nächsten Schritt der Zylinder D wieder eingefahren wird. Die Aktion „R – Auf 0 setzen“ im nächsten Schritt entfällt dann.

Da in unserem Beispiel beim Deaktivieren aller Schritte die Arbeitselemente in Position bleiben sollen, wird hier mit der Speicherfunktion programmiert.

## Schritt S7 „B einfahren“ und Transition T6 „b0\_d0“

8. Die Aktion zu Schritt S7 und die Weberschaltbedingung zu Transition T6 eingeben.

Interlock	Ereignis	Kennung	Aktion
		R - Auf 0 setzen	#zyl_B_aus_einfahren
		<hinzufügen>	

**T6 - b0\_d0:**

#zyl\_B\_eingefahren — & — |

#zyl\_D\_eingefahren — \* — |

## Schritt S8 „D einfahren“ und Transition T6 „b0\_d0“

9. Die Aktion zu Schritt S8 und die Weberschaltbedingung zu Transition T6 eingeben.

Interlock	Ereignis	Kennung	Aktion
		R - Auf 0 setzen	#zyl_D_aus_einfahren
		<hinzufügen>	

**T6 - b0\_d0:**

#zyl\_B\_eingefahren — & — |

#zyl\_D\_eingefahren — \* — |

Da nach den Schritten S7 und S8 der Simultanweig wieder geschlossen wird, haben beide Schritte eine gemeinsame Weberschaltbedingung (Transition T6 „b0\_d0“). Es wird auf den nächsten Schritt weberschaltet wenn beide Schritte S7 und S8 aktiv sind und die Bedingung der Transition T6 erfüllt ist.

**Ein Simultanweig kann nur verlassen werden, wenn die letzten Schritte aller Zweige aktiv sind und die gemeinsame Weberschaltbedingung erfüllt ist.**

## Schritt S9 „C einfahren“ und Transition T8 „c0“

10. Die Aktion zu Schritt S9 und die Weberschaltbedingung zu Transition T8 eingeben.

The screenshot shows the step ladder logic on the left and the configuration panels on the right. The logic includes steps S1 to S9 and transitions T1 to T8. Step S9 is highlighted, and transition T8 is selected.

**Supervision -(v)-: .....**

**Aktionen: .....**

Interlock	Ereignis	Kennung	Aktion
		R - Auf 0 setzen	#zyl_C_aus_einfahren
		<hinzufügen>	

**T8 - c0: .....**

#zyl\_C\_eingefahren &

11. Kettenansicht wählen

The screenshot shows the 'Kettenansicht' (Chain View) of the step ladder logic. The 'Schnittstelle' (Interface) is set to 'Kettenansicht'. The 'Navigation' pane shows 'Ketten (1)' with a new chain selected. The main diagram shows the step ladder logic with step S9 and transition T8 highlighted.

## Schrittfolge [FB1] übersetzen und speichern

Damit ist unser Schrittfolgenbaustein vorerst fertig und sollte jetzt auch übersetzt und gespeichert werden.

12. Schrittfolge [FB1] markieren und Baustein übersetzen anklicken. Danach Projekt speichern.

The screenshot shows the Siemens SIMATIC Manager interface. The 'Projektnavigation' pane on the left shows the project structure, with 'Schrittfolge [FB1]' selected. The main workspace displays a ladder logic diagram for a step sequence (FB1) with steps S1 to S7 and transitions T1 to T7. The 'Info' window at the bottom shows the translation results:

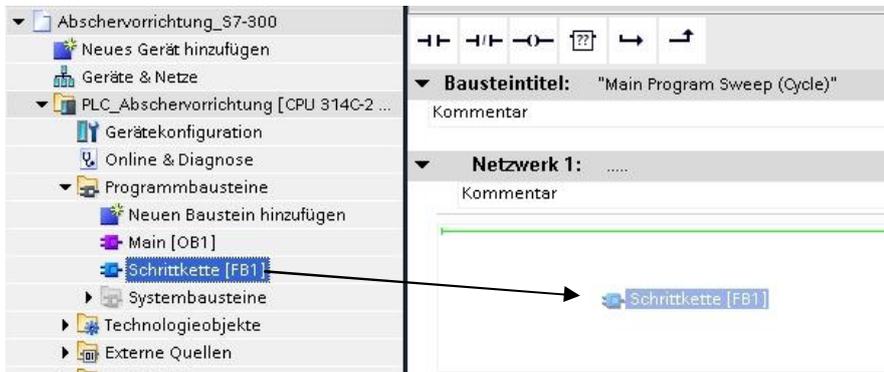
! Pfad	Beschreibung	Fehler	Warnungen	Zeit
! PLC_Abschervorrichtung		0	1	12:48:24
! Programmbausteine		0	1	12:48:24
! Schrittfolge (FB1)		0	1	12:48:24
! Kette 1	Schritt Initialschritt enthält keine Aktionen.	?	1	12:48:25
! Baustein wurde erfolgreich übersetzt.		0	0	12:48:41
! Übersetzen beendet (Fehler: 0; Warnungen: 1)		0	1	12:48:41

Im Info Fenster wird das Übersetzungsergebnis angezeigt.

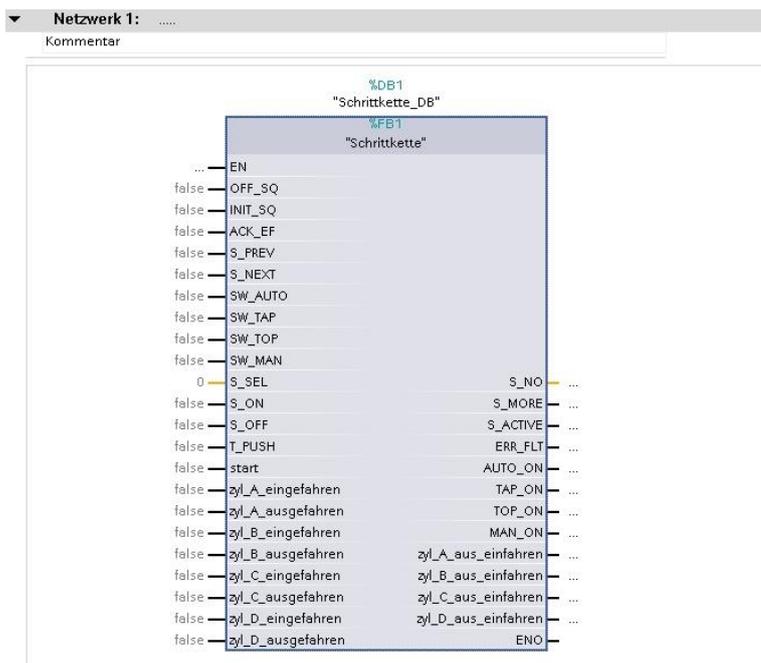
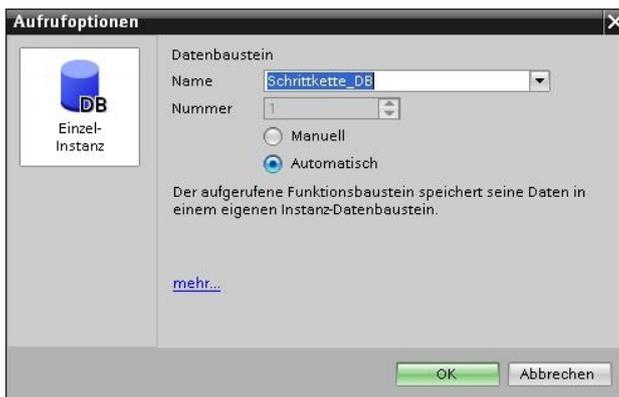
## 5.7 Schrittkette [FB1] im Main [OB1] aufrufen und beschalten

Die Schrittkette [FB1] wird nun im Main [OB1] Baustein aufgerufen, mit einem Instanz-Datenbaustein verknüpft und beschaltet.

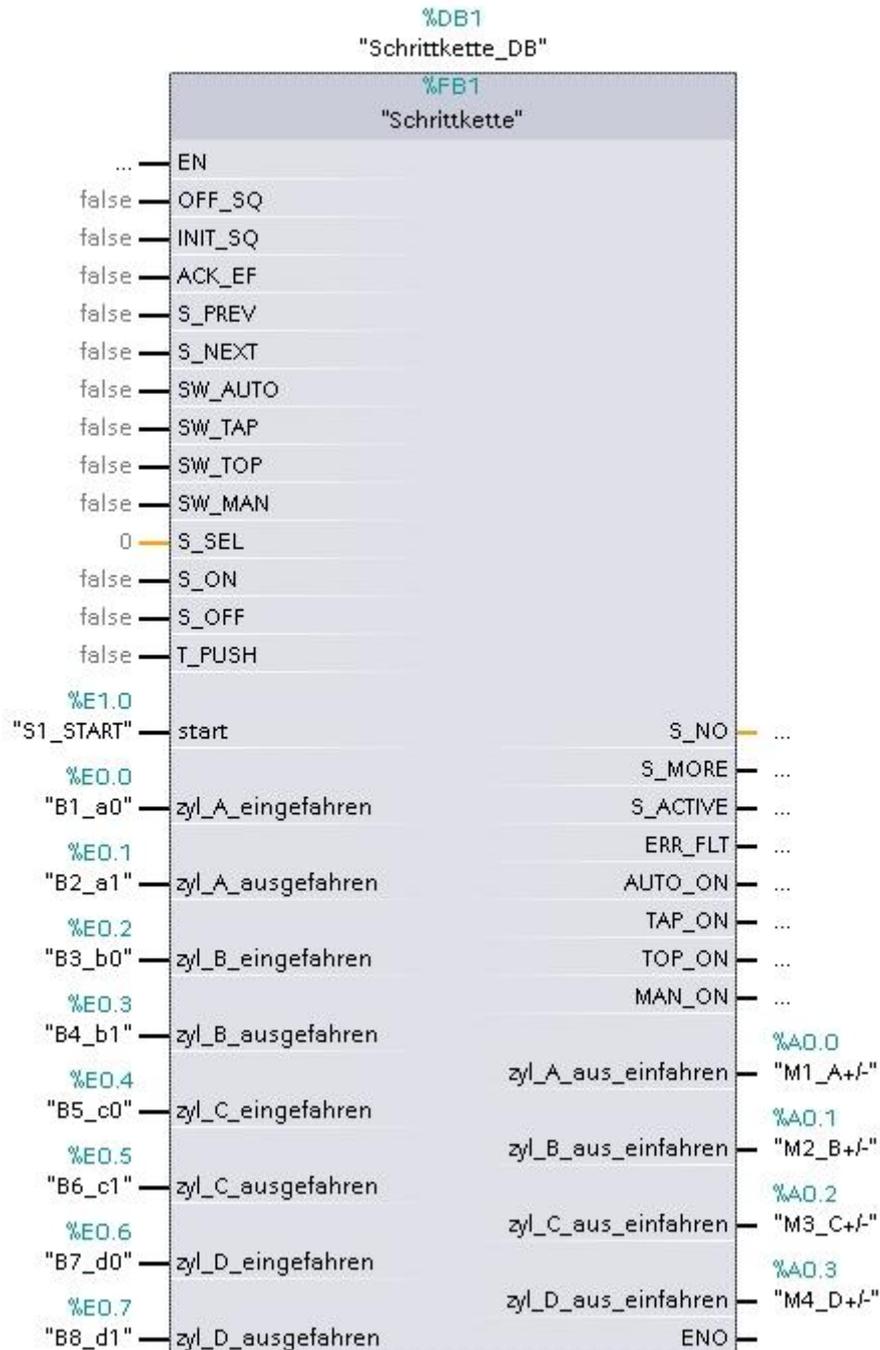
1. Main [OB1] öffnen und die Schrittkette [FB1] ins erste Netzwerk ziehen.



2. Den automatisch generierten Instanz-Datenbaustein **Schrittkette\_DB [DB1]** mit OK bestätigen.



- aufgerufenen Baustein **Schrittfolge [FB1]** beschalten.



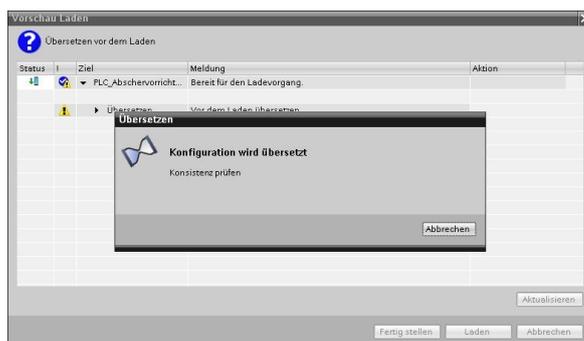
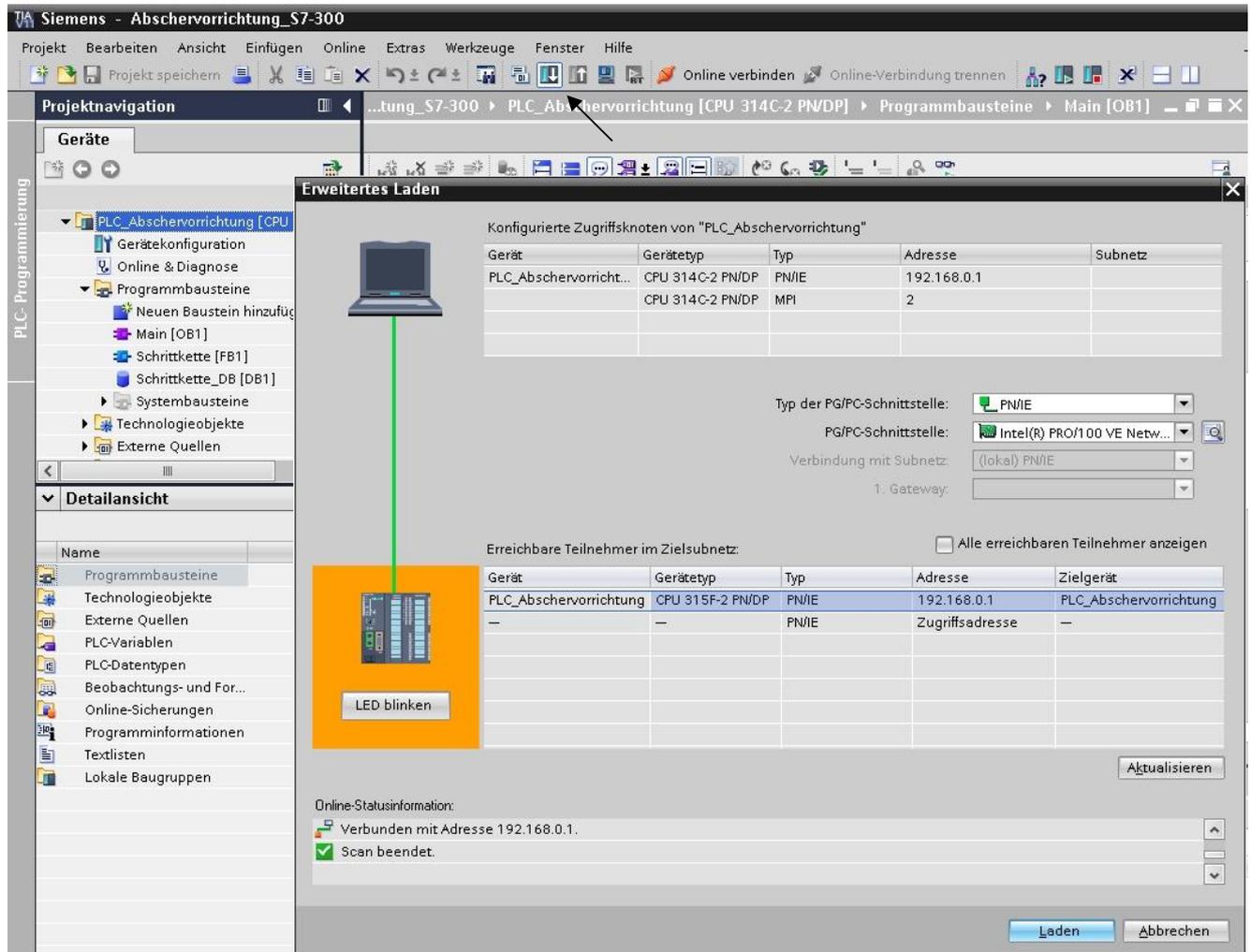
Die restlichen Variablen werden vorerst noch nicht beschaltet.

- Main [OB1] Baustein speichern.

## 5.8 Projekt Abschervorrichtung\_S7-300 in die Steuerung laden

Das Projekt muss noch in die Steuerung geladen werden.

1. **PLC\_Abschervorrichtung** in der Projektnavigation anklicken.
2. Auf die Schaltfläche **Laden in Gerät** klicken.
3. Im Fenster Erweitertes Laden **PG/PC-Schnittstelle auswählen**.
4. Nach beendeten Scanvorgang die CPU auswählen.
5. Auf die Schaltfläche **Laden** klicken.
6. Nach dem Übersetzen mit **Laden** und **Fertig stellen** bestätigen



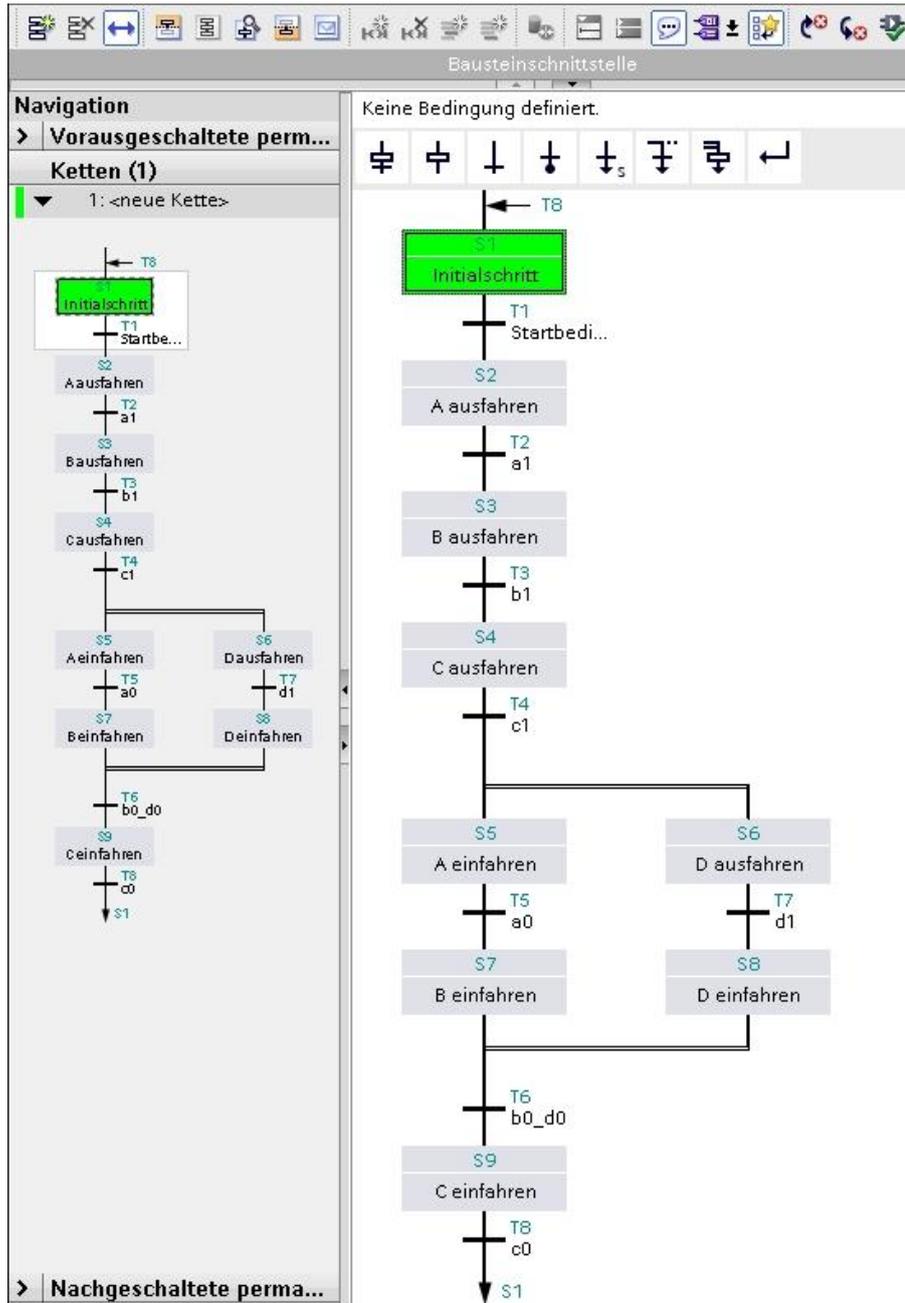
## 6. Testfunktionen

### 6.1 Beobachten der Schrittkette

Der Funktionsbaustein Schrittkette [FB1] kann in der GRAPH Ansicht getestet werden.

Nach dem Öffnen kann mit der Funktion „Beobachten“  der schrittweise Ablauf verfolgt werden.

Die aktiven Schritte und Transitionen werden farbig gekennzeichnet.



In der Einzelschrittansicht können die Aktionen und Transitionen beobachtet werden.

**Interlock -(c)-: .....**

**Supervision -(v)-: .....**

**Aktionen: .....**

Interlock	Ereignis	Kennung	Aktion
		<hinzufügen>	

**T1 - Startbedingung: .....**

FALSE #start =  
 TRUE #zyl\_A\_eingefahren =  
 TRUE #zyl\_B\_eingefahren =  
 TRUE #zyl\_C\_eingefahren =  
 TRUE #zyl\_D\_eingefahren =

Siemens - Abschervorrichtung\_S7-300

Totally Integrated Automation PORTAL

Abschervorrichtung\_S7-300 > PLC\_Abschervorrichtung [CPU 314C-2 PN/DP] > Programmbausteine > Schrittkette [FB1]

Navigation > Vorausgeschaltete perma...  
 Ketten (1)  
 1: <neue Kette>

Keine Bedingung definiert.

Ausführen

mentar

**Interlock -(c)-: .....**

**Supervision -(v)-: .....**

**Aktionen: .....**

Ununterbrochene Schrittktivierungszeit (U): T#2M51S812MS  
 Schrittktivierungszeit (T): T#2M51S812MS

Interlock	Ereignis	Kennung	Aktion
S	- Auf 1 setzen		#zyl_A_aus_eingefahren
		<hinzufügen>	

**T2 - a1: .....**

FALSE #zyl\_A\_ausgefahren =

## 6.2 Schrittkette im Testbetrieb

Im Fenster „**Testen**“ befindet sich die Testfunktion „**Kettensteuerung**“, mit der die GRAPH Ablaufsteuerung in allen Betriebsarten getestet werden kann.

Alle Eingaben und Einstellungen wirken sich wie die entsprechenden FB- Parameter aus. Die Eingaben im Dialogfeld „Kettensteuerung“ können sich von den Einstellungen unterscheiden, die der Programmierer zum Übersetzen der Kette eingerichtet hat. Die Einstellungen die hier vorgenommen werden sind den anderen Einstellungen, die beim Übersetzen eingerichtet wurden, übergeordnet.

### Einstellen der Betriebsart

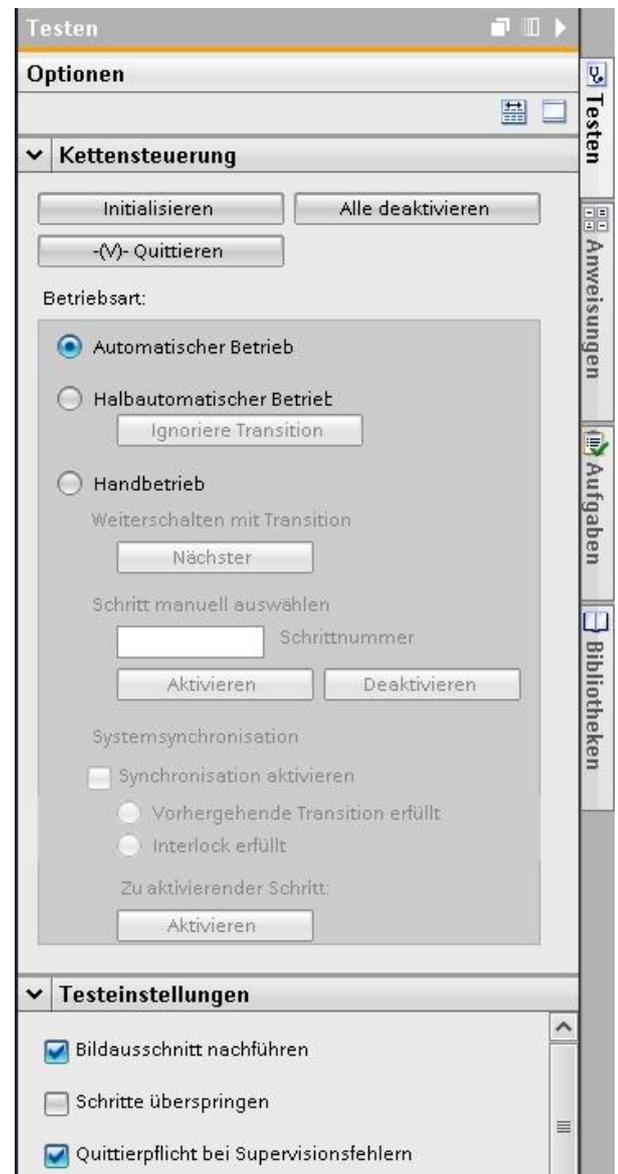
Durch Aktivieren eines der runden Optionsfelder schalten Sie die Ablaufkette in die gewählte Betriebsart. So können Sie z.B.: Ihre Ablaufkette statt im Automatikbetrieb im Handbetrieb ablaufen lassen. Im Handbetrieb können Sie jeden beliebigen Schritt aktivieren oder deaktivieren. Gehen Sie hierzu folgendermaßen vor:

1. Tragen Sie die Schrittnummer des Schritts, den Sie bearbeiten wollen, in das Feld "Schrittnummer" ein oder klicken Sie den gewünschten Schritt einfach an.
2. Wählen Sie die Aktion, die mit dem Schritt durchgeführt werden soll:
  - **Aktivieren:**  
Der angewählte Schritt wird aktiviert, auch wenn die vorherige Transition nicht erfüllt ist.
  - **Deaktivieren:**  
Der angewählte Schritt wird deaktiviert.

Beachten Sie, dass Sie beim Aktivieren eines Schritts der derzeit aktive Schritt deaktiviert wird, da in einer linearen Kette immer nur ein Schritt aktiv sein darf. Sie können also immer nur einen Schritt aktivieren, Ausnahme bei Simultanverzweigungen hier kann in jeden Zweig ein Schritt aktiviert werden.

Nachdem Sie Ihre Kette im Handbetrieb gesteuert haben, können Sie in den Automatikbetrieb zurückkehren.

Unter „Testeinstellungen“ können zusätzliche Parameter eingestellt werden.



### 6.3 Synchronisation der Schrittkette

GRAPH hilft Ihnen beim Auffinden möglicher Synchronisationspunkte zwischen Prozess und Ablaufkette. Ein Prozess ist nicht mehr synchron, wenn er von Hand in einen anderen Zustand gebracht wird. Das kann z.B. durch Umschalten in den Handbetrieb geschehen, indem Sie jeden beliebigen Schritt aktivieren können auch dann, wenn seine voraus geschaltete Transition nicht erfüllt ist.

Um anschließend den eventuell veränderten Prozess wieder automatisch laufen lassen zu können und die möglichen Synchronisationspunkte zu finden, steht die Funktion Synchronisation zur Verfügung.

**Die dem Schritt vorhergehende Transition ist erfüllt und die nachfolgende Transition ist nicht erfüllt.**

1. Schalten Sie dazu die Kette in den Handbetrieb und aktivieren Sie die Synchronisation.
2. Wählen Sie Vorhergehende Transition erfüllt.

Keine Bedingung definiert.

The screenshot displays the Siemens Step 7 LAD editor interface. On the left, a sequence diagram is shown with steps S1 through S9 and transitions T1 through T8. Step S1 is 'Initialschritt', S2 is 'A ausfahren', S3 is 'B ausfahren', S4 is 'C ausfahren', S5 is 'A einfahren', S6 is 'D ausfahren', S7 is 'B einfahren', S8 is 'D einfahren', and S9 is 'C einfahren'. Transitions T1-T8 connect the steps. Steps S5, S6, and S8 are highlighted with a cyan border. On the right, the 'Betriebsart:' (Operating Mode) panel is visible, showing options for 'Automatischer Betrieb', 'Halbautomatischer Betrieb', and 'Handbetrieb' (selected). Below this, there are options for 'System synchronisation' and 'Testeinstellungen' (Test Settings).

**Betriebsart:**

- Automatischer Betrieb
- Halbautomatischer Betrieb
- Handbetrieb

Weiterschalten mit Transition:

Schritt manuell auswählen:  Schrittnummer

**System synchronisation**

- Synchronisation aktivieren
- Vorhergehende Transition erfüllt
- Interlock erfüllt

Zu aktivierender Schritt:

**Testeinstellungen**

- Bildausschnitt nachführen
- Schritte überspringen
- Quittierpflicht bei Supervisionfehlern
- Kette anhalten
- Zeiten deaktivieren
- Alle Interlocks bearbeiten

3. Aktivieren Sie mit der rechten Maustaste die vorgeschlagenen Schritte.

The screenshot shows the SIMATIC Manager interface for step ladder logic. On the left, a navigation pane shows a sequence of steps: S1 (Initialschritt), S2 (A ausfahren), S3 (B ausfahren), S4 (C ausfahren), S5 (A einfahren), S6 (D ausfahren), S7 (B einfahren), and S8 (D einfahren). A context menu is open over step S5, with the option 'Schritt aktivieren' (Activate step) selected. The main workspace shows the step ladder logic with steps S5 and S8 highlighted in green. The right-hand panel shows the 'Betriebsart' (Operating mode) settings, with 'Handbetrieb' (Manual operation) selected. A toolbar at the top right contains buttons for 'Initialisieren', 'Alle deaktivieren', and '(-V)- Quittieren'.

4. Schalten Sie nun die Ablaufkette zurück in den Automatikbetrieb.

This screenshot shows the same step ladder logic as the previous image, but with the 'Automatischer Betrieb' (Automatic operation) mode selected in the 'Betriebsart' settings. In this mode, all steps in the sequence (S1 through S8) are highlighted in green, indicating they are active. The context menu is no longer visible. The 'Testeinstellungen' (Test settings) section at the bottom of the right-hand panel is also visible.

In einer Simultanverzweigung muss jeder Zweig einen aktiven Schritt enthalten.