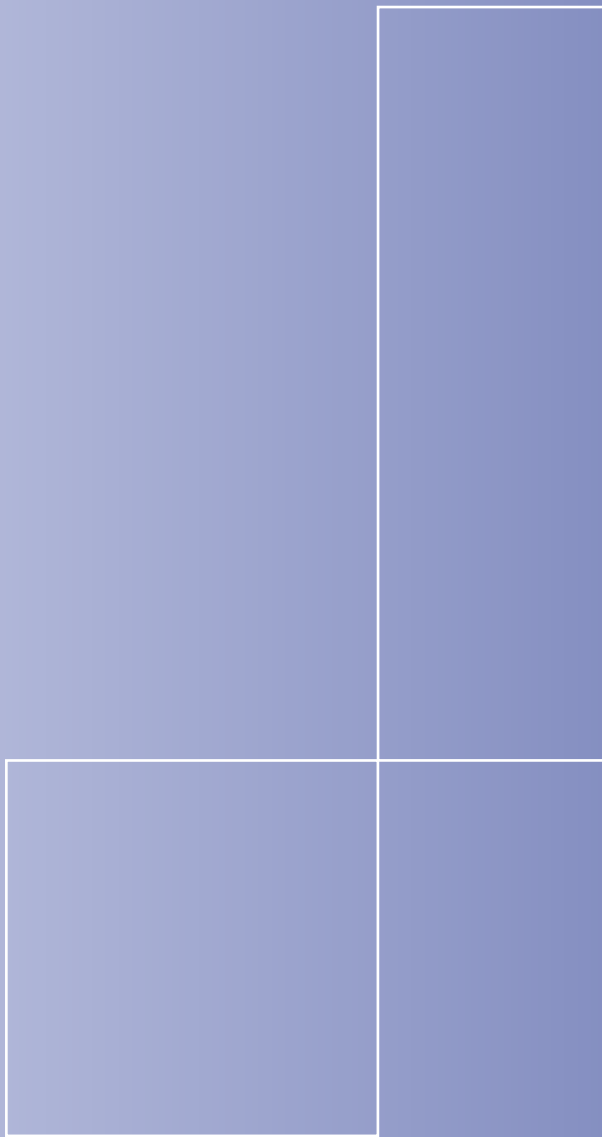


# Erste Schritte mit STEP 7 Lite V3.0

**SIMATIC**  
**STEP 7 Lite V3.0**  
Ausgabe 04/2004



**simatic**  
**STEP 7 Lite**

**SIEMENS**



# **SIEMENS**

## **SIMATIC Software**

### **Erste Schritte mit STEP 7 Lite V3.0**

Getting Started

**04/2004**

**A5E00293882-01**

## Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährungsgrad folgendermaßen dargestellt:



### Gefahr

bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



### Warnung

bedeutet, dass Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



### Vorsicht

bedeutet, dass eine leichte Körperverletzung oder ein Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

### Vorsicht

bedeutet, dass ein Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

### Achtung

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

## Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von qualifiziertem Personal vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

## Bestimmungsgemäßer Gebrauch

Beachten Sie Folgendes:



### Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden. Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

## Marken

SIMATIC®, SIMATIC HMI® und SIMATIC NET® sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen können.

### Copyright © Siemens AG 2004 All rights reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

Siemens AG  
Bereich Automation and Drives  
Geschäftsgebiet Industrial Automation Systems  
Postfach 4848, D-90327 Nürnberg

Siemens Aktiengesellschaft

### Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir Ihnen dankbar.

© Siemens AG 2004  
Technische Änderungen bleiben vorbehalten.

A5E00293882



# Willkommen zu STEP 7 Lite

... der SIMATIC Software für die Erstellung von SPS-Programmen in KOP, FUP oder AWL für SIMATIC S7-300 (inklusive SIMATIC C7), ET 200S und ET 200X. STEP 7 Lite ist konzipiert für Anwender, die unvernetzte Projekte bearbeiten und für SIMATIC-Einsteiger.

Wollen Sie SIMATIC S7-400, dezentrale Peripherie, Kommunikationsbaugruppen CP, Funktionsbaugruppen FM oder Systeme mit mehr als einer CPU einsetzen, müssen Sie die Basissoftware STEP 7 oder STEP 7 Professional verwenden.

## Informationen zu STEP 7 Lite

STEP 7 Lite ist die Software für Einsteiger, aber auch für Experten, die in erster Linie den mittleren Leistungsbereich programmieren. Mit STEP 7 Lite erstellte Programme können über Export-/Import-Funktionen in STEP 7 weiterverwendet werden. Bei der Gestaltung der Bedienoberfläche sind wir gegenüber STEP 7 neue Wege gegangen. Ausgebaute Explorerfunktionen, transparente Projektübersichten und die von Windows gewohnte Bedienphilosophie unterstützen Sie optimal beim Einstieg und beim Arbeiten mit unserer SIMATIC Software.

## Informationen zu diesem Getting Started

Hier lernen Sie die Grundlagen von STEP 7 Lite kennen. Wir zeigen Ihnen die wichtigsten Bildschirmdialoge und Bedienabläufe anhand von praktischen Übungen, die so aufbereitet sind, dass Sie fast mit jedem beliebigen Kapitel beginnen können. Beschreibungen und Bedienabläufe, die Sie durchführen müssen oder über die Sie sich informieren sollten, sind **rot** gekennzeichnet. Kurze Ausflüge in andere Bereiche sind **blau** gekennzeichnet.

0.3

## Voraussetzungen zum Arbeiten mit dem Getting Started

Um die praktischen Übungen zu STEP 7 Lite in diesem Getting Started durchführen zu können, benötigen Sie

- ein SIMATIC Programmiergerät oder einen PC,
- das STEP 7 Lite Softwarepaket und die Autorisierungsdiskette,
- ein Automatisierungssystem SIMATIC S7-300.

Beachten Sie auch die Tabelle mit den Bestellnummern in Kap. 1.

## Weitere Dokumentation

- Auf der CD und nach der Installation von STEP 7 Lite über die Startleiste **Start > Siemens > Dokumentation** finden Sie das elektronische Handbuch "Programmieren mit STEP 7 Lite" zum Ausdrucken.

Wir wünschen Ihnen viel Spaß und Erfolg!

Ihre SIEMENS AG

# Übersicht der Beispielprojekte zum Getting Started

Nach der Installation der STEP 7 Lite Software finden Sie, wenn Sie kein anderes Verzeichnis angegeben haben, im Verzeichnis <Laufwerk>:\Siemens\S7lite\Examples\Deutsch ... die folgenden Programmierbeispiele.

Auf diese Beispiele bezieht sich das vorliegende Getting Started.

- **erste\_schritte\_awl.k7p**
- **erste\_schritte\_fup.k7p**
- **erste\_schritte\_kop.k7p**

Alle Beispiele sind identisch, sie unterscheiden sich lediglich in der verwendeten Programmiersprache.

## Teil 1: Die ersten Schritte mit STEP 7 Lite – die wichtigsten Basiskenntnisse

<b>Überblick und Installation</b>	<b>1</b>
Was werden Sie lernen?	1.2
Zusammenspiel von Hardware und Software	1.4
Leitfaden durch STEP 7 Lite	1.6
Installieren von STEP 7 Lite	1.8
<b>Starten und Bedienen</b>	<b>2</b>
Beispielprojekt öffnen	2.2
Projekthandling	2.6
Aufrufen der Hilfefunktionen	2.8

## Teil 2: Erstellen einer Automatisierungslösung mit STEP 7 Lite

<b>Umsetzen der Aufgabenstellung</b>	<b>3</b>
Aufgabenstellung - Motorenteststand	3.2
Zerlegen Sie den Prozess	3.4
<b>Konfigurieren der Baugruppen</b>	<b>4</b>
Was passiert beim Konfigurieren?	4.2
Neues Projekt anlegen	4.4
In der Sicht Hardware-Konfiguration arbeiten	4.6
Baugruppen parametrieren	4.12
Konfigurationsdaten speichern	4.14
Hardware-Konfiguration in die CPU laden	4.16
<b>Erstellen der Symboltabelle</b>	<b>5</b>
Absolut programmieren	5.2
Symbolisch programmieren	5.4

<b>Erste Programmierschritte</b> .....	<b>6</b>
Entscheiden Sie sich für KOP, FUP, AWL .....	6.2
Im Bausteineditor arbeiten .....	6.4
OB 1 mit KOP programmieren .....	6.6
OB 1 mit AWL programmieren .....	6.12
OB 1 mit FUP programmieren .....	6.18
Querverweise anzeigen .....	6.24
 <b>Verwenden von Funktionsbausteinen</b> .....	 <b>7</b>
Funktionsbaustein (FB) anlegen und öffnen .....	7.2
FB in KOP programmieren .....	7.6
FB in AWL programmieren .....	7.8
FB in FUP programmieren .....	7.10
Instanz-Datenbausteine erzeugen und Aktualwerte ändern .....	7.12
Bausteinaufruf in KOP programmieren .....	7.14
Bausteinaufruf in AWL programmieren .....	7.16
Bausteinaufruf in FUP programmieren .....	7.18
 <b>Verwenden von Funktionen</b> .....	 <b>8</b>
Funktion (FC) anlegen und öffnen .....	8.2
Funktion programmieren .....	8.6
Funktion im OB 1 aufrufen .....	8.8
 <b>Verwenden von globalen Datenbausteinen</b> .....	 <b>9</b>
Globalen Datenbaustein (DB) anlegen und öffnen .....	9.2
Variablen im DB programmieren .....	9.4

**Teil 3: Laden, Testen und Diagnose**

**Laden des Programms in die CPU..... 10**

    Online-Verbindung aufbauen und Online gehen ..... 10.2

    CPU urlöschen und Programm übertragen ..... 10.6

**Testen des Programms ..... 11**

    Programm mit Programmstatus testen..... 11.2

    Variablen beobachten und steuern ..... 11.6

**Fehlerdiagnose ..... 12**

    HW-Diagnose auf einen Blick ..... 12.2

    Baugruppenzustand und Fehlerhistorie ..... 12.5

**Index ..... 13**

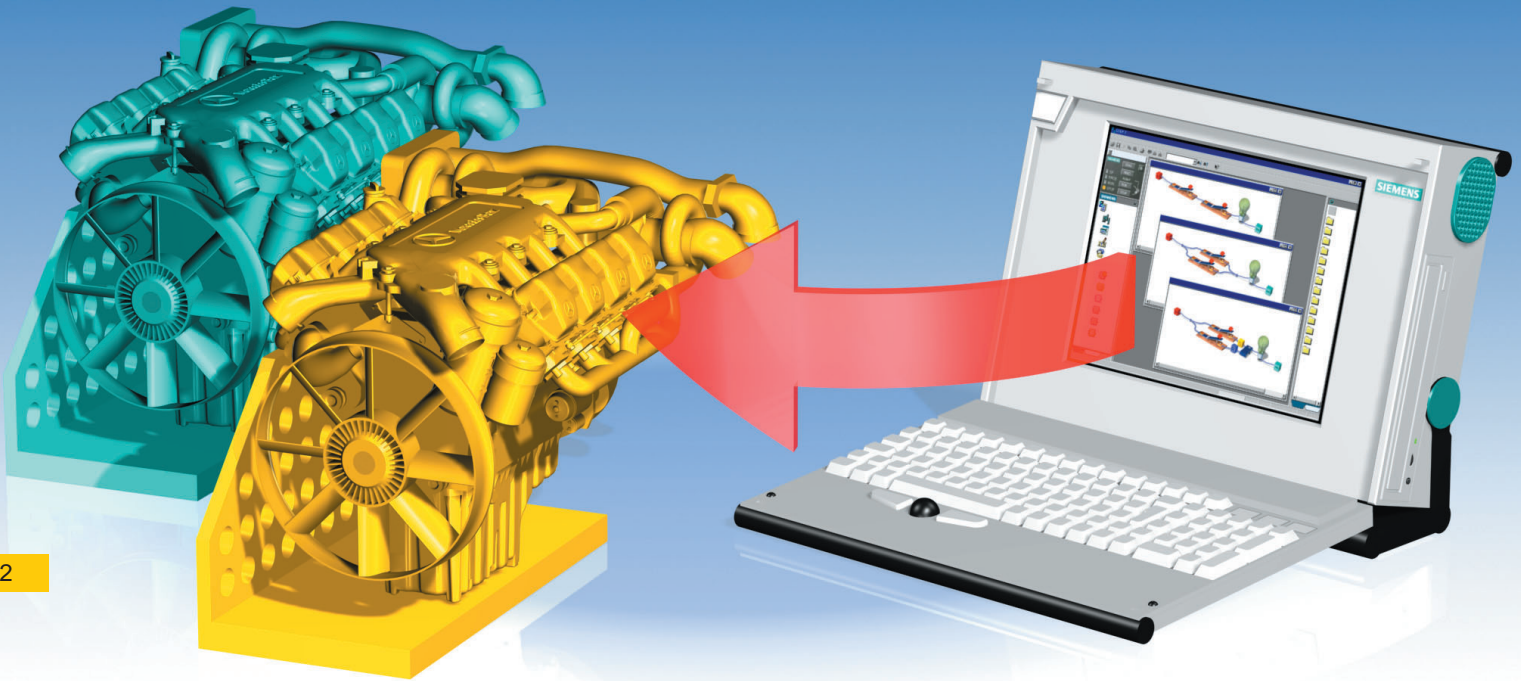


# 1

# Überblick und Installation



# Was werden Sie lernen?



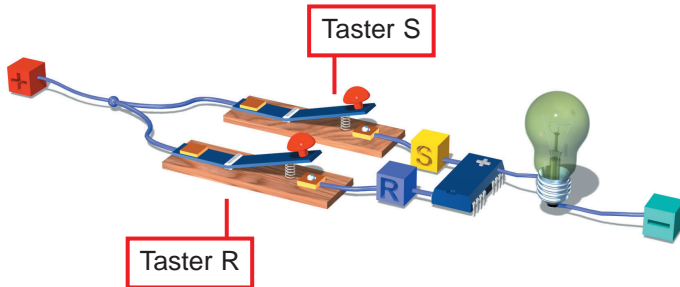
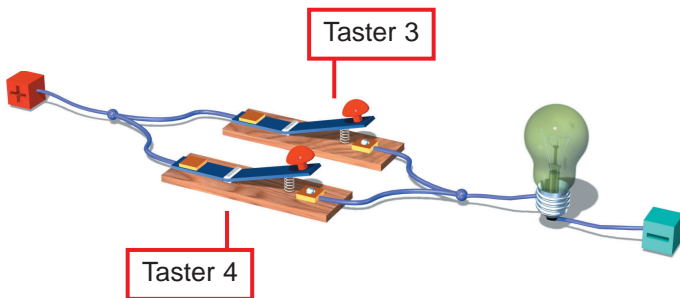
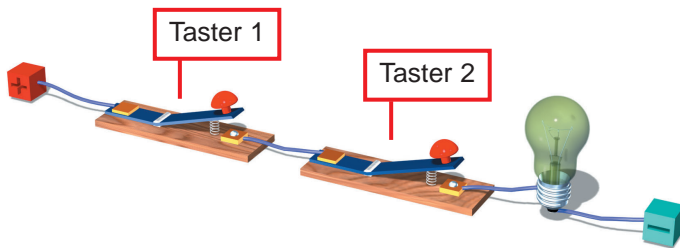
1.2

Anhand praktischer Übungen zeigen wir Ihnen, wie einfach das Programmieren in den Programmiersprachen KOP (Kontaktplan), FUP (Funktionsplan) und AWL (Anweisungsliste) mit STEP 7 Lite ist.

Hierzu legen Sie zunächst ein Projekt mit dem Namen "Getting Started" an. In diesem Projekt erstellen Sie ein SPS-Programm mit den einfachen binären Verknüpfungen UND, ODER, SPEICHERGLIED.

Dann bauen Sie dieses Projekt zu einem SPS-Programm für einen Motor-teststand aus.





## Basis-Know-how

Unsere Programmierbeispiele bauen auf drei fundamentalen, binären Verknüpfungen auf:

### Reihenschaltung

Die erste binäre Verknüpfung, die Sie später programmieren, ist die UND-Funktion. Die UND-Funktion kann in einer elektrischen Schaltung mit zwei Tastern verdeutlicht werden.

Wird Taster 1 **und** Taster 2 gedrückt, brennt die Lampe.

### Parallelschaltung

Die zweite binäre Verknüpfung ist die ODER-Funktion. Die ODER-Funktion kann ebenfalls in einer elektrischen Schaltung dargestellt werden.

Wird Taster 3 **oder** Taster 4 gedrückt, brennt die Lampe.

### Speicherglied

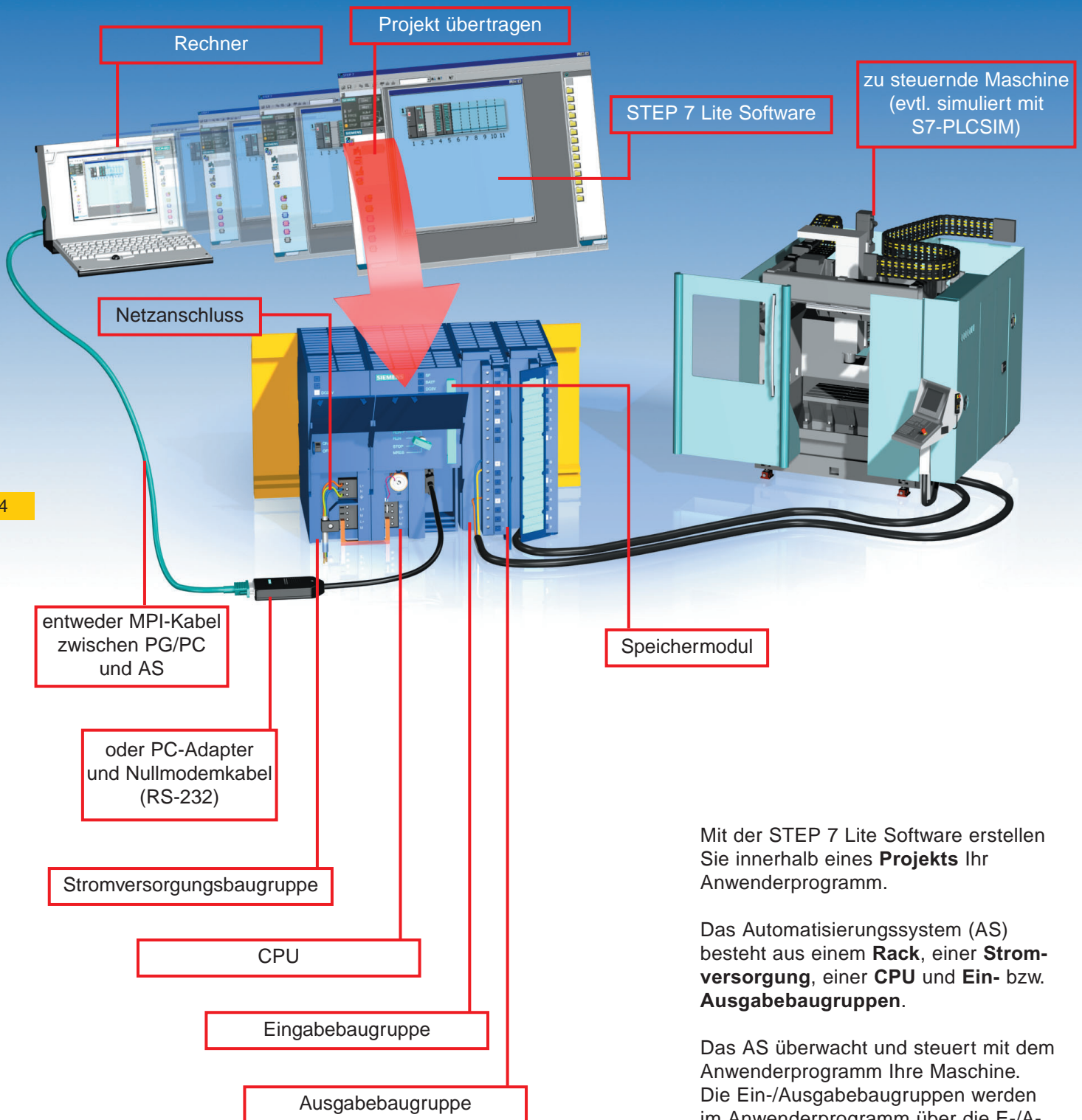
#### (Flipflop setzen/rücksetzen)

Die dritte binäre Verknüpfung ist das Speicherglied. Es reagiert in einer elektrischen Schaltung auf bestimmte Spannungszustände und gibt diese entsprechend weiter.

Wird Taster S gedrückt, brennt die Lampe **solange**, bis Taster R gedrückt wird.

Alle drei Verknüpfungen bauen Sie zu einem konkreten Beispiel aus – zu einem Motorteststand. Hierbei lernen Sie folgende STEP 7 Lite Programmelemente kennen: Organisationsbausteine, Funktionsbausteine, Instanz-Datenbausteine, Funktionen, globale Datenbausteine.

## Zusammenspiel von Hardware und Software



Mit der STEP 7 Lite Software erstellen Sie innerhalb eines **Projekts** Ihr Anwenderprogramm.

Das Automatisierungssystem (AS) besteht aus einem **Rack**, einer **Stromversorgung**, einer **CPU** und **Ein-** bzw. **Ausgabebaugruppen**.

Das AS überwacht und steuert mit dem Anwenderprogramm Ihre Maschine. Die Ein-/Ausgabebaugruppen werden im Anwenderprogramm über die E-/A-Adressen angesprochen.

## Checkliste Komponenten

Wollen Sie das beschriebene Beispielprojekt erstellen, benötigen Sie folgende Komponenten.

<b>SPS-Station</b>	
	Bestellnummern:
Stromversorgung (PS 307 2A)	6ES7307-1BA00-0AA0
Zentralbaugruppe (CPU 315)	6ES7315-1AF03-0AB0
Digitaleingabe (SM32DI 16xDC24V)	6ES7321-1BH02-0AA0
Digitalausgabe (SM322 DO 16xDC24V/0,5A)	6ES7322-1BH01-0AA0
Pufferbatterie (Li) 3,4V	6ES7971-1AA00-0AA0
Profilschiene 480 mm	6ES7390-1AE80-0AA0
<b>Rechner</b>	
SIMATIC-Programmiergerät Power PG, Field PG oder Handelsüblicher PC mit CP 5611	<a href="http://www.ad.siemens.de/simatic-pg">www.ad.siemens.de/ simatic-pg</a>
Betriebssystem Windows 2000 oder Windows XP Home oder Professional Edition Internet-Explorer ab 5.0	
<b>Software</b>	
Software STEP 7 Lite (Floating License)	6ES7810-3CC07-0YA5
<b>Dokumentation</b>	
Erste Schritte mit STEP 7 Lite V3.0	(auf der Software-CD)
<b>Optionspaket</b>	
Simulationssoftware S7-PLCSIM (Floating License)	6ES7841-0CC04-0YA5
Simulationssoftware S7-PLCSIM (Upgrade)	6ES7841-0CC04-0YE5

1

Mit STEP 7 Lite lassen sich Komponenten aus den Baureihen S7-300, ET 200S und ET 200X programmieren. Die Baugruppen, die im Beispielprojekt verwendet wurden, stehen in Klammern. Sie können natürlich auch andere Baugruppen aus den genannten Baureihen verwenden.

2

Wir empfehlen den Einsatz unserer SIMATIC-PGs. Sie sind auch der harten Beanspruchung in der Industrieumgebung gewachsen. Für handelsübliche PCs benötigen Sie eine Erweiterungskarte mit Schnittstellenkabel. Bei SIMATIC-PGs ist diese Schnittstelle bereits integriert.

3

Installationshinweise finden Sie auf der CD in der Datei STEP7Lite\Disk1\Liesmich.WRI.

4

Zusätzlich zum Handbuch "Erste Schritte mit STEP 7 Lite" wird auf der Software-CD noch das elektronische Handbuch "Programmieren mit STEP 7 Lite" und die Online-Hilfe mitgeliefert.

5

S7-PLCSIM simuliert ein abgeschlossenes Automatisierungssystem. S7-PLCSIM ist dann sinnvoll, wenn Sie keine Hardware vor Ort haben, die Lauffähigkeit eines Programms aber testen wollen.

## Leitfaden durch STEP 7 Lite

### Projektieren

Lösung der Automatisierungsaufgabe entwerfen  
Kapitel 3



### Projekt anlegen

Kapitel 4



### Hardware konfigurieren

Kapitel 4



### Programm erstellen

Kapitel 5 – 9



### Programm in die CPU übertragen

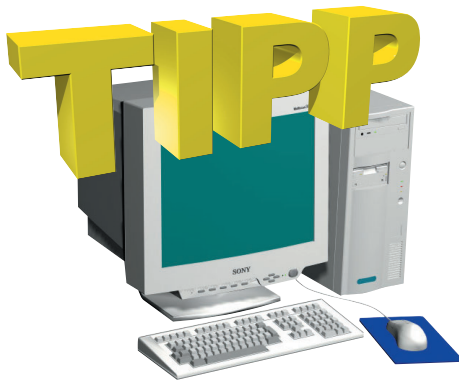
Kapitel 10



### Programm testen

Kapitel 11

Das Projekt ist das zentrale Element in STEP 7 Lite. Innerhalb des Projektes lösen Sie Ihre gesamte Automatisierungsaufgabe – vom Konfigurieren der Hardware bis zum Testen des Programms auf seine Lauffähigkeit.



Bei umfangreichen Programmen mit vielen Ein- und Ausgängen empfehlen wir zunächst die Hardware zu konfigurieren. Dies hat den Vorteil, dass STEP 7 Lite verfügbare Adressen in der Hardware-Konfiguration oder beim Programmieren als "Spickzettel" über den Menübefehl **Ansicht > Adressübersicht** anzeigt.

Erstellen Sie zuerst das Programm, müssen Sie in Abhängigkeit Ihrer gewählten Komponenten die jeweiligen Adressen selbst ermitteln und können diese nicht über STEP 7 Lite abrufen.

Mit der Hardware-Konfiguration können Sie nicht nur Adressen festlegen, sondern auch die Parameter und Eigenschaften von Baugruppen ändern.

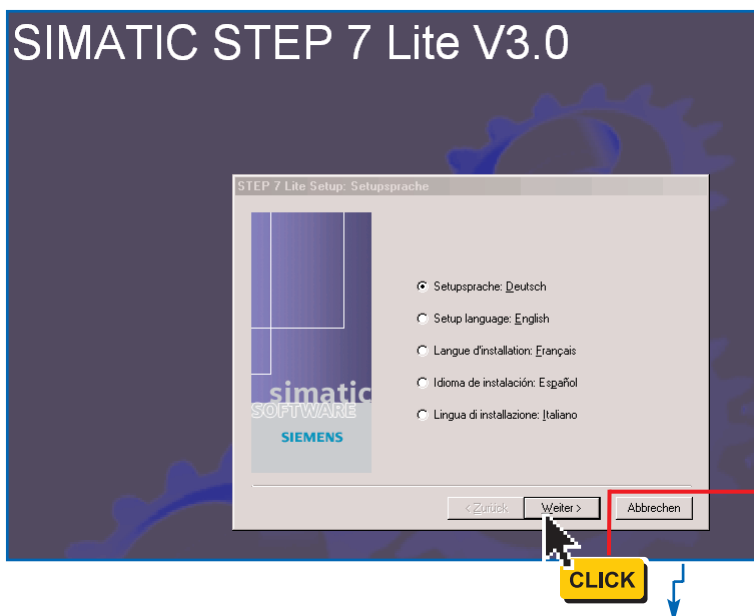
Da im "Getting Started" nur wenige Ein- und Ausgänge notwendig sind, können Sie die Hardware-Konfiguration überspringen und sofort mit dem Programmieren beginnen, wenn Sie dies bevorzugen.

## Installieren von STEP 7 Lite



1.8

### SIMATIC STEP 7 Lite V3.0



Zur Installation benötigen Sie

- die STEP 7 Lite CD mit Installationshinweisen in der Datei **STEP7Lite\Disk1\Liesmich.WRI**,  
und
- den entsprechenden License Key als Nutzungsberechtigung.

1 Legen Sie die STEP 7 Lite CD ein. Das Installationsprogramm wird automatisch oder über **<Laufwerk>:\setup.exe** gestartet.

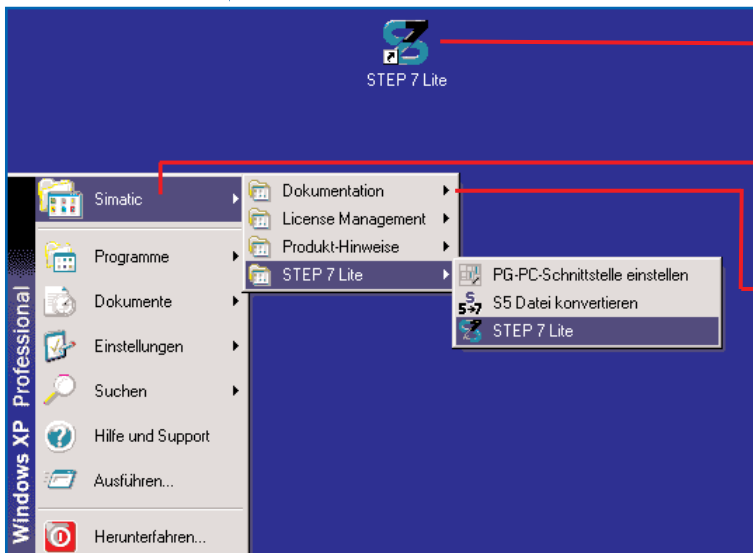
Folgen Sie den Installationsanweisungen.

Legen Sie bei Aufforderung den Daten-  
träger mit dem License Key ein.

Befolgen Sie die Anweisungen zur  
Installation des License Keys.

Entnehmen Sie vor einem Neustart den  
Datenträger.

#### Desktop nach der Installation

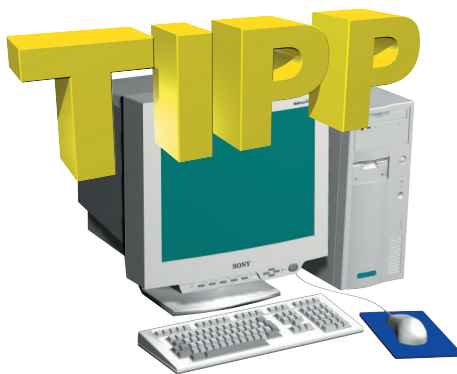


2 Nach der Installation wird STEP 7 Lite  
auf dem Desktop und im Start-Menü  
angezeigt.

3 Jede weitere SIMATIC-Software, die  
Sie installieren, ist über diesen  
SIMATIC-Ordner aufrufbar.

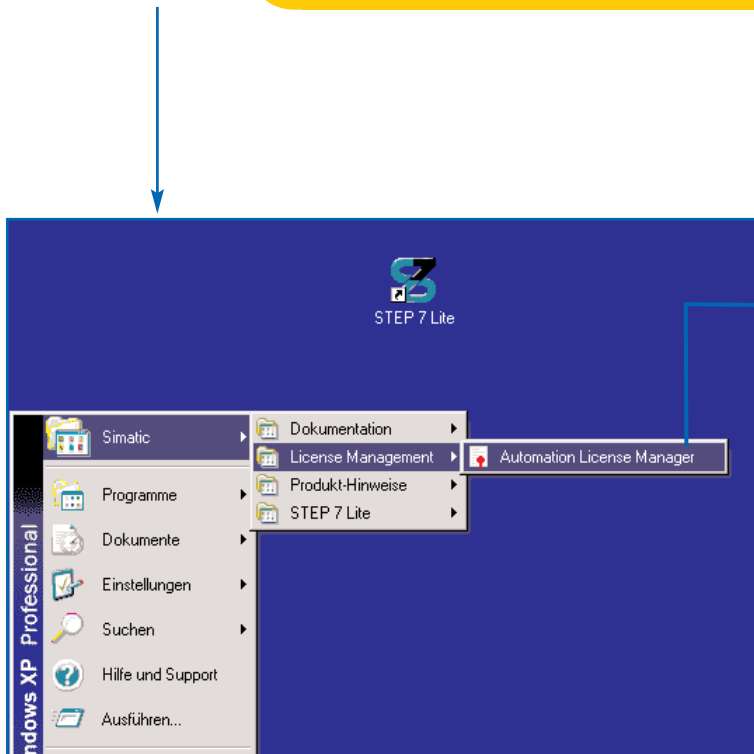
4 Die STEP 7 Lite Dokumentation zum  
Ausdrucken finden Sie unter **Simatic >  
Dokumentation**.

#### License Key transferieren



Wenn für STEP 7 Lite kein gültiger License Key  
installiert ist, wird ein Trial License Key  
verwendet, der mit STEP 7 Lite standardmäßig  
mitgeliefert wird. Mit dem Trial License Key kann  
STEP 7 Lite allerdings nur 14 Tage genutzt  
werden. Mit dem ersten Start von STEP 7 Lite  
ohne einen gültigen License Key wird die Trial  
License aktiviert.



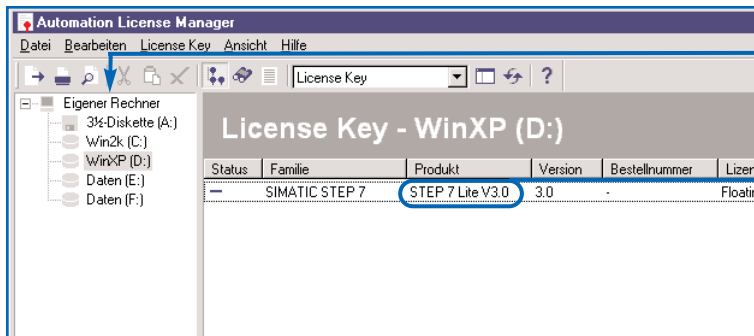


5

Wollen Sie den License Key von einem Rechner auf einen anderen Rechner transferieren, gehen Sie folgendermaßen vor:

Starten Sie den **Automation License Manager**.

Benutzungsoberfläche des  
Automation License Managers



6

Öffnen Sie das Laufwerk, auf dem sich der zu übertragende License Key befindet.

7

Selektieren Sie den License Key und wählen Sie den Menübefehl **License Key > Übertragen**.

Wählen Sie im nachfolgenden Dialogfeld das Laufwerk auf dem Zielrechner aus, auf das der License Key übertragen werden soll.



2

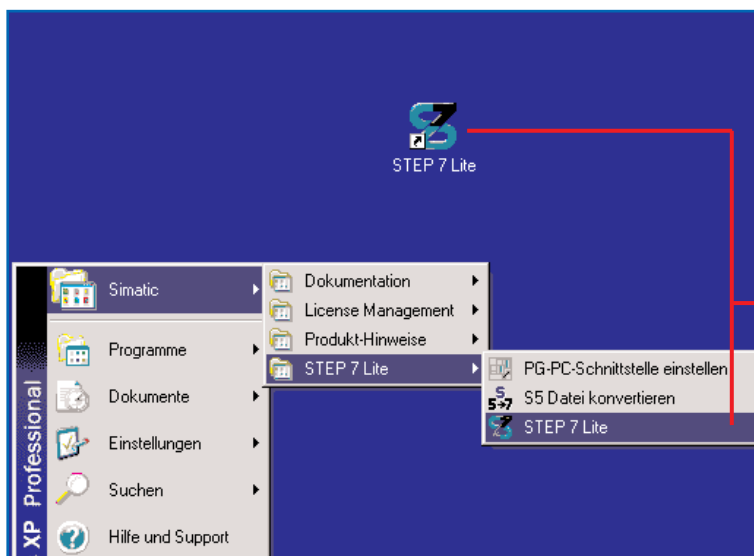
Starten  
und  
Bedienen



## Beispielprojekt öffnen



2.2

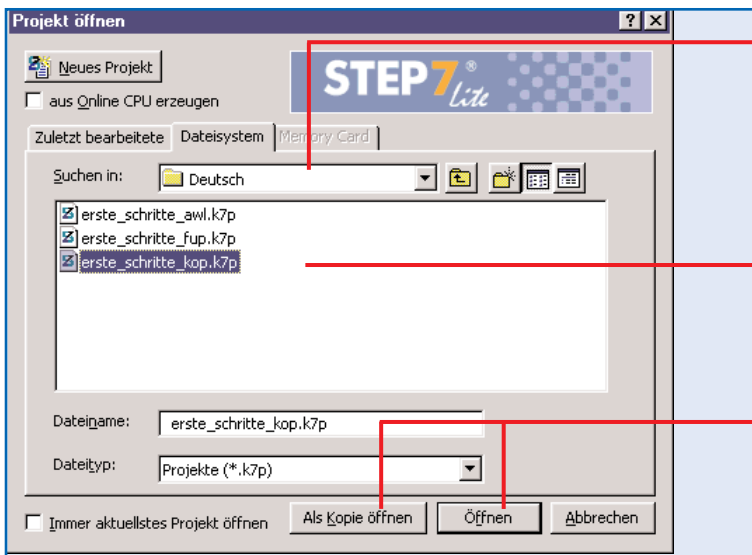


### Beispielprojekt KOP öffnen

Sie haben STEP 7 Lite bereits auf Ihrem Rechner installiert.

In diesem Kapitel erfahren Sie das Wichtigste über die Benutzungsoberfläche.

Starten Sie STEP 7 Lite über das Start-Menü oder das Symbol auf dem Desktop.



2 Rufen Sie das Verzeichnis auf, in dem sich die Programmierbeispiele befinden

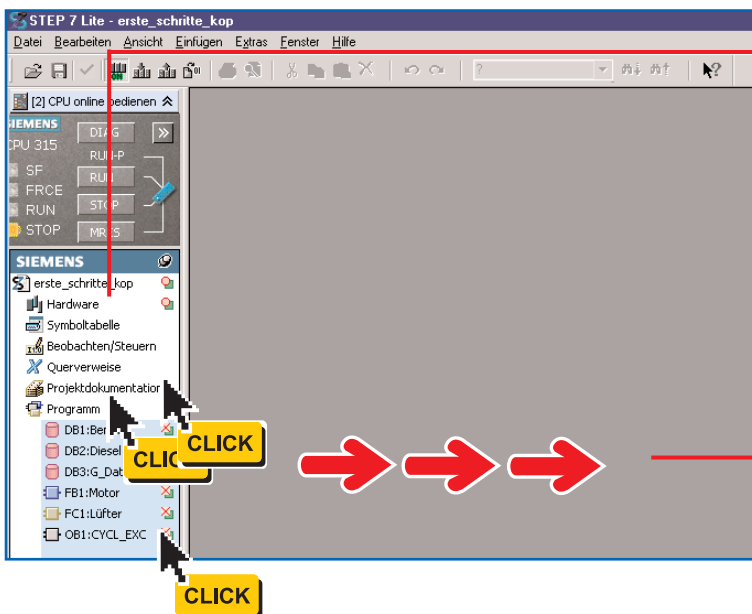
<Laufwerk>:\Siemens\S7Lite  
\Examples\Deutsch ...

3 Wählen Sie hier das KOP-Beispiel.

**erste\_schritte\_kop.k7p**

4 Öffnen Sie das Projekt.  
Wollen Sie sichergehen, dass das mitgelieferte Beispielprojekt nicht unbeabsichtigt geändert wird, öffnen Sie das Projekt als Kopie.

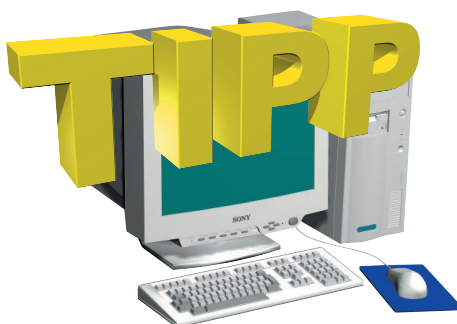
Das gewählte Projekt wird geöffnet



5 Im Projektfenster wird das Projekt "erste\_schritte\_kop" angezeigt.

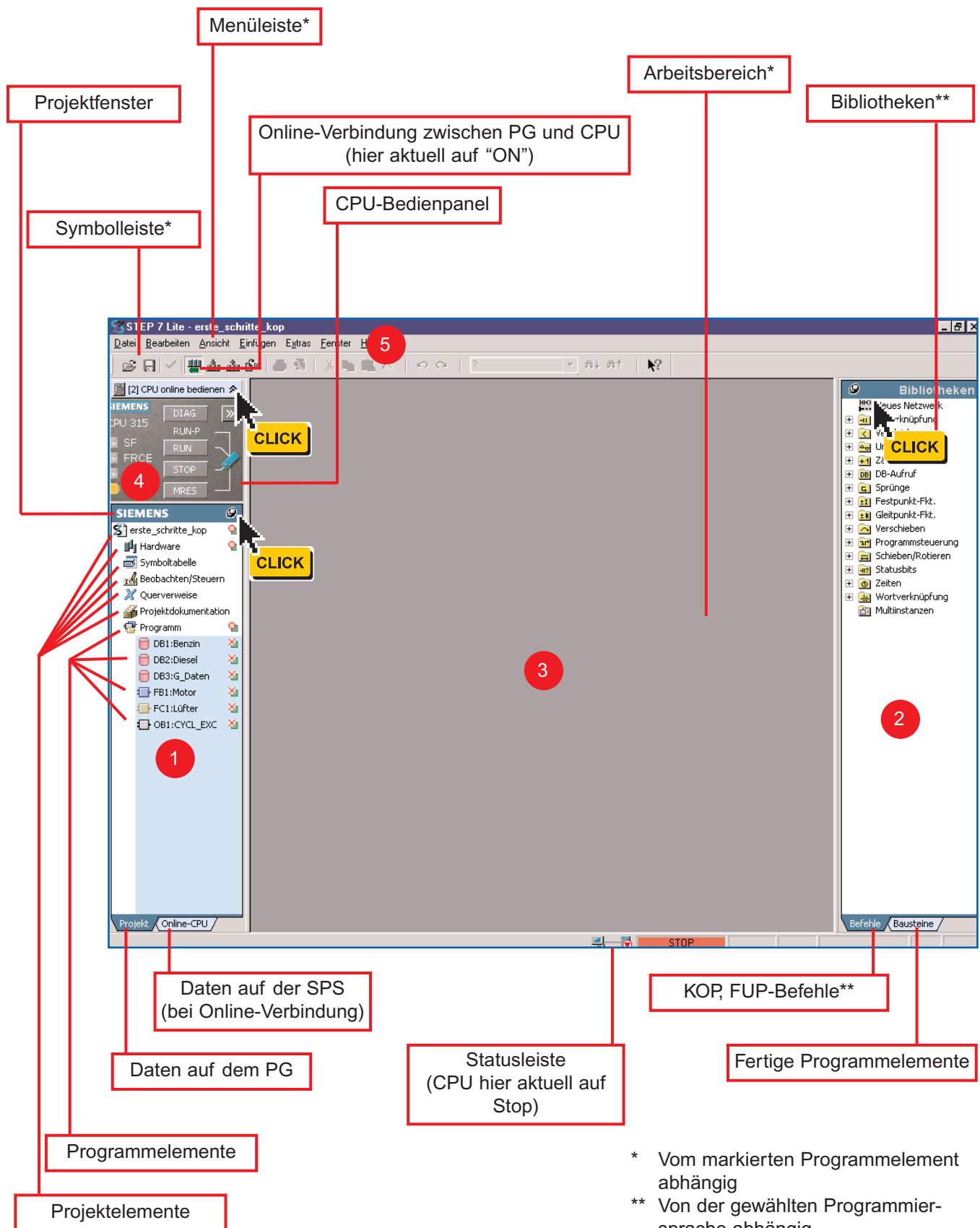
Das Projektfenster links ist das zentrale Navigationsmittel von STEP 7 Lite. Von hier aus verzweigen Sie über die Projektelemente "Hardware", "Symboltabelle" usw. in alle Sichten von STEP 7 Lite.

6 Doppelklicken Sie auf die einzelnen Elemente. Hier im grauen Arbeitsbereich werden die Sichten geöffnet und die Menüleiste oben den Sichten angepasst.



Durch Doppelklicken auf die Projektelemente öffnen Sie nacheinander alle Sichten von STEP 7 Lite. Schließen Sie hin und wieder nicht benötigte Fenster, um den Überblick zu behalten.

# Starten und Bedienen



- \* Vom markierten Programmelement abhängig
- \*\* Von der gewählten Programmiersprache abhängig

## Die Benutzungsoberfläche

Die Benutzungsoberfläche besteht aus 5 Bereichen:

- 1 **Projektfenster**  
Wenn Sie ein neues Projekt anlegen, sind alle Projektelemente, die Sie benötigen, bereits angelegt.
- 2 **Bibliotheken**  
Unter "Bausteine" finden Sie mitgelieferte Bausteine. Unter "Befehle" finden Sie KOP- und FUP-Befehle zur Verwendung in den Bausteinen.
- 3 **Arbeitsbereich**  
Hier werden die Sichten geöffnet, in denen Sie die Projektelemente bearbeiten können.
- 4 **CPU-Bedienpanel**  
Stellt die Frontseite der CPU mit Anzeige- und Bedienelementen dar und erlaubt die Änderung des Betriebszustands.
- 5 **Menüleiste**  
Enthält alle Menüs, die in STEP 7 Lite zur Verfügung stehen – z. B. bei geöffnetem Baustein zum Umschalten der Programmiersprache den Menübefehl **Ansicht > KOP**.



Durch Klicken auf das Erweiterungssymbol können Sie das CPU-Bedienpanel ein- oder ausblenden. Durch Klicken auf die Pinnadel wird die Ansicht des Projektfensters und der Bibliotheken arretiert oder nicht arretiert. Bei nicht arretierter Ansicht, kann durch Verschieben des Mauszeigers zum Rand hin der Arbeitsbereich vergrößert oder verkleinert werden.



## Projekthandling

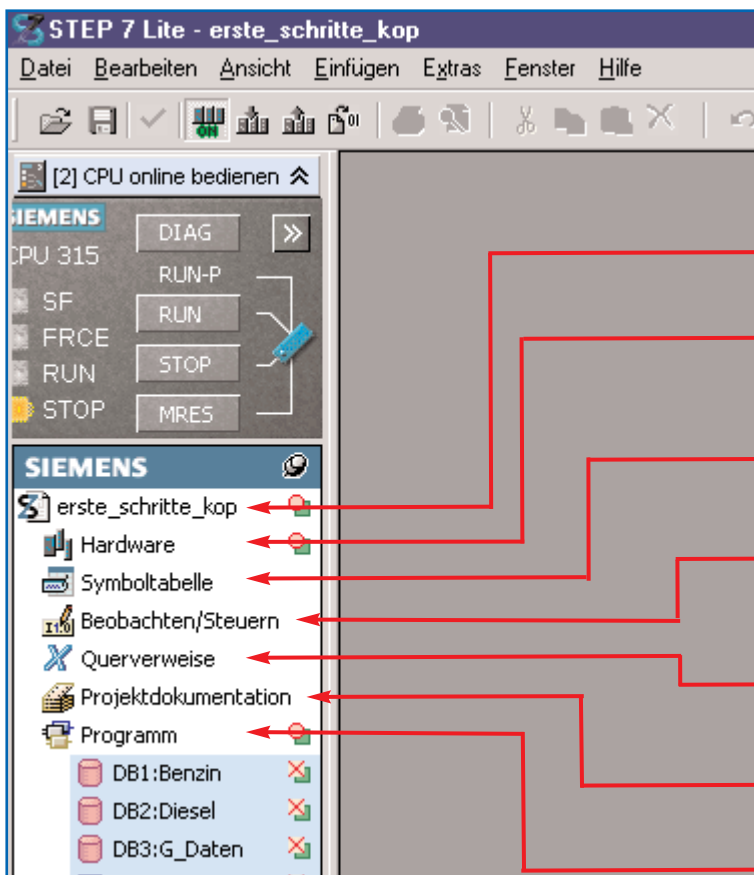


2.6

### Was ist ein Projekt?

Die Projektdaten eines STEP 7 Lite Projekts umfassen alle Daten einer SIMATIC S7-300, C7 oder einem modularen Dezentralen Peripheriesystem ET 200X bzw. ET 200S (stand alone).

Projekte dienen dazu, die bei der Erstellung einer Automatisierungslösung anfallenden Daten geordnet abzulegen.



## Projekthandling

Mit den Elementen eines Projektes sind die folgenden Arbeitsschritte verbunden:

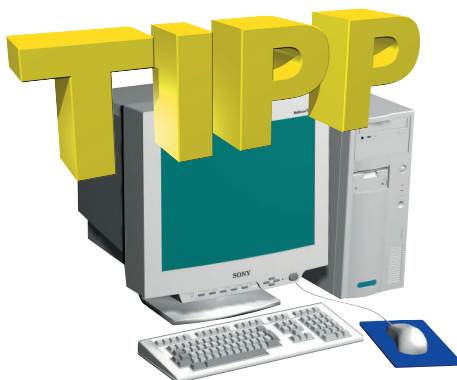
- 1 Projekt anlegen, speichern
- 2 HW konfigurieren, Baugruppen parametrieren und HW-Fehler diagnostizieren
- 3 Symbole festlegen für symbolisches Programmieren
- 4 Zum Testen des Programms Operanden in der CPU beobachten, steuern, forcen
- 5 Verwendete Operanden und Programmstruktur auswerten
- 6 Projektdokumentation individuell zusammenstellen
- 7 SPS-Anwenderprogramm aus Bausteinen erstellen.

2.7

## Dateihandling

Das Projekt speichern Sie unter seinem Namen im Datenformat ... **.k7p**.

In STEP 7 Lite können Sie immer nur eine **.k7p-Datei** öffnen.



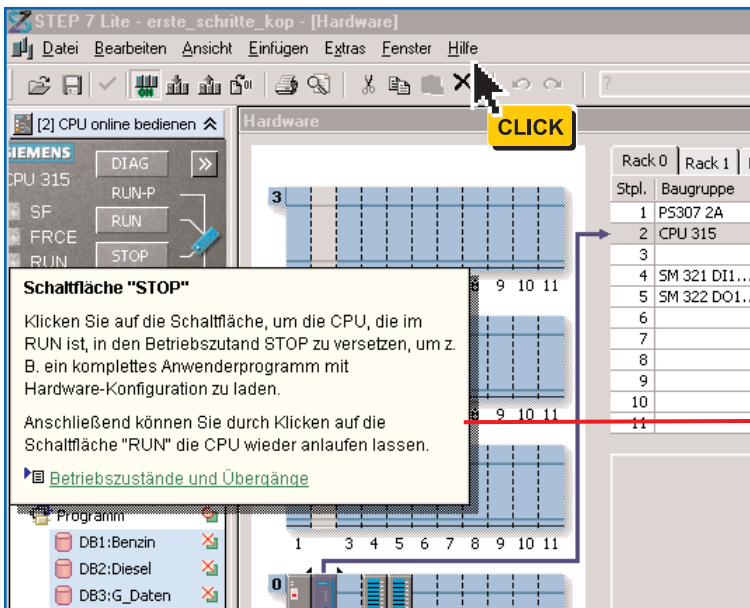
**Unfallgefahr** – Bei Online-Verbindung zwischen Programmiergerät und CPU können Sie über das CPU-Bedienpanel z. B. Bewegungen in einer Anlage auslösen.

Daher nie "RUN" wählen, wenn Sie nicht ausschließen können, dass Personen gefährdet werden können.

## Aufrufen der Hilfefunktionen



2.8



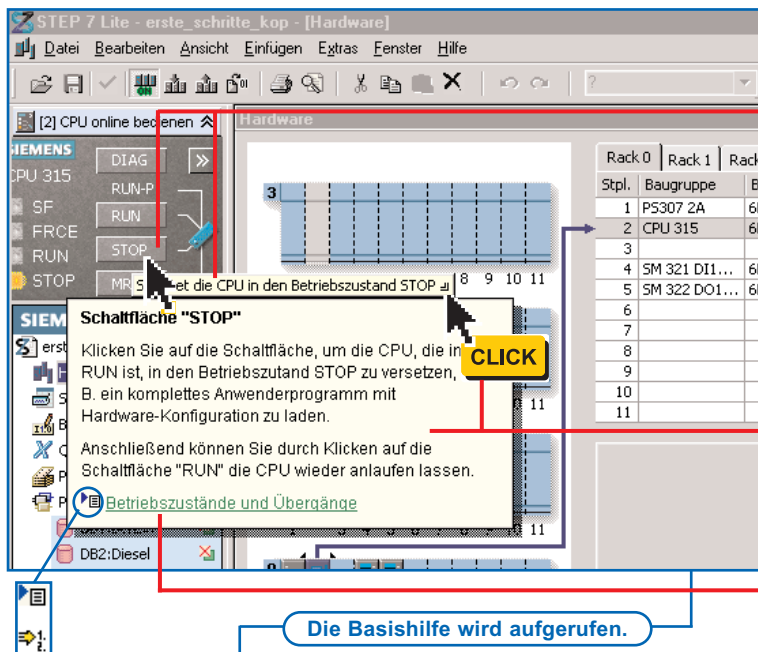
### Windows-Hilfe

Sofern Sie schon mit Microsoft Programmen gearbeitet haben, fällt Ihnen der Umgang mit der STEP 7 Lite-Hilfe nicht schwer.

- 1 Drücken Sie **F1**:  
Sie rufen die ausführliche Basishilfe auf.
- 2 Drücken Sie **Shift + F1**:  
Klicken Sie anschließend mit dem Fragezeichen-Cursor auf eine Schaltfläche. Sie rufen die Direkthilfe für diese Schaltfläche auf.

Alternativ rufen Sie diese beiden Hilfearten über **Hilfe** in der Menüleiste auf.





## Dreigeteilte Hilfe

### Kurzhilfe

Stellen Sie den Cursor ohne Click z. B. auf die Schaltfläche **STOP**.

Nach einigen Sekunden wird die Kurzhilfe angezeigt – mit Text wie links, sofern eine Online-Verbindung besteht.

### Direkthilfe

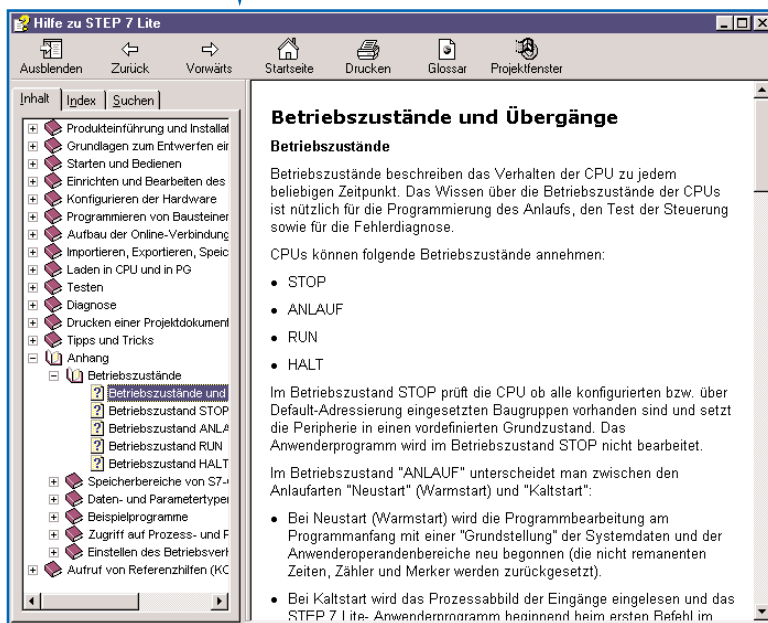
Klicken Sie auf die kleinen Pfeile. Die Direkthilfe wird zusätzlich angezeigt.

### Basishilfe

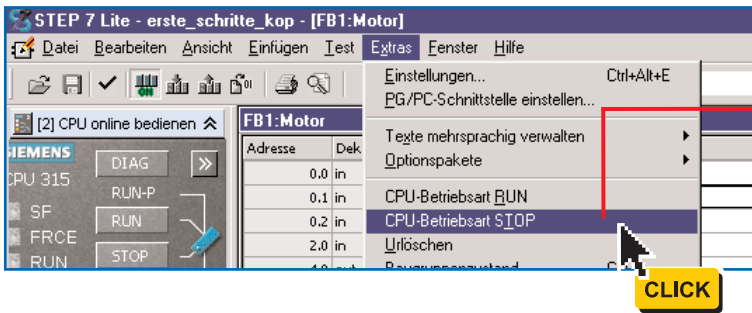
Klicken Sie auf den Link. Die ausführliche Basishilfe zum markierten Thema wird in einem eigenen Fenster aufgerufen.

Hinweis:

Die unterschiedlichen Symbole der Links kennzeichnen die Hilfeart, die in der Basishilfe aufgerufen wird.  
Blatt = Hintergrundinformation  
Liste = Handlungsanweisungen



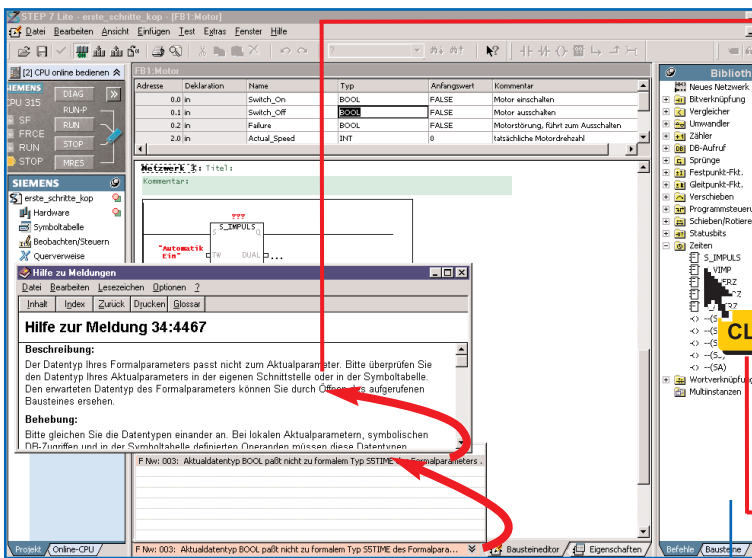
# Starten und Bedienen



## Noch mehr Hilfe

Hilfe zu Menübefehlen

Drücken Sie **SHIFT + F1**. Klicken Sie in einem Pulldown-Menü auf einen Menübefehl. Die Hilfe zum Menübefehl wird angezeigt.



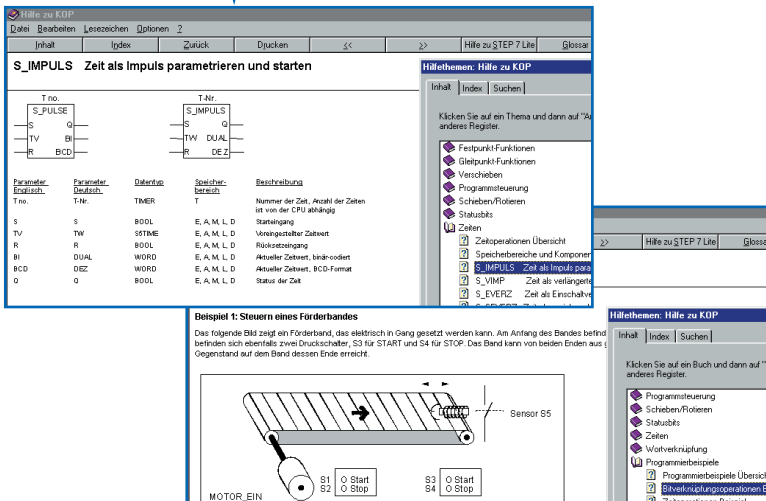
Hilfe zu Fehlermeldungen

Zusätzlich können Sie auch Hilfe zu Fehlermeldungen aufrufen, die in der Statuszeile angezeigt werden. Klicken Sie auf die Fehlermeldung und rufen Sie über die rechte Maustaste **Hilfe** auf.

Referenzhilfe

Drücken Sie **SHIFT + F1** und klicken Sie dann z. B. auf **S\_IMPULS**. Die Referenzhilfe wird in einem weiteren Fenster geöffnet.

Die Referenzhilfe wird geöffnet



In der Referenzhilfe erhalten Sie Hilfe zum selektierten Befehl oder Baustein.

Sie können von hier aus z. B. auch Programmierbeispiele aufrufen.

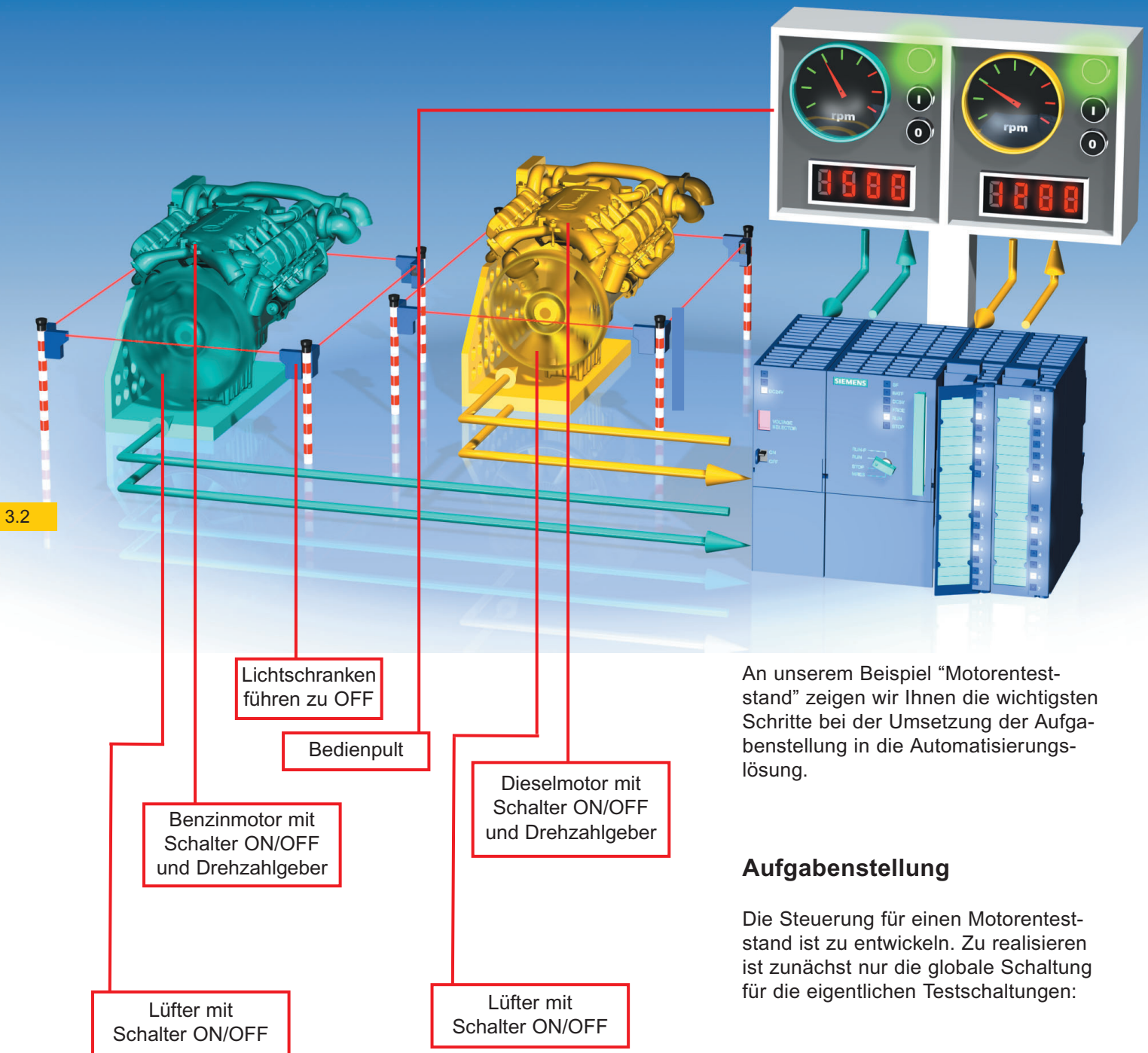
Haben Sie sich mit der Bedienung von STEP 7 Lite vertraut gemacht, schließen Sie die Kopie des Beispielprojektes.

# 3

## Umsetzen der Aufgaben- stellung



## Aufgabenstellung – Motoren- teststand



2. Eine Lichtschranke um jeden Motor sichert den Gefahrenbereich ab und führt unabhängig vom programmierten Beispiel zu NOT AUS.
3. Mit dem Ein- und Ausschalten des Motors wird gleichzeitig ein elektrisch angetriebener **Lüfter** ein- bzw. ausgeschaltet.
4. Nach dem Ausschalten soll der Lüfter noch 4 Sekunden nachlaufen.
5. Der Bediener soll ein Signal erhalten, sobald die Motoren Ihre Soll-drehzahlen erreicht haben:

**Benzinmotor = 1.500 U/min**  
**Dieselmotor = 1.200 U/min**

## Lösung

Schon vorab die Lösung: In den Beispielprogrammen finden Sie im **OB 1** das Signal "Solldrehzahl erreicht" realisiert im

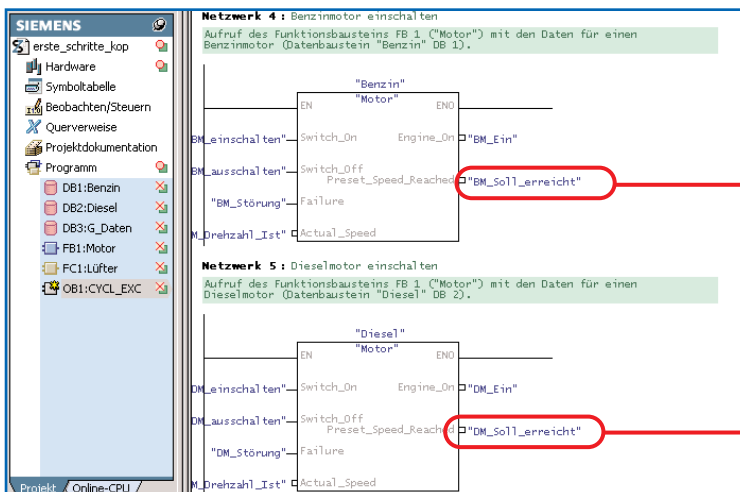
- **Netzwerk 4** für den Benzinmotor. und im
- **Netzwerk 5** für den Dieselmotor.

Mehr hierzu, auch im Kapitel 7, Abschnitt "Bausteinanruf programmieren".

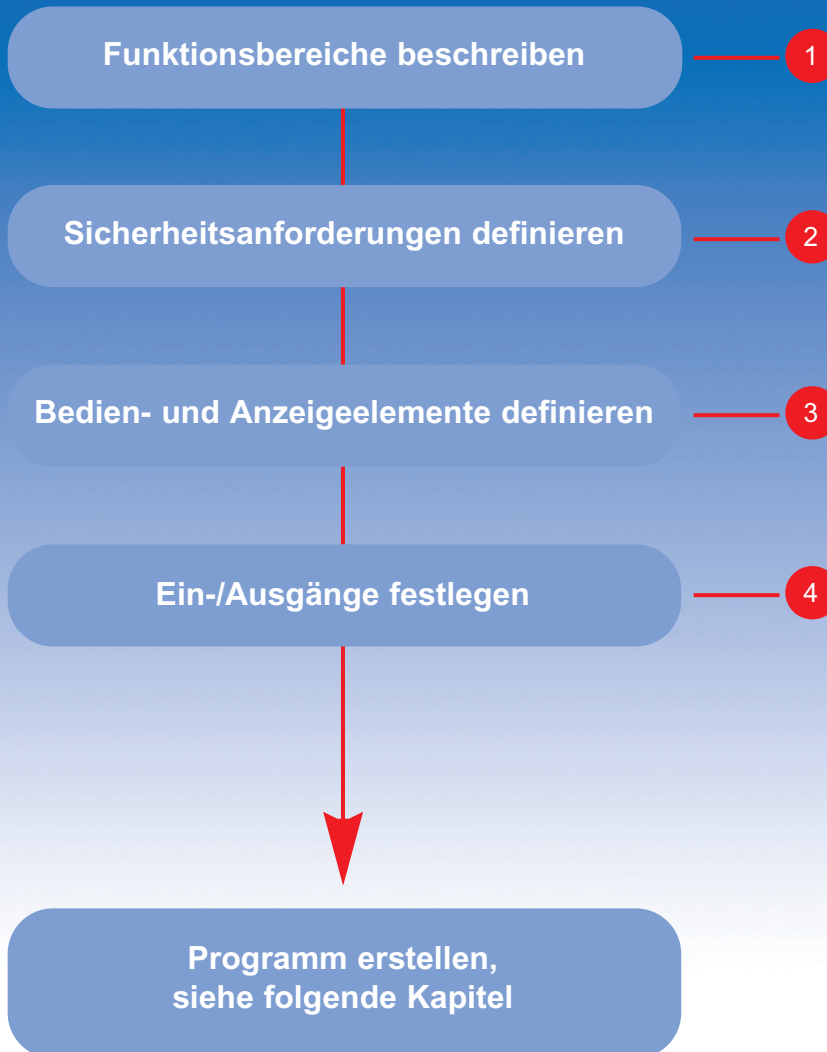
Über das Signal "Solldrehzahl erreicht" könnte nun ein Testprozess angestoßen werden, z.B.:

- Start einer Abgas-Vergleichsmessung oder
- Start einer Drehzahl-Standfestigkeitsmessung.

Dies ist aber nicht Bestandteil unseres Beispielprogramms.



# Zerlegen Sie den Prozess



Bevor Sie mit der Programmierung beginnen, zerlegen Sie den Prozess.

Die grundsätzliche Vorgehensweise die Sie für jedes beliebige Projekt verwenden können, ist oben dargestellt.

Jeder Schritt lässt sich weiter unterteilen. Je höher die Granularität der Prozesszerlegung desto einfacher ist die Strukturierung des Anwenderprogramms.

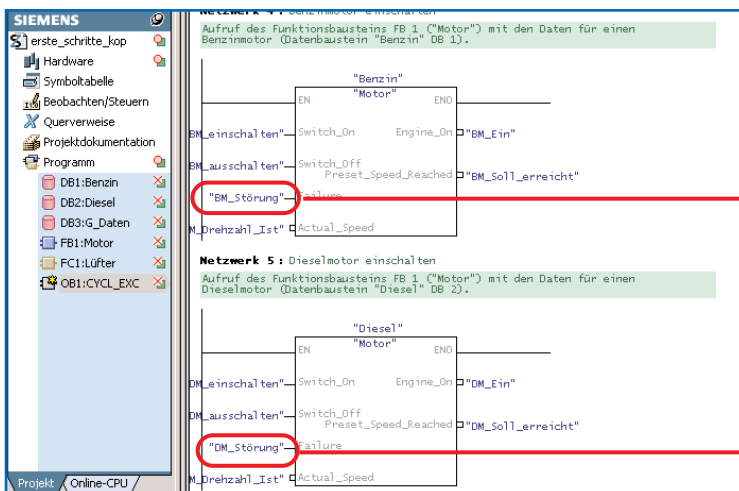


Funktionsbereich	Zugehörige Geräte
Funktionsbereich A	= Benzinmotor = Drehzahlgeber = Lüfter
Funktionsbereich B	= Dieselmotor = Dehzahlgeber = Lüfter

1

Funktionsbereiche beschreiben:

- Unterteilen Sie den Prozess in zusammengehörige Gruppen.
- Definieren Sie die Elemente, die diesen Bereich steuern.
- Definieren Sie elektrische, mechanische und logische Ein- und Ausgänge für jede Aufgabe.
- Definieren Sie Verriegelungen und Abhängigkeiten zwischen den Aufgaben.



2

Sicherheitsanforderungen definieren:

Das ist in unserem Beispiel die Sicherheitsabschaltung, die in der Realität wesentlich aufwendiger programmiert werden muss.

3.5

3

Bedien- und Anzeigeelemente definieren:

Jeder Prozess benötigt ein Bedien- und Beobachtungssystem, das menschliches Eingreifen in den Prozess ermöglicht.

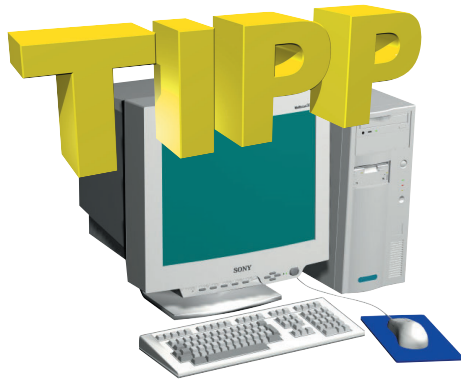
Status	Symbol	Adresse	Datentyp
	Automatik Ein	E 0.5	BOOL
	Automatikbetrieb	A 4.2	BOOL
	Benzin	DB 1	FB 1
	BM_ausschalten	E 1.1	BOOL
	BM_Drehzahl_Ist	MW 2	INT
	BM_Ein	A 5.0	BOOL
	BM_einschalten	E 1.0	BOOL
	BM_Lüfter_ein	A 5.2	BOOL
	BM_Nachlauf	T 1	TIMER
	BM_Soll_erreicht	A 5.1	BOOL
	BM_Störung	E 1.2	BOOL
	CYCL_EXC	OB 1	OB 1
	Diesel	DB 2	FB 1

4

Ein- und Ausgänge festlegen:

Selbst für unser kleines Beispiel benötigen Sie für den Benzinmotor **BM** drei physikalische **Ein-** bzw. **Ausgänge**.

Einen guten Überblick über alle Ein- und Ausgänge liefert Ihnen die Symboltabelle in Kapitel 5.

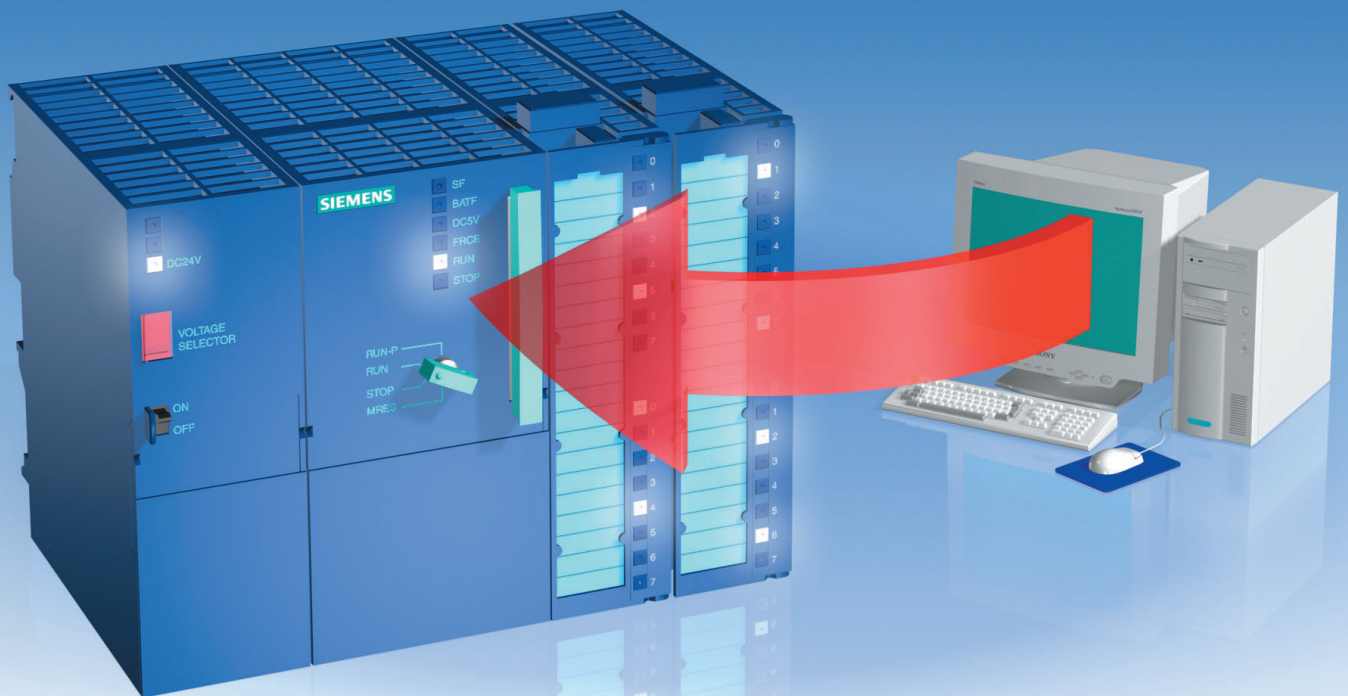


Einsteigern empfehlen wir unsere SIEMENS-Trainingskurse, in denen Sie anhand von Beispielen lernen, wie Sie Prozesse mit SIMATIC automatisieren können.

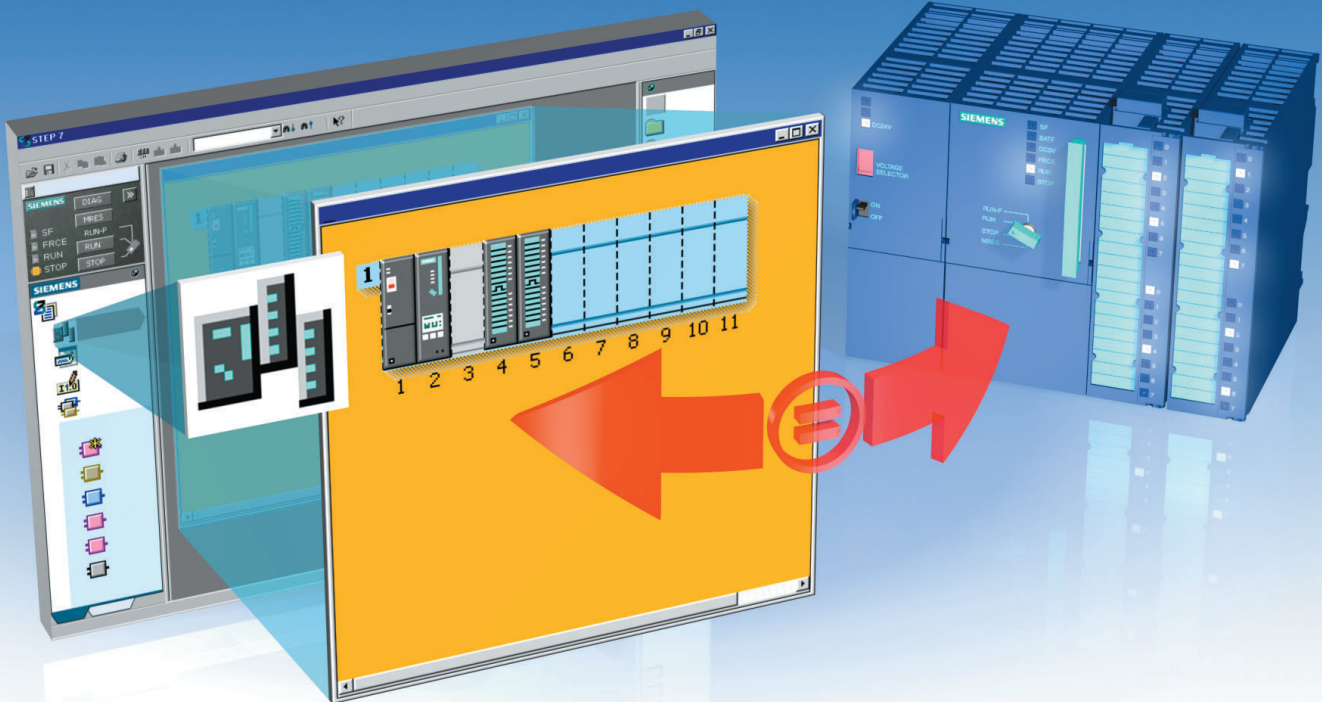


# 4

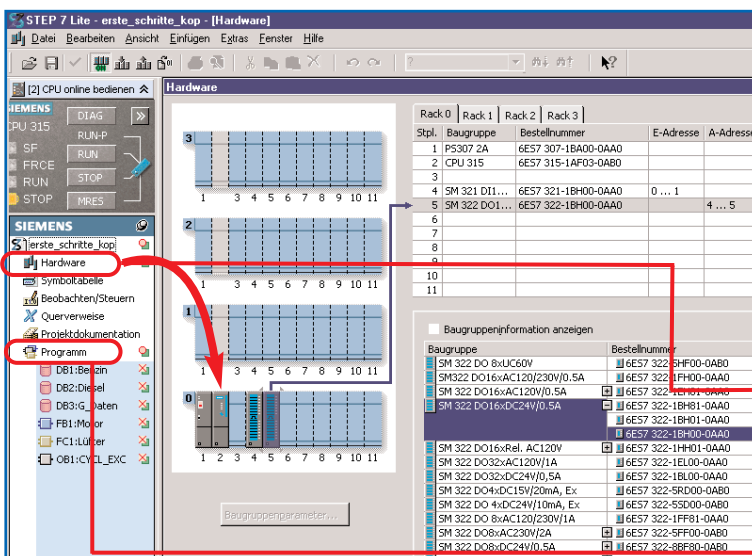
## Konfigurieren der Baugruppen



## Was passiert beim Konfigurieren?



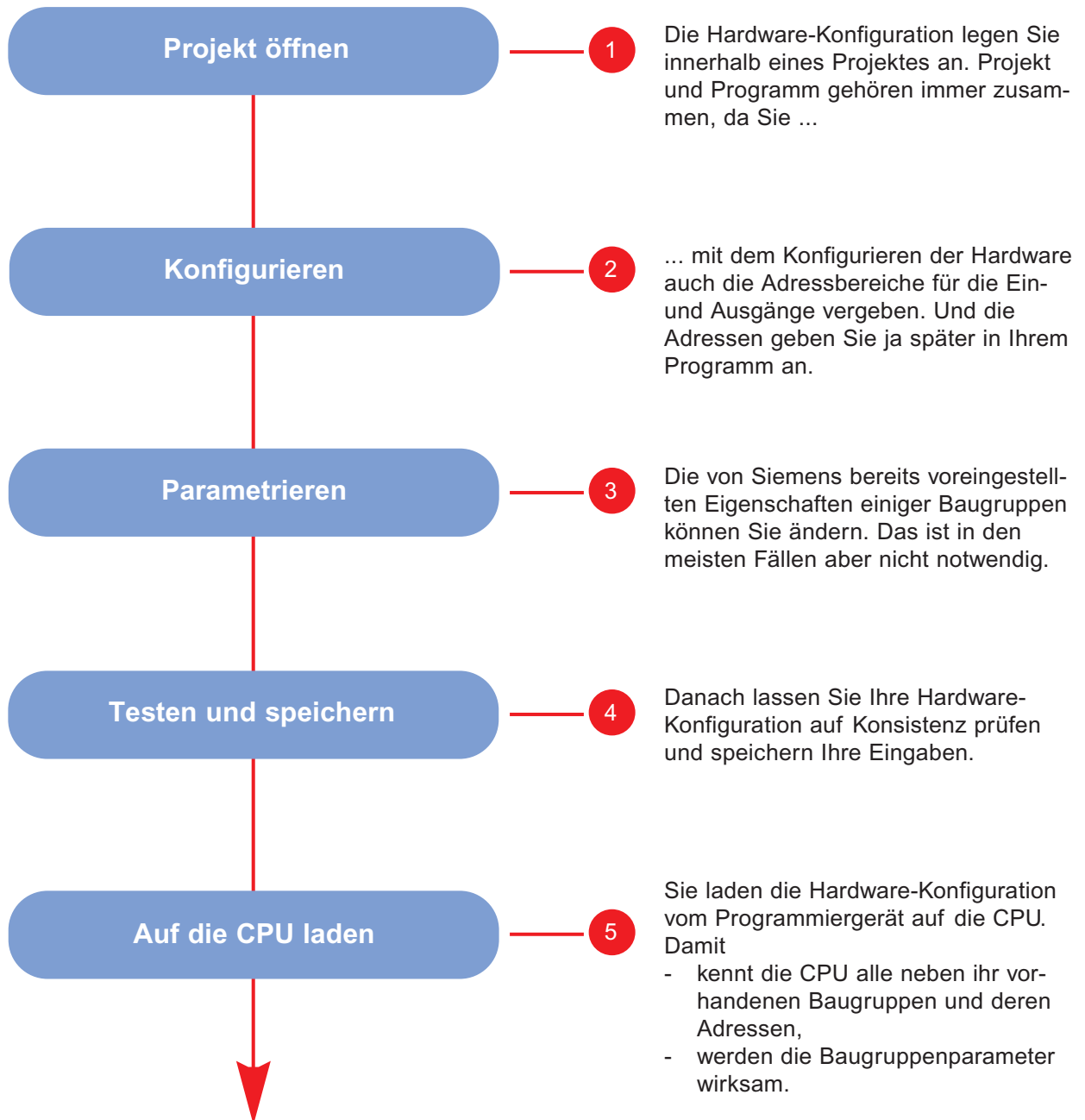
Beim Konfigurieren der Hardware bauen Sie die SPS-Station auf Ihrem Programmiergerät nach. Sie können diese Konfiguration in andere STEP 7 Lite-Projekte kopieren, ggf. modifizieren und in andere existierende Stationen laden. Beim Anlauf des Automatisierungssystems vergleicht die CPU die mit STEP 7 Lite erstellte Sollkonfiguration mit der tatsächlichen Istkonfiguration der Anlage. Eventuelle Fehler können so sofort erkannt und gemeldet werden.



1 Geben Sie im Projektfenster im Element Hardware in einem grafisch dargestellten Rack alle Baugruppen an, die in Ihrer SPS-Station integriert sind.

2 Genau für diese Hardware-Konfiguration editieren Sie später im Element Programm Ihr Anwenderprogramm.

## Hardware-Konfiguration im Überblick



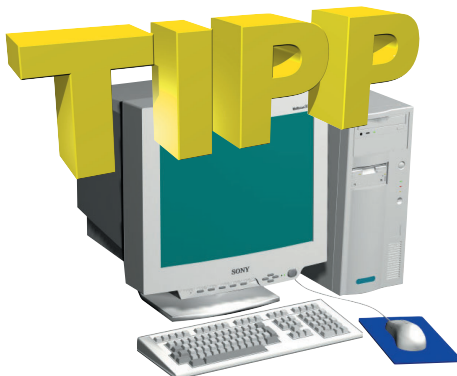
# Neues Projekt anlegen



4.4

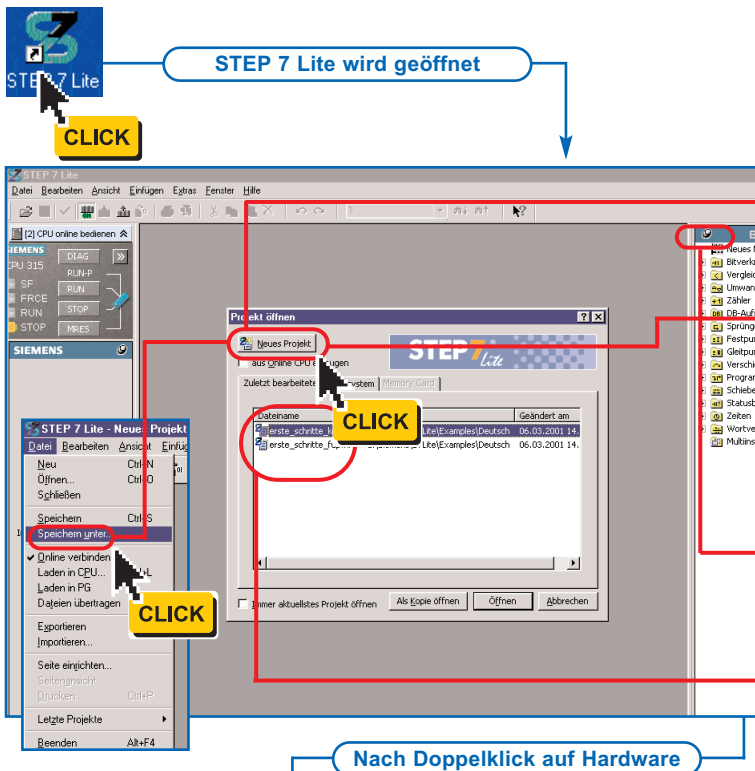
Ihr Projekt "Getting Started" muss die bei Ihnen vorhandene Hardware-Konfiguration enthalten, nicht die in unseren Beispielen.

1 In diesem Kapitel legen Sie ein neues Projekt mit dem Titel "Getting Started" an. Es wird hier als oberster Eintrag angezeigt.  
In den folgenden Kapiteln werden Sie dieses Projekt immer weiter ausbauen.



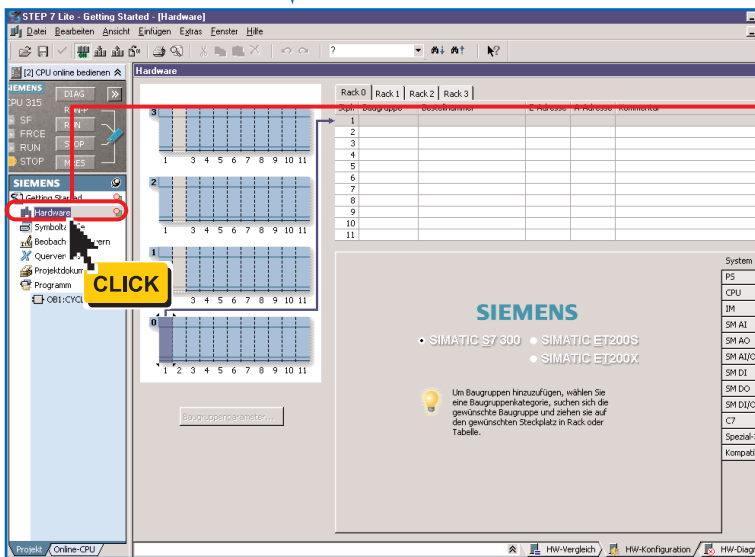
Bei der Hardware-Konfiguration können Sie sich an den mitinstallierten Beispielprojekten orientieren. Die Beispielprojekte finden Sie unter:  
Laufwerk:\Siemens\S7Lite\Examples\Deutsch...

Die Hardware-Konfiguration ist in allen drei Beispielprojekten gleich.



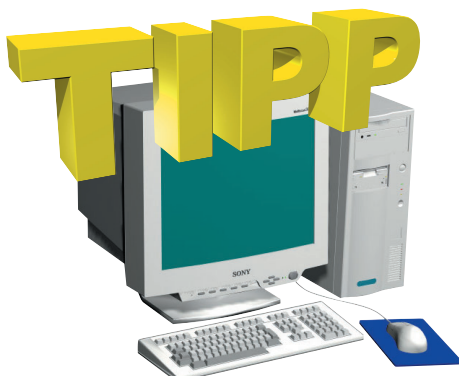
## Neues Projekt anlegen

- 1 Starten Sie STEP 7 Lite durch Doppelklick.
- 2 Öffnen Sie ein neues Projekt.
- 3 Speichern Sie das neue Projekt unter dem Namen  
"Getting Started"
- 4 Klicken Sie auf die Pinnadel und blenden Sie hierdurch die Bibliotheken aus.
- 5 Wollen Sie sich an einer unserer Konfigurationen orientieren, öffnen Sie in einer zweiten Instanz von STEP 7 Lite zusätzlich eines der mitgelieferten Beispielprojekte.



- 6 Doppelklicken Sie auf Hardware. Das Fenster für die Hardware-Konfiguration wird im Arbeitsbereich geöffnet.

4.5



Hardware-Konfigurationen lassen sich von einem in ein anderes Projekt kopieren. Wollen Sie z. B. sofort mit Kapitel 5 weitermachen, öffnen Sie eines der mitgelieferten Beispielprojekte und kopieren das Element **Hardware** in Ihr Projekt "Getting Started" (siehe Schritt 5).



## In der Sicht Hardware-Konfiguration arbeiten

The screenshot shows the 'Hardware' configuration window in STEP 7 Lite. It includes a left sidebar with project navigation, a central rack diagram, a hardware catalog, and a configuration table.

**Grafische Sicht der Hardware-Konfiguration**: Points to the rack diagram showing slots 1-11 for Rack 0, 1, 2, and 3.

**Konfigurationstabelle**: Points to the table with columns: Stpl., Baugruppe, Bestellnummer, E-Adresse, A-Adresse, and Kommentar.

**Hardwarekatalog**: Points to the catalog listing SIMATIC S7 300, ET200S, and ET200X modules.

**System**: Points to the system selection list on the right, including PS, CPU, IM, SM AI, SM AO, SM AI/O, SM DI, SM DO, SM DI/O, C7, Spezial-300, and Kompatibel.

**Element Hardware**: Points to the 'Hardware' button in the left sidebar.

**Parametrierung von Baugruppen**: Points to the 'Baugruppenparameter...' button.

**Wahl des SIMATIC Systems**: Points to the 'SIMATIC S7 300', 'SIMATIC ET200S', and 'SIMATIC ET200X' options.

**Inkonsistenz-Piktogramme**: Points to the 'Inkonsistenz-Piktogramme' button in the bottom right.

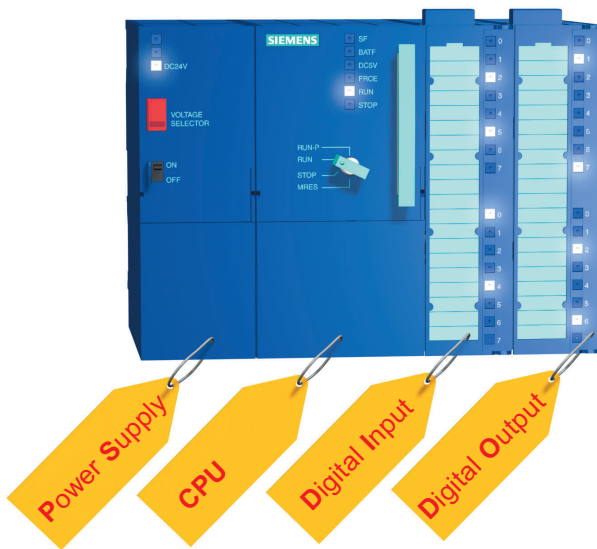
**Unterschiede zwischen SPS-Station und CPU anzeigen (nur wenn Online-Verbindung zur CPU besteht)**: Points to the 'Unterschiede zwischen SPS-Station und CPU anzeigen' button.

**Technische Probleme an den Baugruppen anzeigen (nur wenn Online-Verbindung zur CPU besteht)**: Points to the 'Technische Probleme an den Baugruppen anzeigen' button.

### Überblick

Haben Sie im Projektfenster das Element Hardware durch Doppelklick geöffnet, wird im Arbeitsbereich rechts daneben die Sicht "Hardware" angezeigt.

Hier wählen Sie im Hardware-Katalog die Baugruppen aus, die in Ihrer SPS-Station vorhanden sind.



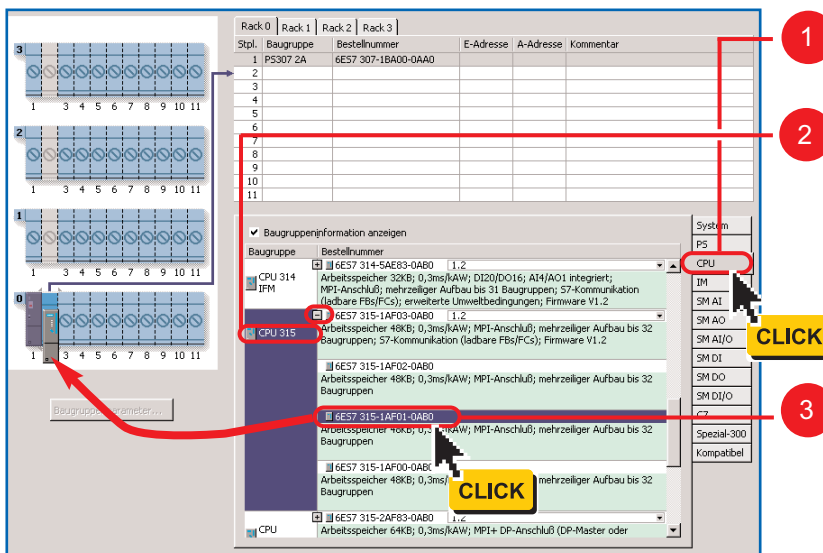
## Hardware konfigurieren

In den Beispielprogrammen sind die folgenden Baugruppen hinterlegt:

- 1 **Power Supply** = Stromversorgung
- 2 **CPU** = SPS-Baugruppe
- 3 **Digital Input** = Digitaleingabebaugruppe
- 4 **Digital Output** = Digitalausgabebaugruppe

Die Bestellnummern sind auf der Baugruppenvorderseite aufgedruckt.

Konfigurieren Sie Ihre Baugruppen wie nachfolgend beschrieben.



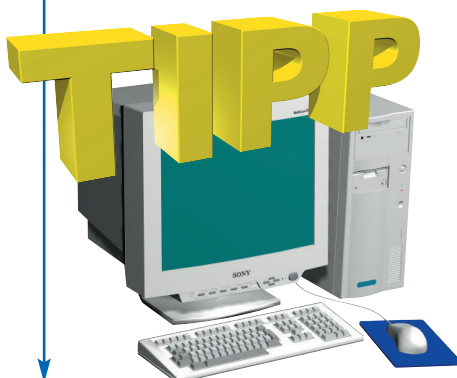
1 Klicken Sie auf **CPU**.

2 Navigieren Sie bis zur **CPU 315** und klicken Sie auf das Icon "+". Alle älteren Ausgabestände der CPU 315 werden angezeigt.

3 Ziehen Sie Ihre CPU via Drag&Drop auf das Rack. Gemäß den Steckplatzregeln sind nicht zulässige Plätze durch ein Verbotssymbol gesperrt.

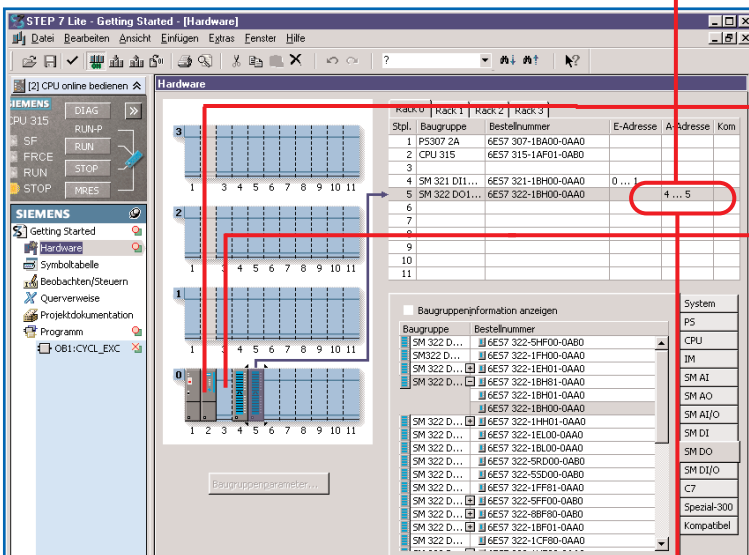
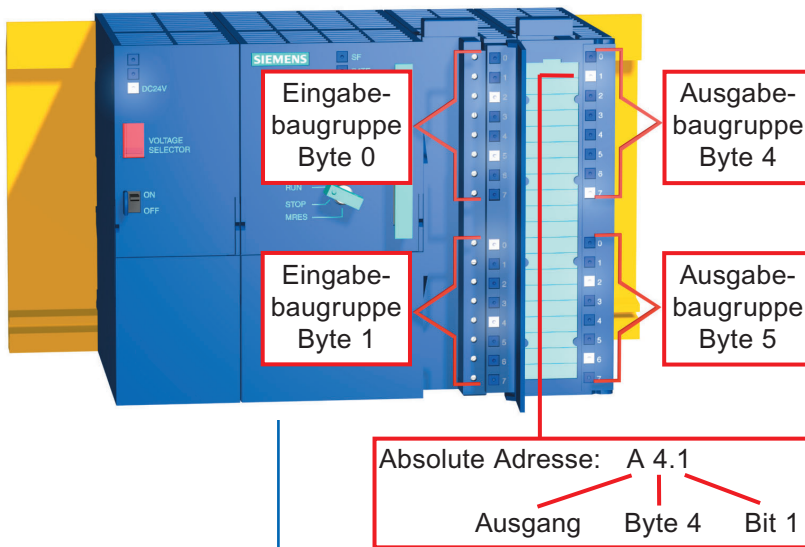
Verfahren Sie so für alle Baugruppen.

Ein weiteres Fenster öffnet sich



Wollen Sie die Hardware-Konfiguration später auf Ihre CPU laden, müssen Sie natürlich die bei Ihnen tatsächlich vorhandene Hardware konfigurieren und nicht die unseres Beispielprojektes.

# Konfigurieren der Baugruppen



## Ergebnis der Konfiguration

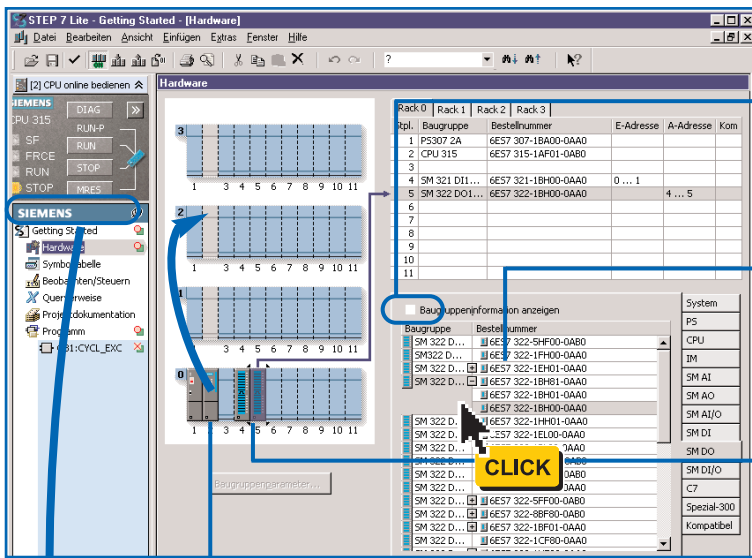
4 Sie stecken von links nach rechts: Stromversorgung, CPU, Eingabebaugruppe, Ausgabebaugruppe.

5 Baugruppen müssen lückenlos gesteckt sein, da sonst die Baugruppen nach der Lücke nicht über den Rückwandbus mit Strom versorgt werden.

Ausnahme in STEP 7 Lite: Steckplatz 3 ist reserviert für die Anschaltungsbaugruppe (IM), mit der Sie die darüber stehenden Racks anbinden können. Stecken nur im untersten Rack Baugruppen, so darf hier in der Konfiguration eine Lücke auftreten.

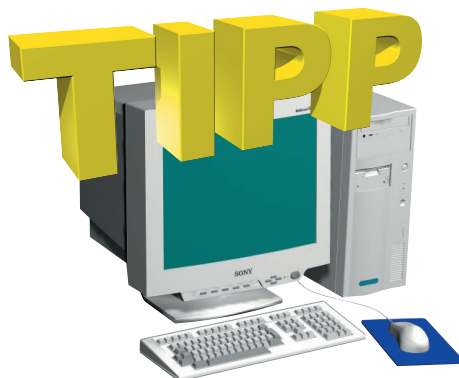
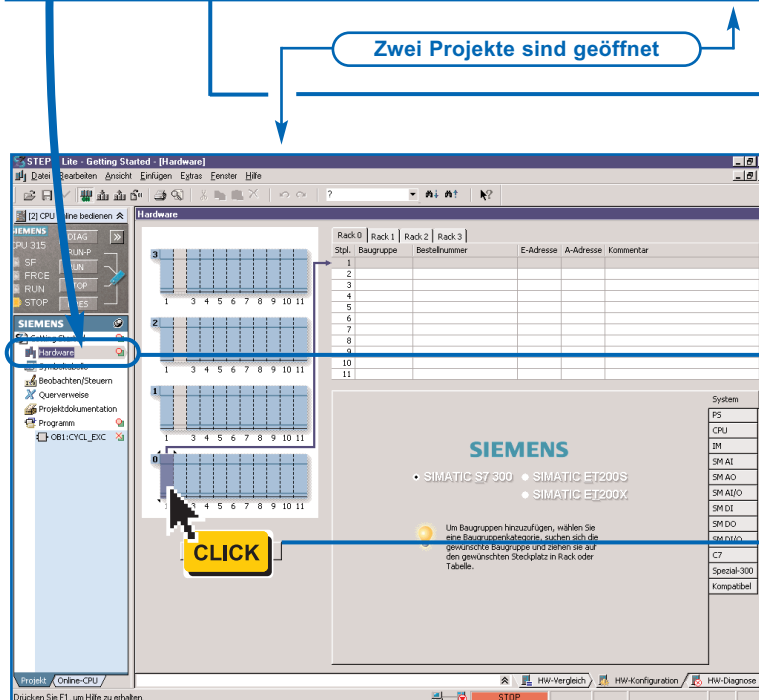
6 In der **Konfigurationstabelle** wurden in den Spalten **E-** und **A-Adresse** automatisch die Bytes für die Adressangabe gesetzt. Sie sind wesentlicher Bestandteil der Adressenangabe für die Programmerstellung.



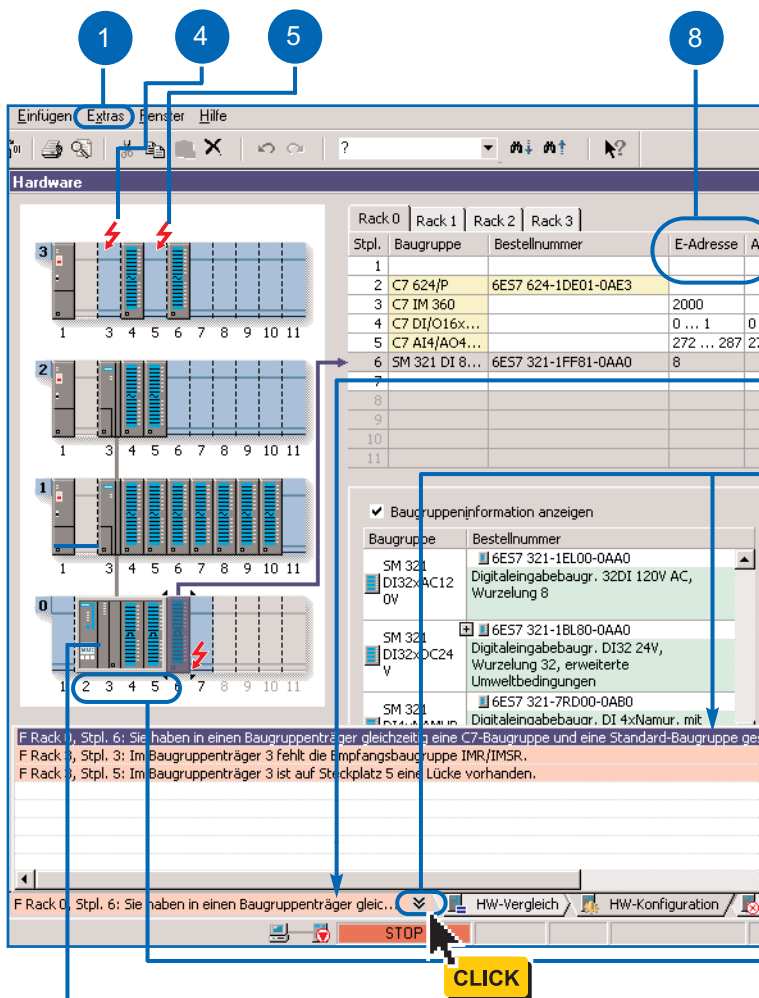


## Tipps – Baugruppen stecken

- 1 Blenden Sie die **Baugruppeninformation** aus, wenn Sie mehr Überblick benötigen, z. B. wenn Sie gezielt nach Bestellnummern suchen.
- 2 Stecken Sie Baugruppen z. B. mit Rechtsklick auf die Baugruppe und **Baugruppe stecken** im Kontextmenü.
- 3 Zwischen zwei Baugruppen können Sie weitere einfügen. Die Baugruppen werden nach rechts auf die nebenliegenden Steckplätze verschoben.
- 4 Innerhalb des Racks können Sie eine oder mehrere Baugruppen mit der Umschalttaste markieren und per Drag&Drop verschieben oder kopieren.
- 5 Öffnen Sie in einer zweiten Instanz von STEP 7 Lite ein weiteres Projekt und kopieren Sie z. B. die gesamte Hardware-Konfiguration per Drag&Drop von einem Projekt in ein anderes.
- 6 Löschen Sie eine Baugruppe z. B. über das Kontextmenü nach Rechtsklick.



Probieren Sie ruhig alle Funktionen aus, die Sie auch in anderen Windows-Anwendungen verwenden. In STEP 7 Lite haben wir viele Windows-Funktionen, wie Kontextmenüs, Drag&Drop, Arbeiten mit Tastenkombinationen usw. realisiert.



## Beispiel einer Maximal-konfiguration

Zur Demonstration haben wir eine große Konfiguration angelegt und ein paar Fehler eingebaut.

1 Zum Fehlersuchen rufen Sie auf: **Extras > Konsistenz prüfen.**

2 Ein eventuell vorhandener Konfigurationsfehler wird hier angezeigt.

3 Sind mehrere Fehler vorhanden, werden sie angezeigt, wenn Sie mit der linken Maustaste auf das Erweiterungssymbol klicken.

Vorhandene Fehler:

4 In Rack 3 fehlt hier eine Anschaltungsbaugruppe (IM). Daher besteht keine Verbindung zum Rack 3. In Rack 1 und 2 ist diese Anschaltungsbaugruppe vorhanden.

5 Freie Steckplätze sind nicht erlaubt.

6 Das ist ein C7-Komplettsystem (in der Konfigurationstabelle gelb hinterlegt). Die Baugruppe rechts daneben passt nicht dazu.

Haben Sie alle Fehler behoben, prüfen Sie die Konsistenz erneut.

Und sonst:

7 In STEP 7 Lite ist pro Projekt nur eine CPU zulässig. Sie befindet sich immer in Rack 0. Die Steckplätze darüber bleiben frei.

8 Beachten Sie, wie in der Konfigurationstabelle die Adressbereiche hochgezählt werden.



#### Online-Hilfe: F1

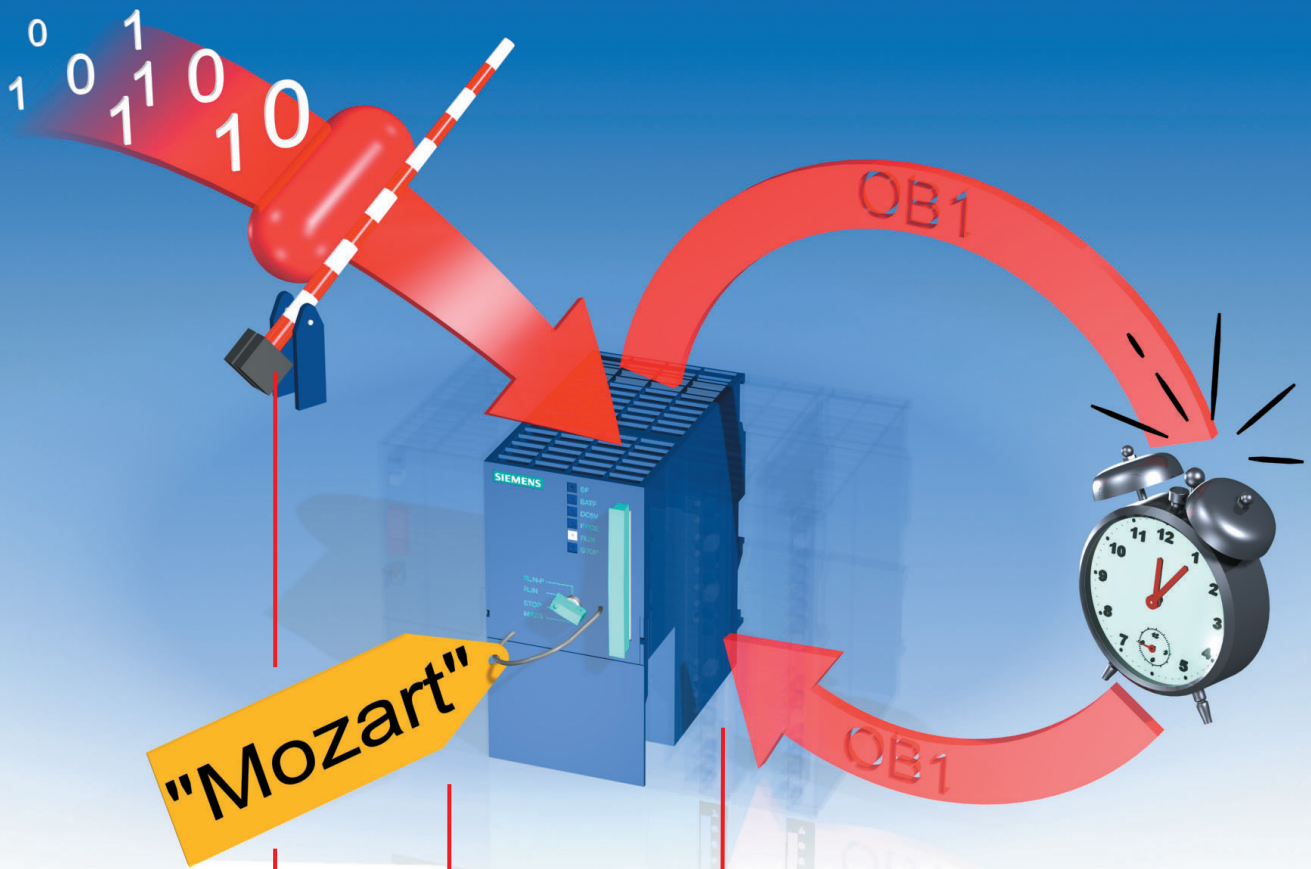
- In der Hilfe zu STEP 7 Lite finden Sie unter **Inhalt > Konfigurieren der Hardware > Konfigurieren der Baugruppen** die globalen Regeln zum Konfigurieren.
- Unter **Index > Steckplatzregeln** finden Sie die wichtigsten Regeln zum Stecken selbst.

#### Navigieren im Hardware-Katalog mit der MLFB

Die MLFB-Nummer ist die Siemens-Bestellnummer.

Kennen Sie die MLFB-Nummer der Baugruppe, die Sie im HW-Katalog suchen, können Sie bei selektiertem HW-Katalog über das Eingabefeld "Zu suchender Text" in der Symbolleiste die MLFB eingeben, **Return** drücken und die Baugruppe wird Ihnen angezeigt.

# Baugruppen parametrieren



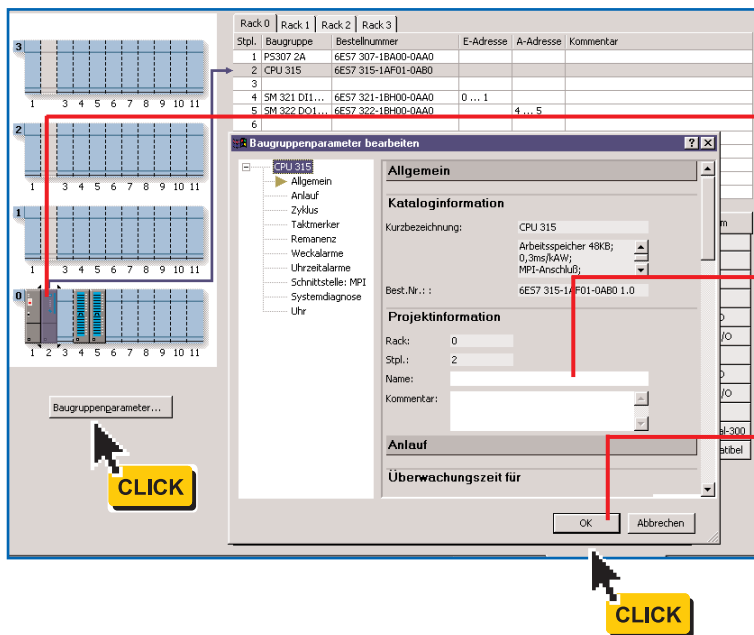
4.12

### Was ist Parametrieren?

Das Betriebsverhalten einiger analoger und digitaler Baugruppen und das der CPU lässt sich einstellen: wir nennen das "Parametrieren".

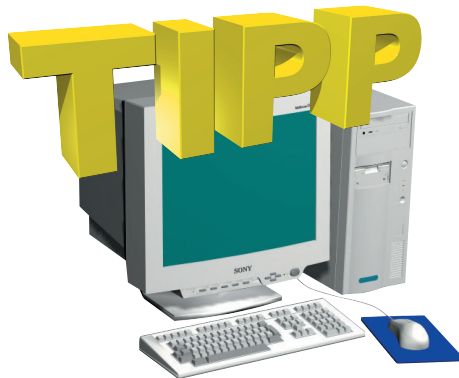
Parametrier-Beispiele für eine CPU:

- 1 – Sie können den zyklischen Ablauf des Programms in der CPU durch einen Weckalarm unterbrechen.
- 2 – Sie geben der CPU einen Namen. Hier "Mozart".
- 3 – Oder Sie können die CPU mit einem Passwort gegen Schreib-/Lesezugriff über MPI schützen.



## Parametrieren Sie die CPU 315

- 1 Markieren Sie die CPU 315. Klicken Sie auf die Schaltfläche **Baugruppenparameter**.
- 2 Geben Sie im Fenster **Baugruppenparameter bearbeiten** im Feld **Namen** den Namen "Mozart" ein.
- 3 Übernehmen Sie Ihre Eingaben mit **OK**. Das Fenster wird geschlossen.



Alle Parameter wurden von uns grundeingestellt und passen für nahezu alle Standardfunktionen.

4.13

Stimmt irgend etwas nach einer Änderung nicht mehr – nicht so schlimm – im Hardwarekatalog sind die Grundeinstellungen für alle Baugruppen noch vorhanden.

### Online-Hilfe: F1

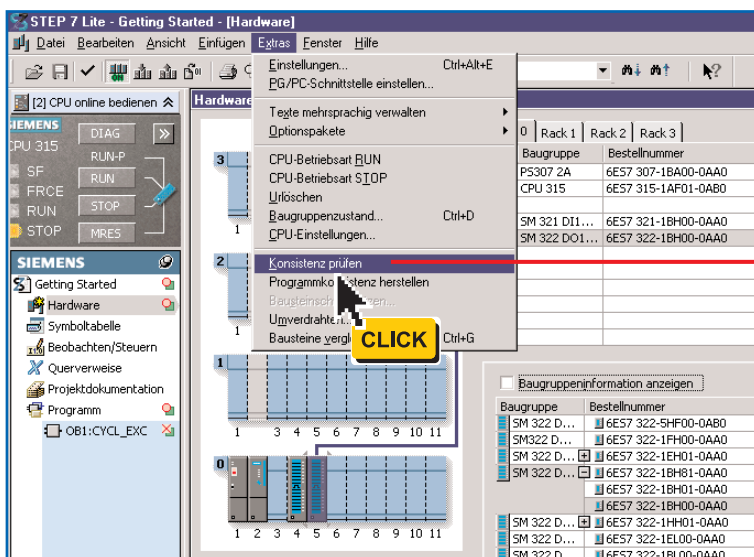
Parameter für CPUs stehen häufig im Zusammenhang mit Organisationsbausteinen.

Im **Index** finden Sie daher z. B. unter **Weckalarm** die Erklärung für die **Organisationsbausteine für den Weckalarm (OB 30 bis OB 38)**.

## Konfigurationsdaten speichern



4.14



### Konfigurationsdaten prüfen

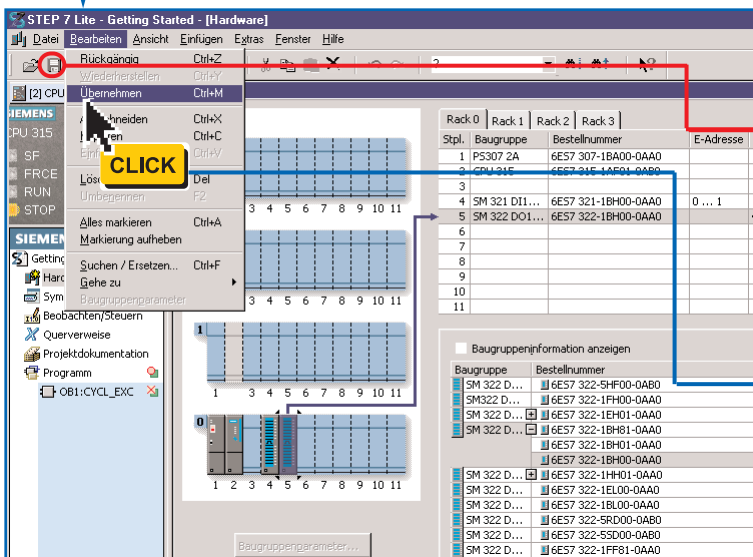
Bevor Sie Ihre Konfiguration speichern, sollten Sie grundsätzlich prüfen, ob Sie bei der Eingabe keine Fehler gemacht haben.

Rufen Sie den Menübefehl **Extras > Konsistenz prüfen** auf. Es wird geprüft, ob aus Ihren Eingaben Konfigurationsdaten erzeugt werden können.

Bestätigen Sie die Meldung "Die Konfiguration ist fehlerfrei!" mit **OK**.

Ohne Fehler geht es weiter





## Konfigurationsdaten speichern

Wählen Sie im Menü **Datei > Speichern** oder klicken Sie auf das Diskettensymbol in der Symbolleiste.

Nicht nur die Hardware-Konfiguration, sondern alle Elemente des Projekts werden gespeichert.

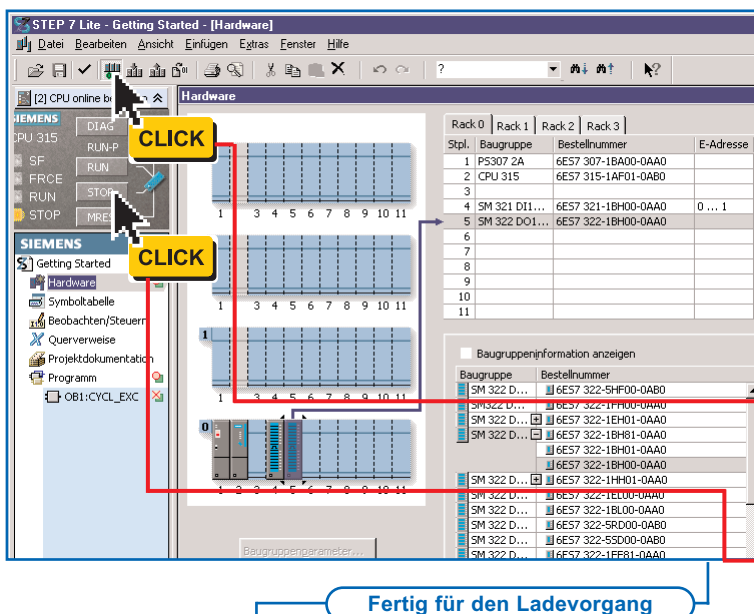
Wählen Sie den Menübefehl **Bearbeiten > Übernehmen**, werden Ihre Konfigurationsdaten (immer der Inhalt des aktiven Fensters) in einer temporären Datei zwischengespeichert. Diese Art, Daten zu sichern, ist immer dann zu empfehlen, wenn Sie Änderungen eventuell wieder rückgängig machen wollen. Schließen Sie Ihr Projekt nachdem Sie immer nur Daten übernommen haben, werden Sie gefragt, ob Sie Ihre Änderungen speichern möchten.



## Hardware-Konfiguration in die CPU laden



4.16



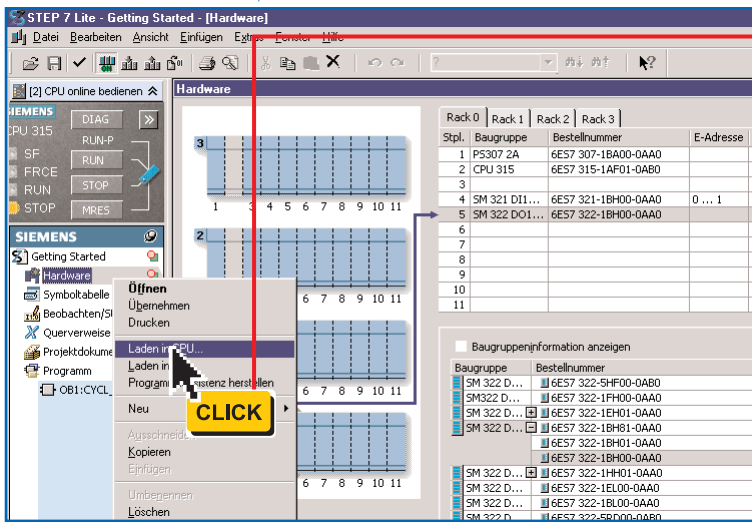
### Vorbereiten und Laden

Mit dem Laden übertragen Sie alle Konfigurationsdaten in die CPU. Zuvor müssen Sie jedoch CPU und Programmiergerät "Online verbinden". Das ist im Kapitel 10 ausführlich beschrieben.

Das Wichtigste zum "Online verbinden":

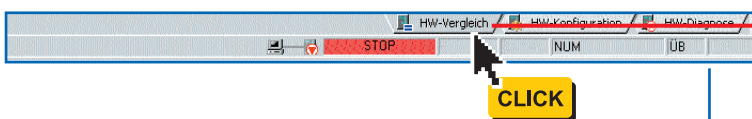
1 Nach dem Anschließen der Kabel und dem Umräumen der CPU klicken Sie auf **Online verbinden**.

2 Stellen Sie im CPU-Bedienpanel die CPU in den Betriebszustand **STOP**. In der Fußzeile leuchtet STOP rot.



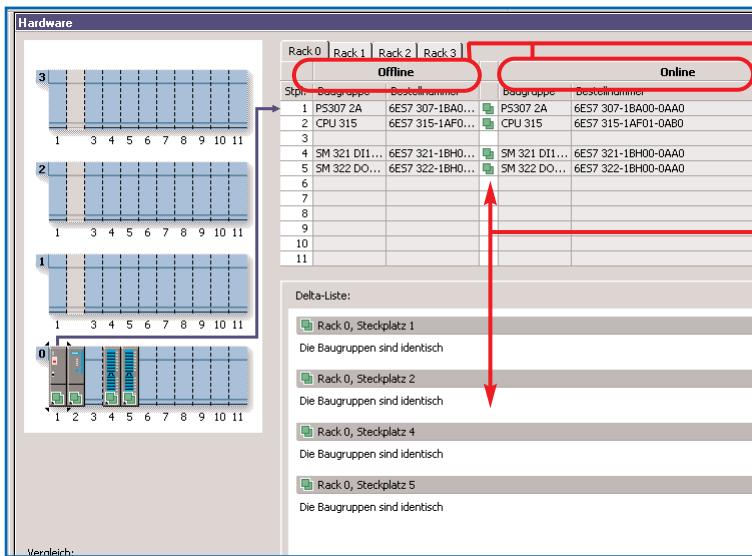
3 Wählen Sie nach Rechtsklick auf **Hardware** die Funktion **Laden in CPU**.

Die Hardware-Konfigurationsdaten werden auf die CPU geladen. Die Baugruppen werden parametrisiert.



4 Klicken Sie auf das Register **HW-Vergleich**.

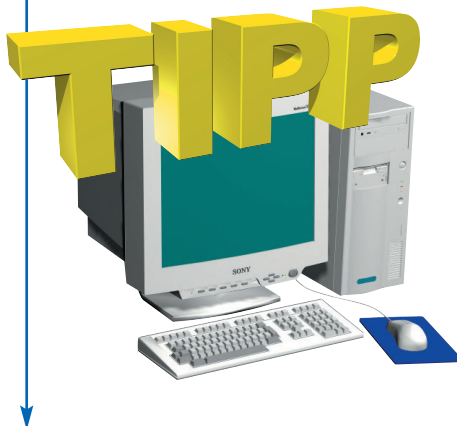
Die Sicht "Hardware vergleichen" wird geöffnet



5 Hier sehen Sie, dass die Konfigurationsdaten auf Ihrem Programmiergerät (Offline) mit den auf die CPU geladenen Daten (Online) nun übereinstimmen.

6 Sollten die Daten nicht übereinstimmen, weisen Piktogramme auf Inkonsistenzen hin. Die Delta-Liste spezifiziert die Inkonsistenzen näher.

Mehr Details zum Bildschirm



Mit dem Befehl **Laden in PG** können Sie eine Hardware-Konfiguration von der CPU in das Programmiergerät zurückladen. Das ist der typische Servicefall, wenn Sie sich mit dem Programmiergerät an einem Schaltschrank befinden und einen Fehler analysieren wollen.

# Konfigurieren der Baugruppen

Vergleichstabelle  
Gegenüberstellung von Online/Offline/Physik

Differenzpiktogramm baugruppenbezogen

Hardware

Stpl.	Baugruppe	Bestellnummer	Baugruppe	Bestellnummer
1	P5307 5A	6ES7 307-1EA0...	P5307 2A	6ES7 307-1BA00-0AA0
2	CPU 314	6ES7 314-1AE8...	CPU 315	6ES7 315-1AF01-0AB0
3				
4	SM 321 DI1...	6ES7 321-1BH0...	SM 321 DI1...	6ES7 321-1BH00-0AA0
5	SM 322 DO...	6ES7 322-1BH0...	SM 322 DO...	6ES7 322-1BH00-0AA0
6				
7				
8				
9				
10				
11				

Delta-Liste:

Rack 0, Steckplatz 1

Unterschiedliche Baugruppen

Offline: 6ES7 307-1EA00-0AA0

Online: 6ES7 307-1BA00-0AA0

Rack 0, Steckplatz 2

Unterschiedliche Baugruppen

Offline: 6ES7 314-1AE84-0AB0

Online: 6ES7 315-1AF01-0AB0

Rack 0, Steckplatz 4

Die Baugruppen sind identisch

Rack 0, Steckplatz 5

Die Baugruppen sind identisch

Vergleich:

- Offline-Online
- Offline-Physik
- Online-Physik

Alles Aktualisieren (F5)

Projekt Online-CPU

HW-Vergleich HW-Konfiguration HW-Diagnose

STOP

... zeigt den Unterschied zwischen der am PG im Register Projekt eingegebenen Konfiguration (Offline) und der Konfiguration, die in die CPU geladen wurde

... zeigt den Unterschied zwischen der eingegebenen Konfiguration und der tatsächlich gesteckten Hardware

... zeigt den Unterschied zwischen der geladenen Konfiguration und der tatsächlich gesteckten Hardware

## Überblick

- 1 Im Register **HW-Konfiguration** geben Sie wie zuvor beschrieben Ihre Hardware ein.
- 2 Liegen Störungen vor, vergleichen Sie im Register **HW-Vergleich**, ob die Konfigurationsdaten übereinstimmen. Hierzu gleich mehr.
- 3 Gibt es trotzdem noch Probleme, können Baugruppen defekt sein. Das prüfen Sie im Register **HW-Diagnose**. Hierzu mehr in Kapitel "12 Fehlerdiagnose".

## Fehler erkennen:

Sie haben die Konfiguration in die CPU geladen. Register **HW-Vergleich** ist aufgerufen.

Im Projektfenster wird hinter **Hardware** ein **Sammelpiktogramm** angezeigt. Das heißt, eine oder mehrere Baugruppen stimmen nicht überein.

Piktogramme auf den Baugruppen im **Rack** kennzeichnen diese Baugruppen.

---

### Vergleich: Offline - Online

Online: Konfiguration, die auf die CPU geladen wurde.

Offline: Konfiguration auf dem Programmiergerät.

Wenn Sie das Register **HW-Vergleich** anklicken, ist hier per Default die Schaltfläche **Vergleich: Offline-Online** angewählt. In der **Delta-Liste** werden die Differenzen bei Konfiguration und Parametrierung aufgelistet.

4.19

---

### Vergleich: Offline - Physik Vergleich: Online - Physik

Physik: Die Konfiguration, die eine CPU von sich aus erkennt, ohne dass überhaupt eine Konfiguration geladen wurde, wird Physik genannt.

Über die entsprechende Schaltfläche können Online- und Offline-Konfiguration der Physik gegenüber gestellt werden.

## Symbole

Hier die wichtigsten Symbole aus der Hardware-Konfiguration.



Die Baugruppe im Projekt stimmt nicht mit der Baugruppe in der Online-CPU überein.



Die tatsächlich gesteckte Baugruppe entspricht der projektierten Baugruppe, hat aber andere Baugruppenparameter.



Die Baugruppe ist zwar projektiert, aber online nicht vorhanden.



Symbolisiert eine "vermutlich identische Baugruppe". Der Baugruppentyp der tatsächlich gesteckten Baugruppe und der projektierten Baugruppe stimmen überein, ob auch die Bestellnummer übereinstimmt, kann nicht ermittelt werden.



Betriebszustand RUN



Betriebszustand STOP

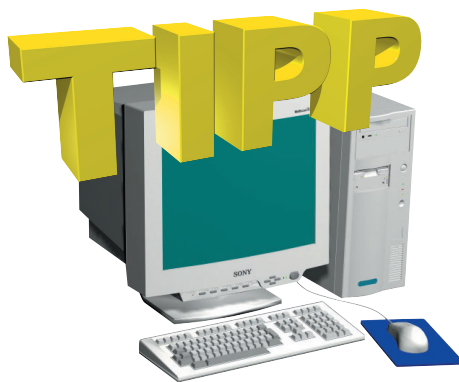


Betriebszustand HALT



Störung

4.20



### Symbole (Icons)

Über die Kurzinfo zu den Symbolen erhalten Sie weitere Hinweise.

Unter **F1 > Index > Symbole** finden Sie Übersichten der Symbole, die im Projektfenster, im Rack und in der Vergleichstabelle angezeigt werden können.

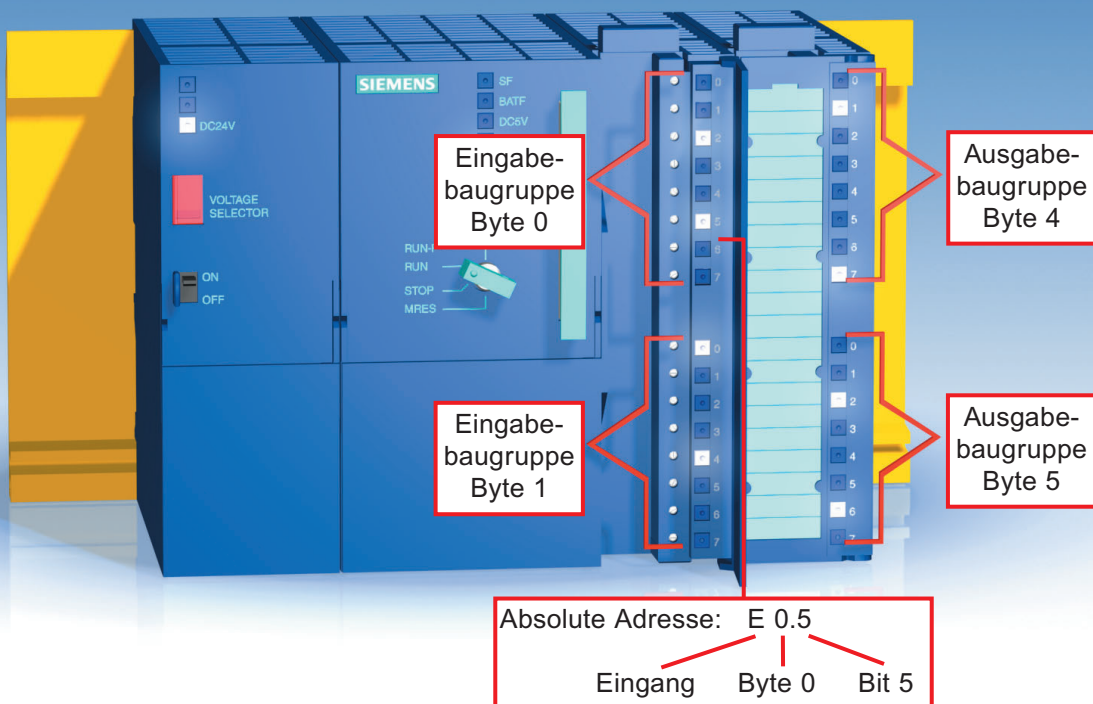
# 5

## Erstellen der Symboltabelle

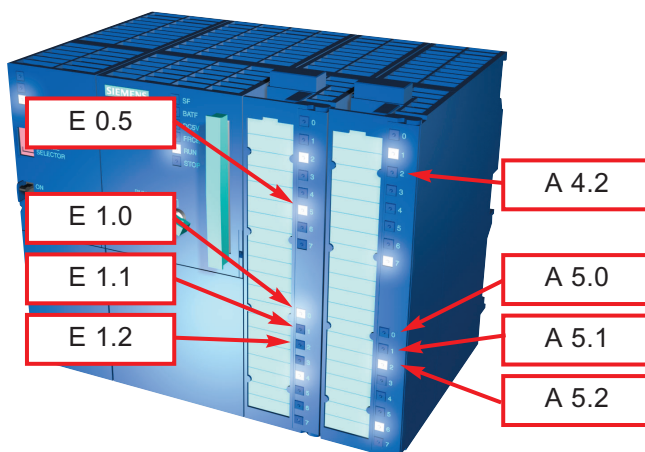




## Absolut programmieren



5.2



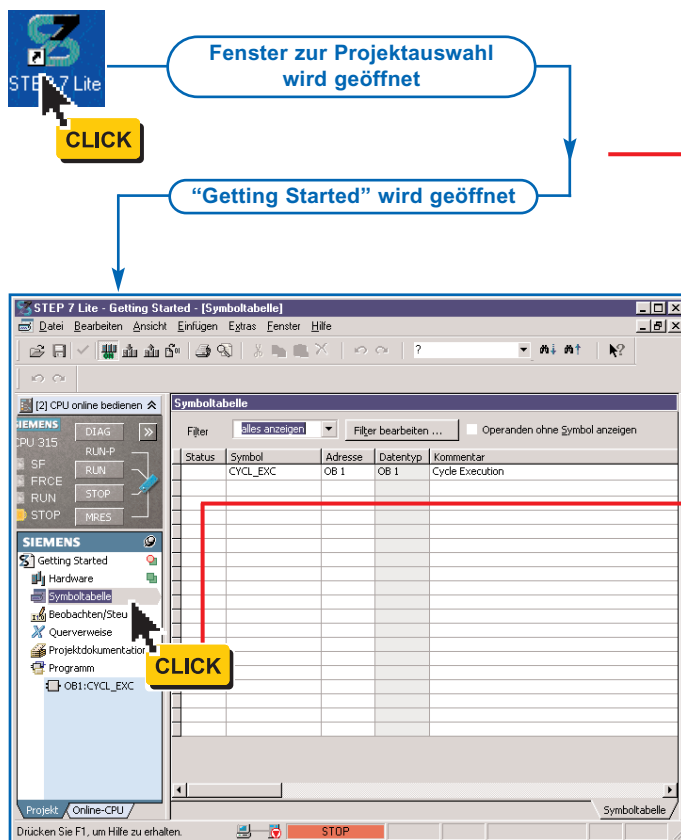
### Adressvergabe

In Kapitel 4 ist beschrieben, wie die absoluten Adressen beim Konfigurieren der Hardware automatisch vergeben werden. Zur Erinnerung:

Jeder Ein- und Ausgang hat durch den Hardwareaufbau eine vorgegebene absolute Adresse.

Die absolute Adresse kann durch frei wählbare Namen (Symbole) ersetzt werden (z. B. E 0.5: Automatik Ein). Die Vergabe der Symbole ist von der Programmiersprache KOP, FUP, AWL unabhängig.





## Symboltabelle und absolute Adressen

1 Öffnen Sie STEP 7 Lite, doppelklicken Sie im Fenster **Projekt öffnen** auf das Projekt “Getting started.k7p”, das Sie in Kapitel 4 angelegt haben.

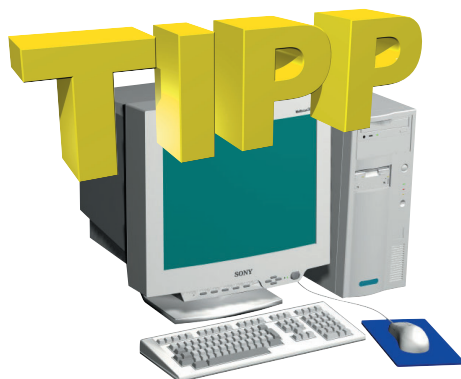
Ihr Projekt besteht momentan nur aus den vordefinierten Projektelementen und aus dem Programmelement **OB 1**.

2 Doppelklicken Sie im Projektfenster auf das Element **Symboltabelle**.

Die Symboltabelle besteht momentan nur aus dem vordefinierten Organisationsbaustein OB 1.

Wollen Sie in Ihrem Programm mit absoluten Adressen arbeiten, müssen Sie keine weiteren Eintragungen machen. Schließen Sie das Fenster einfach wieder.

Im Beispielpunkt wird symbolisch programmiert. Gehen Sie bitte dazu wie auf den folgenden Seiten beschrieben vor.



Die absolute Programmierung sollten Sie nur dann nutzen, wenn Sie in Ihrem STEP 7 Lite-Programm nur wenige Ein- und Ausgänge ansprechen müssen.

## Symbolisch programmieren

Tabelle filtern (z.B. nur Ausgänge anzeigen)

Sortierreihenfolge ändern durch Klick in die Kopfzeile

Ausführliche Kommentierung möglich

Symboltabelle kopieren zwischen Projekten

Symbole

Absolute Adressen

STEP 7 Lite - erste\_schritte\_kop - [Symboltabelle]

Filter: alles anzeigen Filter bearbeiten ... Operanden ohne Symbol anzeigen

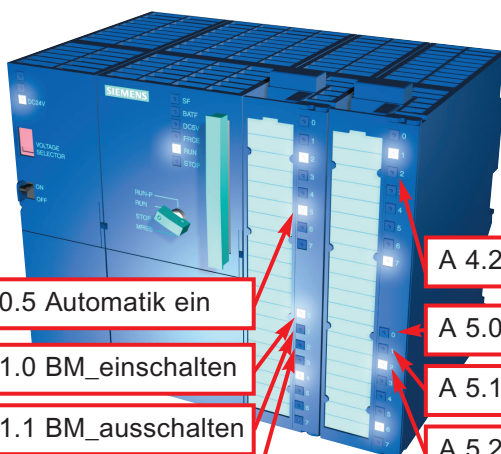
Status	Symbol	Adresse	Datentyp	Kommentar
	Automatik_Ein	E 0.5	BOOL	für die Speicherfunktion (einschalten)
	Automatikbetrieb	A 4.2	BOOL	Speicherfunktion
	Benzin	DB 1	FB 1	Daten für Benzinmotor
	BM_ausschalten	E 1.1	BOOL	Benzinmotor ausschalten
	BM_Drehzahl_Ist	MW 2	INT	tatsächliche Drehzahl für den Benzinmotor
	BM_Ein	A 5.0	BOOL	Kommando für Benzinmotor einschalten
	BM_einschalten	E 1.0	BOOL	Benzinmotor einschalten
	BM_Lüfter_ein	A 5.2	BOOL	Kommando für Benzinmotor-Lüfter einschalten
	BM_Nachlauf	T 1	TIMER	Nachlaufzeit für Benzinmotor-Lüfter
	BM_Soll_erreicht	A 5.1	BOOL	Anzeige "Benzinmotor Sollzahl erreicht"
	BM_Störung	E 1.2	BOOL	Benzinmotor Störung
	CYCL_EXC	OB 1	OB 1	Cycle Execution
	Diesel	DB 2	FB 1	Daten für Dieselmotor
	DM_ausschalten	E 1.5	BOOL	Dieselmotor ausschalten
	DM_Drehzahl_Ist	MW 4	INT	tatsächliche Drehzahl für den Dieselmotor
	DM_Ein	A 5.4	BOOL	Kommando für Dieselmotor einschalten
	DM_einschalten	E 1.4	BOOL	Dieselmotor einschalten
	DM_Lüfter_einsc...	A 5.6	BOOL	Kommando für Dieselmotor-Lüfter einschalten
	DM_Nachlauf	T 2	TIMER	Nachlaufzeit für Dieselmotor-Lüfter

### Die Symboltabelle

In der Symboltabelle weisen Sie allen absoluten Adressen, die Sie in Ihrem späteren Programm ansprechen, einen symbolischen Namen und den Datentyp zu, z. B. für den Eingang "E 0.5" das Symbol "Automatik ein".

Diese Namen gelten für Ihr gesamtes Projekt und werden als globale Symbole bezeichnet.

Mit Hilfe der symbolischen Programmierung können Sie die Lesbarkeit Ihres Programms deutlich verbessern.



E 0.5 Automatik ein

E 1.0 BM\_einschalten

E 1.1 BM\_ausschalten

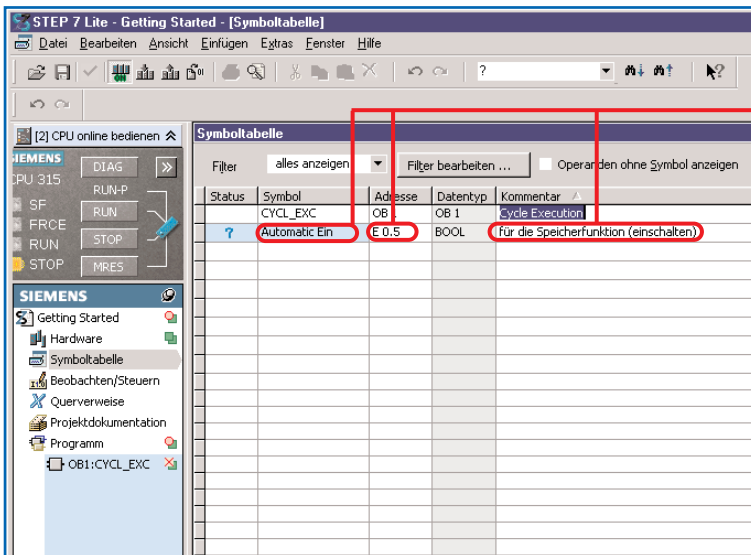
E 1.2 BM\_Störung

A 4.2 Automatikbetrieb

A 5.0 BM\_Ein

A 5.1 BM\_Soll\_erreicht

A 5.2 BM\_Lüfter\_ein



## Symboltabelle ausfüllen

Tragen Sie für Adresse "E0.5" in Spalte **Symbol** "Automatic Ein" ein. Tragen Sie in Spalte **Kommentar** den Kommentar wie links angegeben ein.

Bei der Eingabe

**Return** = Eine Zeile tiefer

**Ctrl + z** = Rückgängig

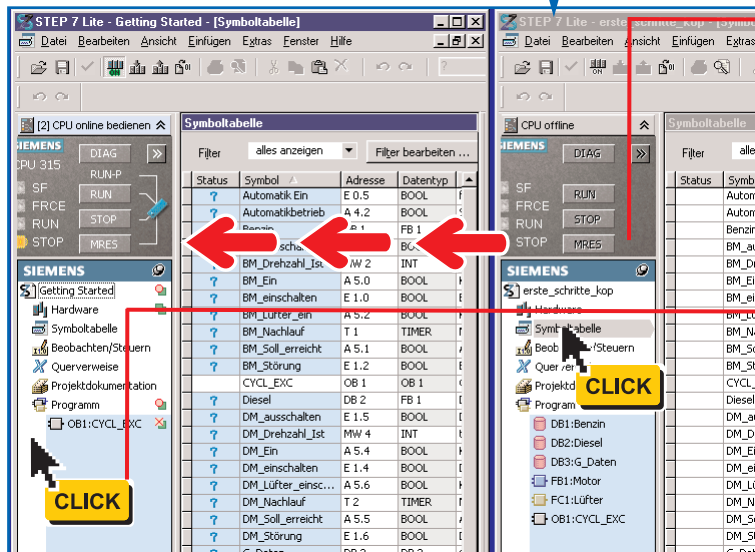
Sichern Sie Ihre Eintragungen mit **Datei > Speichern**.

## Symboltabelle kopieren

Da es für das Projekt "Getting Started" recht viele Symbole sind, kopieren Sie die Symboltabelle aus einem der mitgelieferten Beispielpunkte.



Eine zweite Instanz von STEP 7 Lite wird geöffnet



Öffnen Sie in einer zweiten Instanz von STEP 7 Lite zusätzlich das Projekt "erste\_schritte\_kop.k7p".

Klicken Sie im Projekt "erste\_schritte\_kop" mit der rechten Maustaste auf **Symboltabelle** und wählen Sie im Kontextmenü **Kopieren**.

Klicken Sie im Projekt "Getting Started" mit der rechten Maustaste in das Projektfenster und wählen Sie im Kontextmenü **Einfügen**.

Bestätigen Sie die Sicherheitsabfrage für das Überschreiben mit "OK".

Schließen Sie das Projekt "erste\_schritte\_kop". Speichern Sie das Projekt "Getting Started" über **Datei > Speichern**.

## Datentypen

Die Datentypen bestimmen die Art der Signale, die die CPU verarbeiten soll.

STEP 7 Lite verwendet u. a. die Datentypen links.

**BOOL**  
**BYTE**  
**WORD**  
**DWORD**

- Daten diesen Typs sind Bitkombinationen. 1 Bit (Typ BOOL) bis 32 Bit (DWORD).

**CHAR**

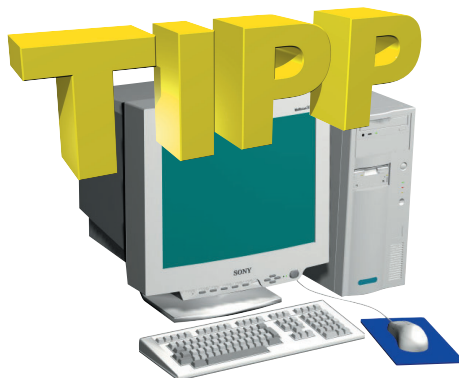
- Daten diesen Typs belegen genau 1 Zeichen des ASCII-Zeichensatzes.

**INT**  
**DINT**  
**REAL**

- Daten diesen Typs stehen für die Verarbeitung numerischer Werte zur Verfügung (z. B. zum Berechnen von arithmetischen Ausdrücken).

**S5TIME**  
**TIME**  
**DATE**  
**TIME\_OF\_DAY**

- Daten dieses Typs repräsentieren die unterschiedlichen Zeit- und Datumswerte innerhalb von STEP 7 Lite (z. B. zum Einstellen des Datums oder zum Eingeben des Zeitwerts).



Mehr Informationen zu Datentypen (z. B. zulässige Wertebereiche und Verwendungsbeispiele) finden Sie, indem Sie mit dem Hilfezeiger auf einen Datentyp klicken und zu **Einführung zu Daten- und Parametertypen** springen.

# 6

## Erste Programmierschritte



# Entscheiden Sie sich für KOP, FUP oder AWL



6.2

### Seiten 6.6 bis 6.11

Der Kontaktplan (KOP) eignet sich z. B. für Anwender aus dem gewerblichen Elektrobereich.

### Seiten 6.12 bis 6.17

Die Anweisungsliste (AWL) eignet sich z. B. für Anwender aus dem Umfeld der Informatik.

### Seiten 6.18 bis 6.23

Der Funktionsplan (FUP) eignet sich z. B. für Anwender aus dem Umfeld der Schaltungstechnik.

Egal ob KOP, FUP oder AWL, mit STEP 7 Lite erstellen Sie Ihr Programm in der gleichen Programmieroberfläche, dem Bausteineditor. Abhängig von der gewählten Programmiersprache wird die Oberfläche angepasst.

Entscheiden Sie sich z. B. für die Programmierung mit KOP, finden Sie Informationen dazu auf den Seiten 6.6 bis 6.11.

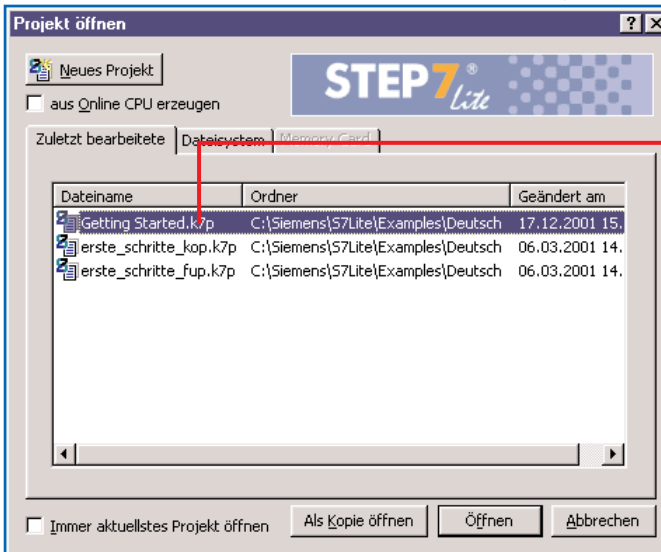




Das Dialogfeld für die Projektauswahl wird geöffnet.

## OB 1 öffnen.

Öffnen Sie STEP 7 Lite.

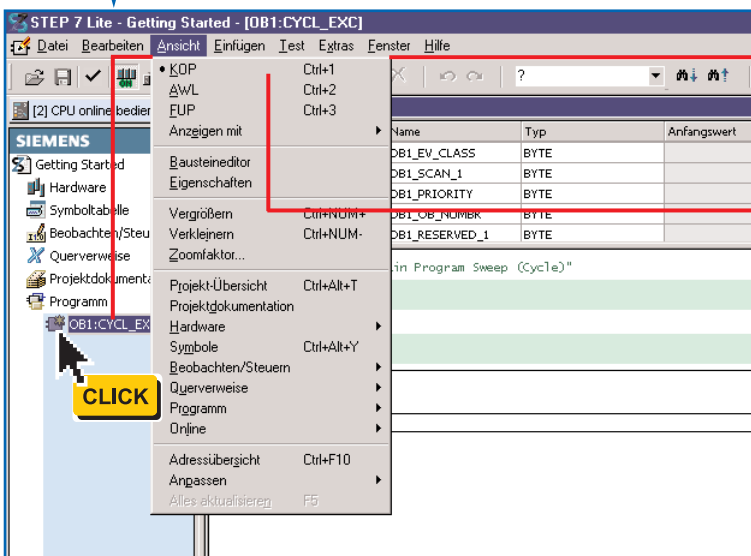


Doppelklicken Sie im Dialogfeld **Projekt öffnen** auf das Projekt "Getting Started.k7p".

Dieses Projekt haben Sie bereits in Kapitel 4 angelegt und in Kapitel 5 haben Sie für dieses Projekt die Symboltabelle ausgefüllt.

Ist dies nicht der Fall, öffnen Sie einfach ein "Neues Projekt" und kopieren Sie die Symboltabelle aus einem mitgelieferten Beispielprojekt.

STEP 7 Lite wird geöffnet.



Doppelklicken Sie auf den **OB 1**. Im Arbeitsbereich wird der Bausteineditor geöffnet.

Klicken Sie auf **Ansicht**. In diesem Menü können Sie prüfen, ob aktuell **KOP**, **FUP** oder **AWL** angewählt ist. Hier stellen Sie die Ansicht auch um.

### Hinweise:

Einige Befehle können nicht in allen drei Programmiersprachen angezeigt werden. Sie werden immer in AWL angezeigt.

Informationen zu den einzelnen Befehlen in KOP, FUP und AWL finden Sie über **F1 > Index > Referenzhilfen**.

Sie können im Bausteineditor arbeiten.



## Im Bausteineditor arbeiten

Ansicht der Programmiersprache ändern

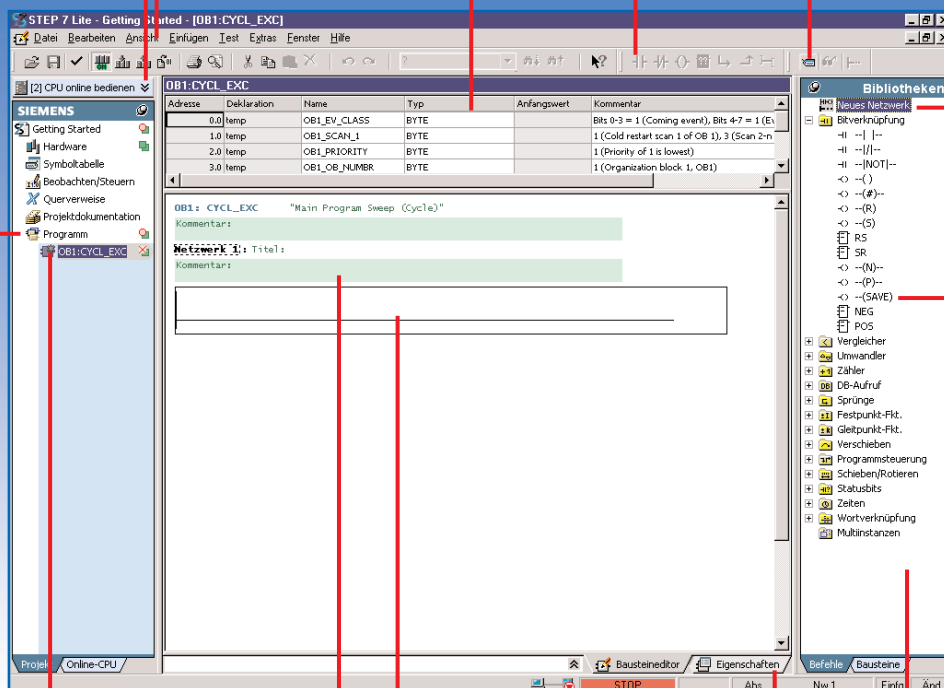
Die wichtigsten Befehle für KOP und FUP

Neues Netzwerk einfügen

CPU-Bedienpanel ein-/ausblenden

Variablendeklarationstabelle

Umschalten symbolische/absolute Programmierung



Projektelement "Programm"

Programmelement "OB 1"

Titel- und Kommentarfeld zum Netzwerk

Netzwerk für die Programmeingabe

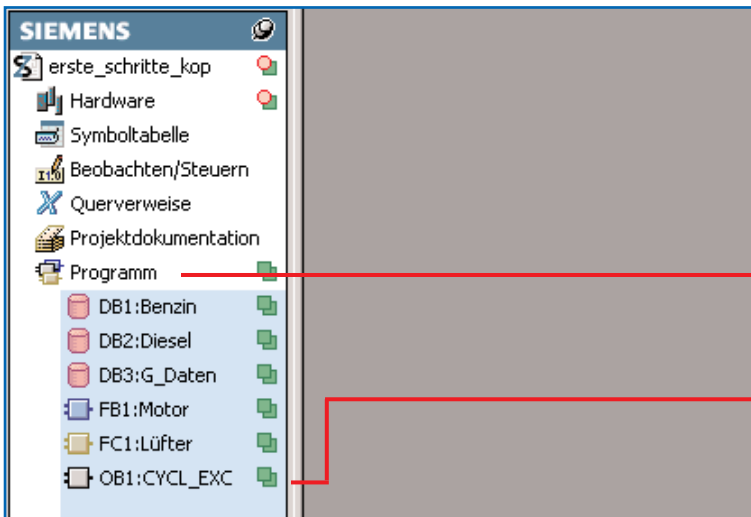
Bausteineigenschaften festlegen:  
z. B. Symbolischen Namen ändern

Alle Befehle für KOP und FUP

Klick mit dem Hilfezeiger ruft die Referenzhilfe zum Befehl auf

Im Bausteineditor programmieren Sie alle Bausteine.

Stellvertretend für die Programmiersprachen haben wir hier die Ansicht für KOP abgebildet.



## Projektelement "Programm"

In STEP 7 Lite ist das Anwenderprogramm in Bausteine aufgeteilt. Umfangreiche Programme lassen sich so übersichtlich programmieren.

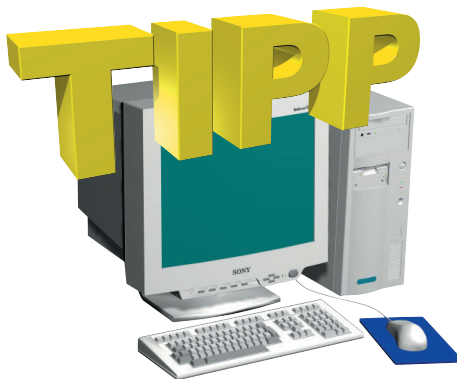
Unterhalb des Projektelements **Programm** werden diese Bausteine angezeigt.

In einem neu angelegten Projekt befindet sich nur der **OB 1**, der durch STEP 7 Lite automatisch angelegt wird. Später fügen Sie weitere Bausteine hinzu, z. B:

OB = Organisationsbaustein  
DB = Datenbaustein  
FB = Funktionsbaustein  
FC = Funktion

Der Organisationsbaustein OB 1 ist die Schnittstelle zum Betriebssystem der CPU und enthält das Hauptprogramm. Im OB 1 werden meist weitere Bausteine aufgerufen und die spezifischen Parameter übergeben, die für die Steuerung des Prozesses erforderlich sind.

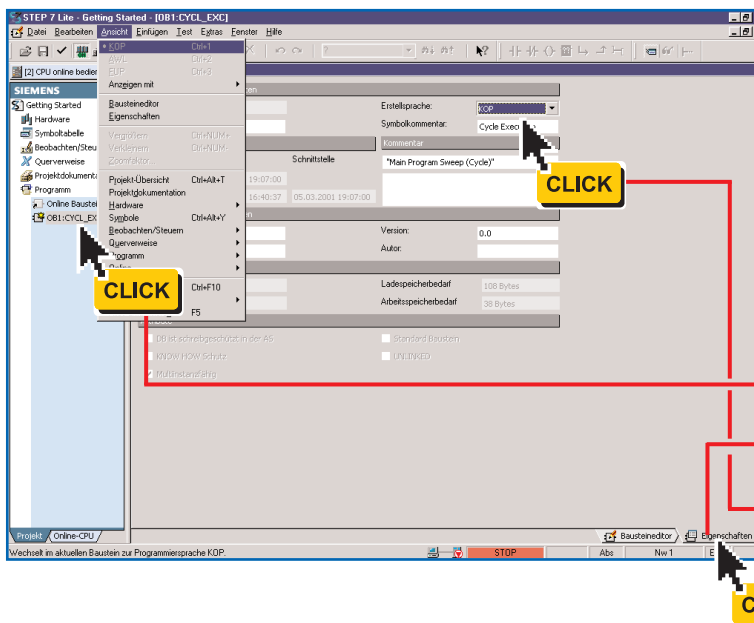
6.5



Mit dem Kontextmenübefehl **Umbenennen** können Sie einen Baustein umbenennen.

Als Einsteiger benötigen Sie vielleicht mehr Informationen zum Arbeiten mit Bausteinen. Klicken Sie hierzu links in das Projektfenster und drücken Sie dann **F1 > Index > Aufrufhierarchie im Anwenderprogramm**.

## OB 1 mit KOP programmieren



Im Folgenden werden Sie eine Reihenschaltung, eine Parallelschaltung und die Speicherfunktion Setzen und Rücksetzen in KOP (Kontaktplan) programmieren.

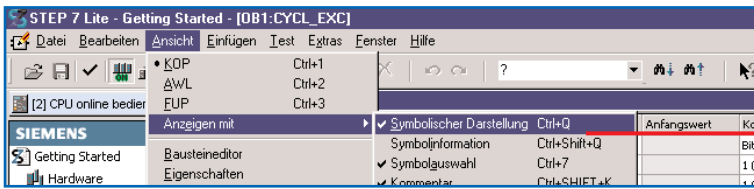
Legen Sie die Programmiersprache fest, in der der OB 1 programmiert und zukünftig auch geöffnet werden soll:

Doppelklicken Sie auf **OB 1**.

Klicken Sie auf **Eigenschaften**.

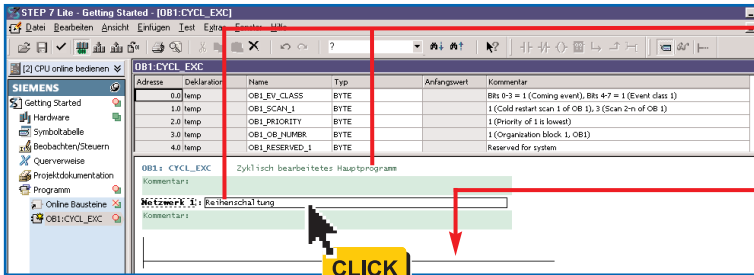
Wählen Sie **KOP**. Der OB 1 wird zukünftig in KOP geöffnet.

Verlassen Sie das Dialogfeld **Eigenschaften**. Im Menü **Ansicht** ist jetzt auch **KOP** markiert.



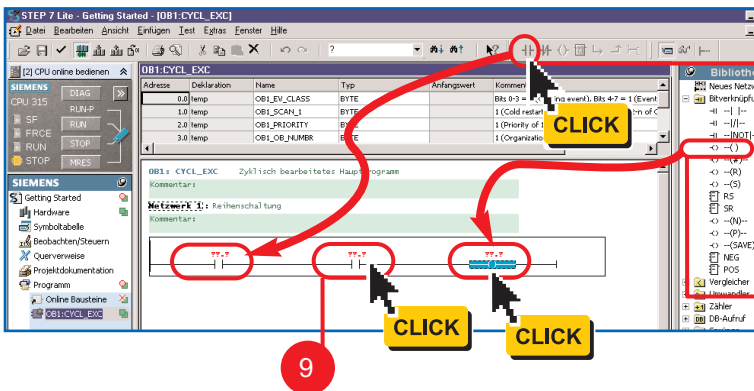
## Reihenschaltung mit KOP programmieren

Wählen Sie im Menü **Ansicht** die symbolische Darstellung aus.



Geben Sie bei **OB 1** "Zyklisch bearbeitetes Hauptprogramm" ein. Geben Sie bei **Netzwerk 1** "Reihenschaltung" ein.

Markieren Sie den leeren Strompfad durch einen Klick.

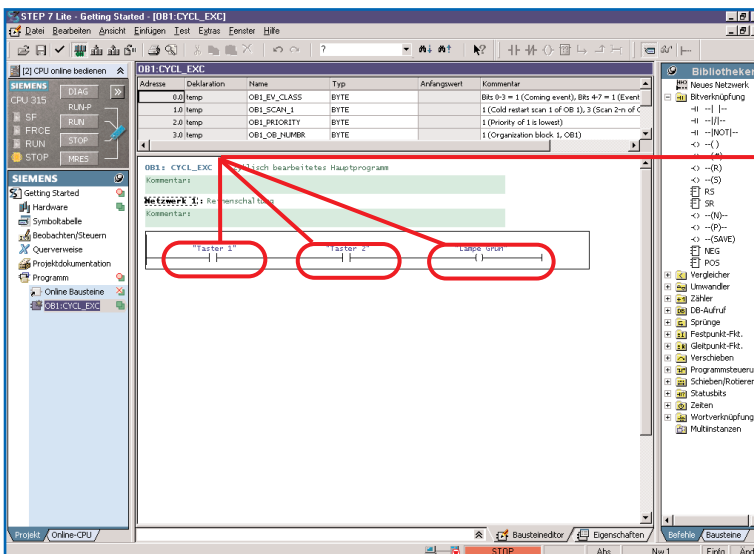


Fügen Sie drei Programmelemente auf verschiedene Art ein:

Klicken Sie auf das Symbol für den Schließer. Er wird sofort eingefügt.

Klicken Sie mit der rechten Maustaste auf den Strompfad und wählen Sie im Kontextmenü Schließer.

Ziehen Sie die Spule mit Drag&Drop in den Strompfad.



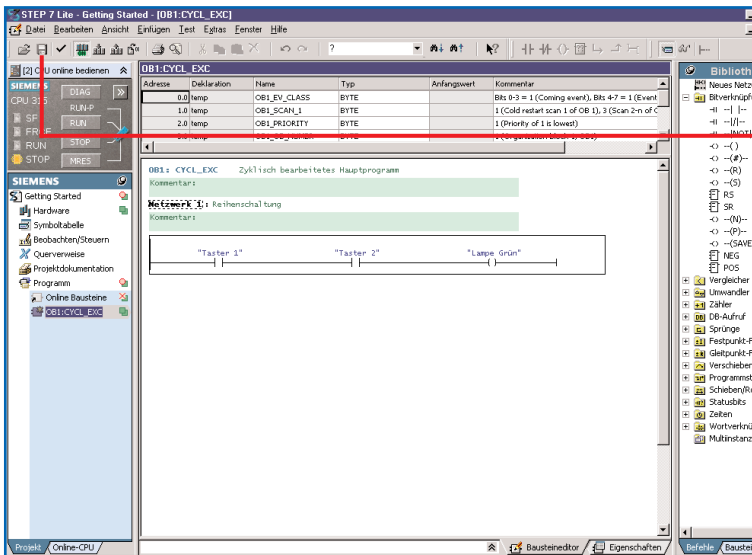
In der Reihenschaltung fehlt noch die Adressierung von Schließern und Spule:

Klicken Sie auf **???** und tragen Sie den symbolischen Namen "Taster 1" ein (mit Anführungszeichen). Oder wählen Sie den Namen aus der Symbolauswahlliste aus, die Ihnen bei Klick auf **???** angeboten wird.

Bestätigen Sie mit **Return**.

Tragen Sie für den zweiten Schließer den symbolischen Namen "Taster 2" ein.

Tragen Sie für die Spule den Namen "Lampe Grün" ein.

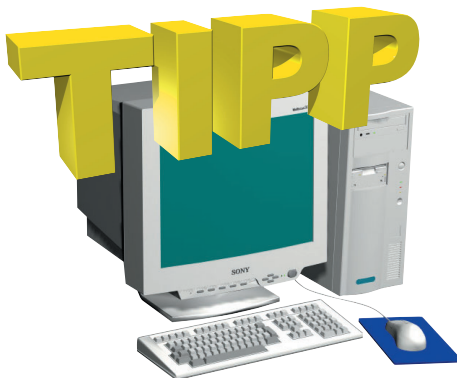


Die Reihenschaltung ist vollständig programmiert.

12 Speichern Sie Ihre Eingaben, falls keine Symbole mehr rot gekennzeichnet sind, durch Klick auf das Symbol Diskette.

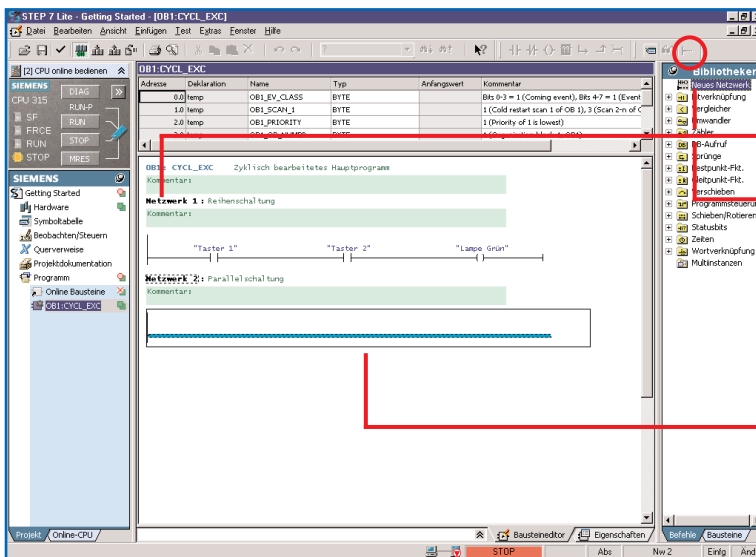
Nicht nur Ihre Eingaben in OB 1, sondern alle Elemente des Projekts werden gespeichert.

13 Wählen Sie den Menübefehl **Bearbeiten > Übernehmen**, werden Ihre Eingaben (immer der Inhalt des aktiven Fensters) in einer temporären Datei zwischengespeichert. Diese Art, Daten zu sichern, ist immer dann zu empfehlen, wenn Sie Änderungen eventuell wieder rückgängig machen wollen. Schließen Sie Ihr Projekt nachdem Sie immer nur Daten übernommen haben, werden Sie gefragt, ob Sie Ihre Änderungen speichern möchten.



Symbole werden rot gekennzeichnet, wenn z. B. das Symbol nicht in der Symboltabelle enthalten ist oder ein Syntaxfehler vorliegt.

Speichern ist dann nicht möglich und am unteren Rand des Editorfensters erscheint eine Klartextfehlermeldung, die Hinweise für die richtige Vorgehensweise enthält.



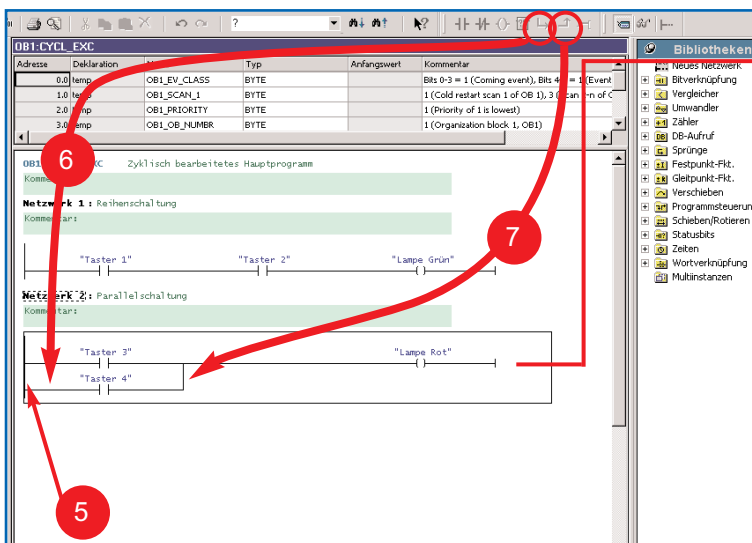
## Parallelschaltung mit KOP programmieren

1 Markieren Sie das Netzwerk 1.

2 Fügen Sie ein neues Netzwerk ein.

Das geht alternativ auch über ein Symbol in der Symbolleiste, über die rechte Maustaste oder über CTRL R.

3 Markieren Sie wieder den Strompfad.



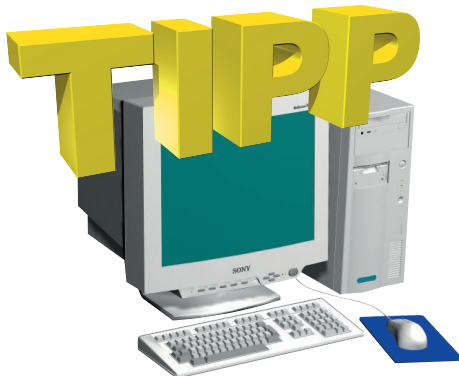
4 Fügen Sie einen Schließer und eine Spule ein. Hier "Taster 3" und "Lampe Rot".

5 Markieren Sie die linke Stromschiene.

6 Fügen Sie einen Parallelzweig ein. In den Parallelzweig kommt ein weiterer Schließer.

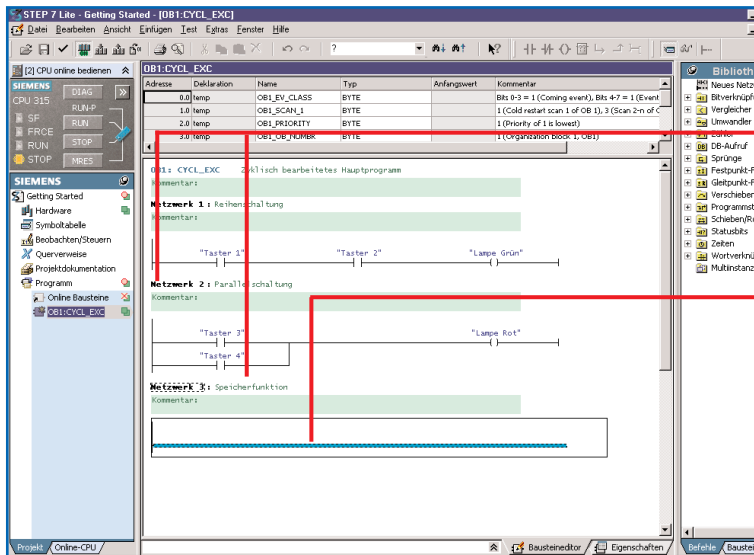
7 Schließen Sie die Verzweigung via Symbol oder durch Ziehen der Doppelpfeilspitze, die nach dem Setzen des Schließers sichtbar ist.

8 Im Parallelzweig fehlt nur noch die Adressierung. Geben Sie die Namen wie in der Abbildung ein. Speichern Sie Ihre Eingaben.



Vergeben Sie prägnante, kurze Namen für die Netzwerke. Das erleichtert die Suche beim Scrollen mit dem rechten Scrollbalken in umfangreichen Programmen. Beim Scrollen werden die Namen angezeigt.

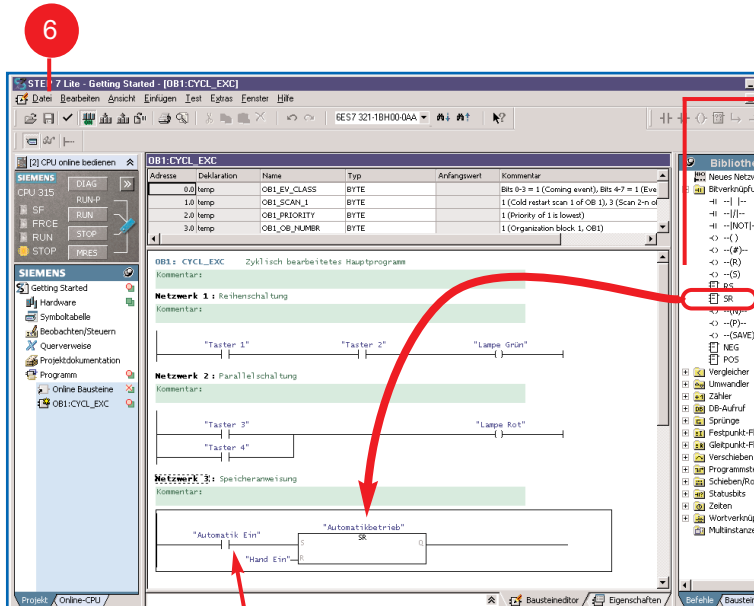
# Erste Programmierschritte



## Speicherfunktion mit KOP programmieren

1 Markieren Sie das Netzwerk 2, fügen Sie ein weiteres Netzwerk ein und tragen Sie den Titel "Speicherfunktion" ein.

2 Markieren Sie wieder den Strompfad.



3 Navigieren Sie in der Befehlsübersicht über **Bitverknüpfungen** zum **SR-Element** und fügen Sie das Element ein.

4 Fügen Sie vor dem Eingang S einen Schließer ein.

5 Tragen Sie folgende symbolische Namen ein:

- oberer Schließer "Automatik Ein",
- Eingang R "Hand Ein",
- SR-Element "Automatikbetrieb".

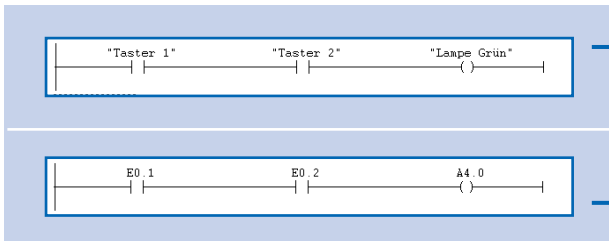
6 Speichern Sie Ihre Eingaben über **Datei > Speichern**.



## Programmieroberfläche anpassen

Über Menübefehle können Sie in STEP 7 Lite die Programmieroberfläche individuell einrichten.

Menü **Ansicht** - Beispiele:



1

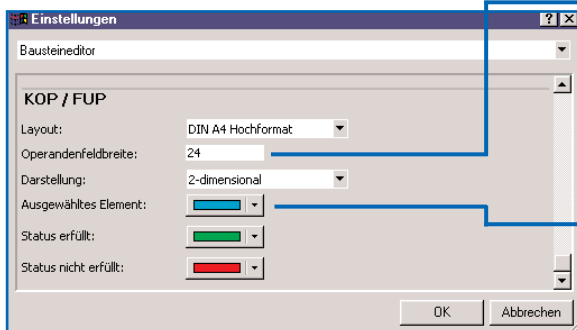
Symbolische Adressierung in KOP:  
**Ansicht > Anzeigen mit > Symbolischer Darstellung** aktivieren

2

Absolute Adressierung in KOP:  
**Ansicht > Anzeigen mit > Symbolischer Darstellung** deaktivieren

Programmiersprache wechseln  
**Ansicht > KOP/FUP/AWL**

Menübefehl **Extras > Einstellungen** - Beispiele:



3

Zeilenumbruch der symbolischen Adressierung zwischen dem 10. und 24. Zeichen einstellen:  
**Extras > Einstellungen > KOP/FUP > Operandenfeldbreite**

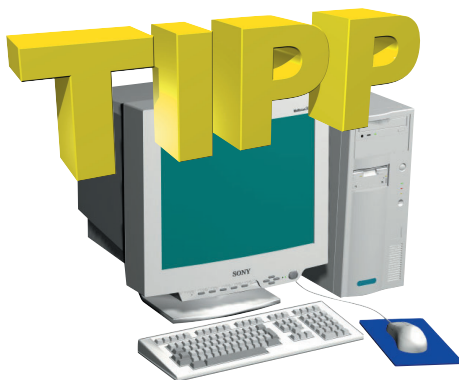
4

Farbdarstellungen im Strompfad ändern:  
**Extras > Einstellungen > KOP/FUP > Ausgewähltes Element**

7

Schließen Sie den Baustein über das Fenster-Symbol **Schließen**.

6.11

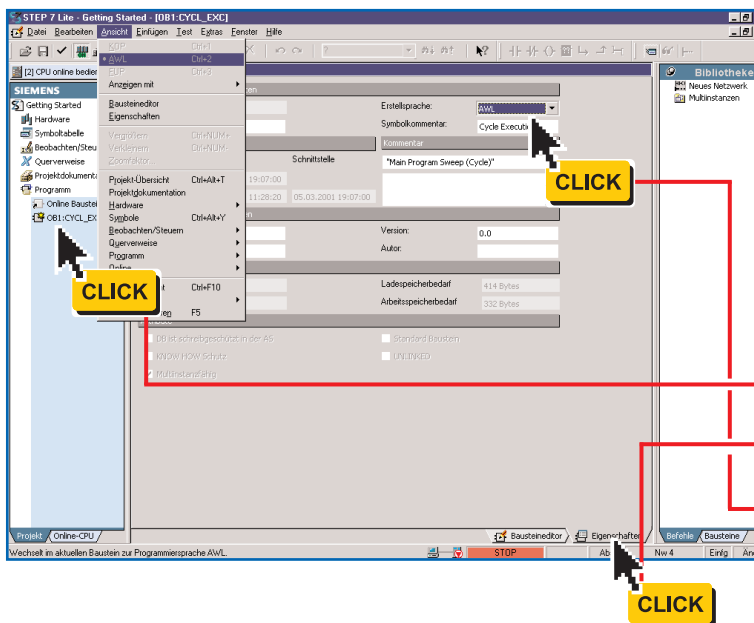


Insbesondere der Menübefehl **Extras > Einstellungen** bietet viele Möglichkeiten, das Erscheinungsbild von STEP 7 Lite mit Farben, Schriftarten, Operandenfeldbreite usw. zu ändern.

## OB 1 mit AWL programmieren



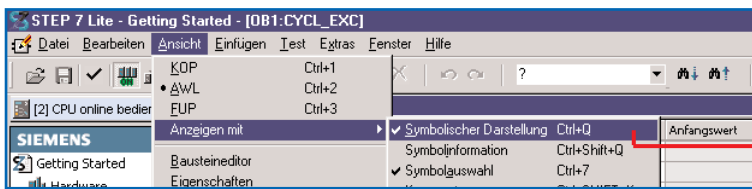
6.12



Im Folgenden werden Sie eine UND-Anweisung, eine ODER-Anweisung und die Speicheranweisungen Setzen bzw. Rücksetzen in AWL (Anweisungsliste) programmieren.

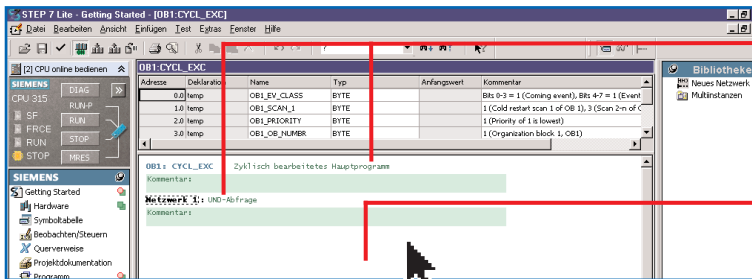
Legen Sie die Programmiersprache fest, in der der OB 1 programmiert und zukünftig geöffnet werden soll:

- 1 Doppelklicken Sie auf **OB 1**.
- 2 Klicken Sie auf **Eigenschaften**.
- 3 Wählen Sie **AWL**. Der OB 1 wird zukünftig in AWL geöffnet.
- 4 Verlassen Sie das Dialogfeld **Eigenschaften**. Im Menü **Ansicht** ist ebenfalls **AWL** markiert.



## UND-Anweisung mit AWL programmieren

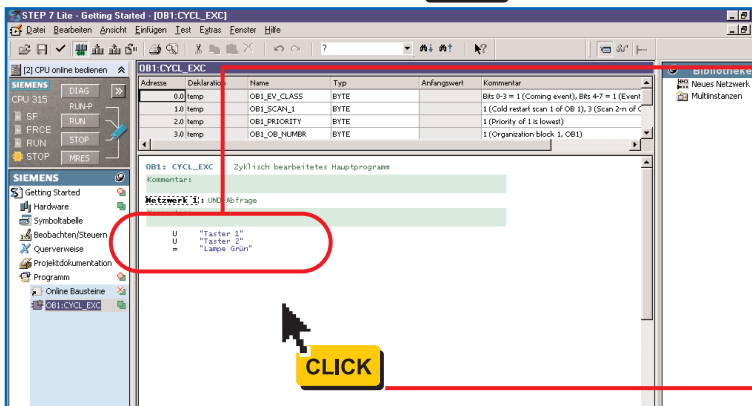
Wählen Sie im Menü Ansicht die symbolische Darstellung aus.



Geben Sie bei **OB 1** "Zyklisch bearbeitetes Hauptprogramm" ein. Geben Sie bei **Netzwerk 1** "UND-Abfrage" ein.



Klicken Sie in den Eingabebereich.



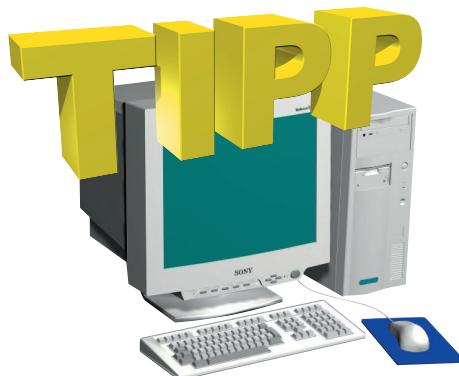
Schreiben Sie in die erste Programmzeile ein "U" (UND), danach ein Leerzeichen und dann das Symbol "Taster 1" (mit Anführungszeichen).

Schließen Sie die Zeile mit Return ab. Der Cursor springt in die neue Zeile.

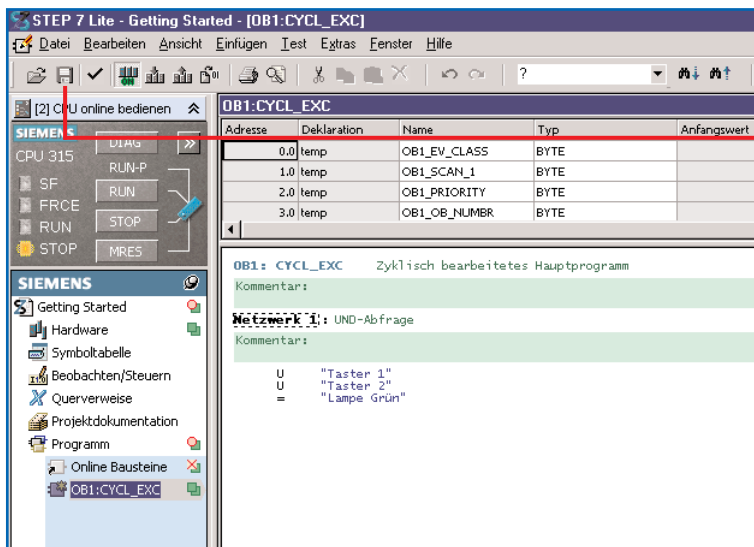
Schreiben Sie in der nächsten Zeile wieder ein "U" und klicken Sie diesmal mit der rechten Maustaste in den Eingabebereich.

Wählen Sie über die rechte Maustaste den Kontextmenübefehl **Symbol einfügen** und fügen Sie aus der Liste "Taster 2" ein.

Geben Sie in der nächsten Zeile ein "=" ein und danach entweder über die Tastatur oder über das Kontextmenü "Lampe grün" ein.



In einer Eingabezeile müssen Sie nicht am Zeilenanfang mit Ihrer Eingabe beginnen. Egal wo Sie beginnen, nach Return stellt STEP 7 Lite die Anweisungen übersichtlich, spaltengerecht untereinander.



11

Die UND-Abfrage ist vollständig programmiert.

Speichern Sie Ihre Eingaben, falls kein Text mehr rot gekennzeichnet ist, durch Klick auf das Symbol Diskette.

Nicht nur Ihre Eingaben in OB 1, sondern alle Elemente des Projekts werden gespeichert.

12

Wählen Sie den Menübefehl **Bearbeiten > Übernehmen**, werden Ihre Eingaben (immer der Inhalt des aktiven Fensters) in einer temporären Datei zwischengespeichert.

Diese Art, Daten zu sichern, ist immer dann zu empfehlen, wenn Sie Änderungen eventuell wieder rückgängig machen wollen.

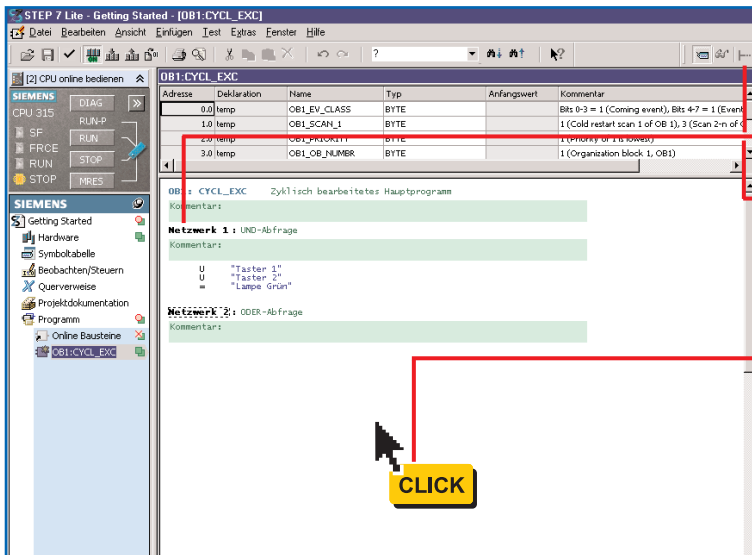
Schließen Sie Ihr Projekt nachdem Sie immer nur Daten übernommen haben, werden Sie gefragt, ob Sie Ihre Änderungen speichern möchten.

6.14



Symbole werden rot gekennzeichnet, wenn z. B. das Symbol nicht in der Symboltabelle enthalten ist oder ein Syntaxfehler vorliegt.

Speichern ist dann nicht möglich und am unteren Rand des Editorfensters erscheint eine Klartextfehlermeldung, die Hinweise für die richtige Vorgehensweise enthält.

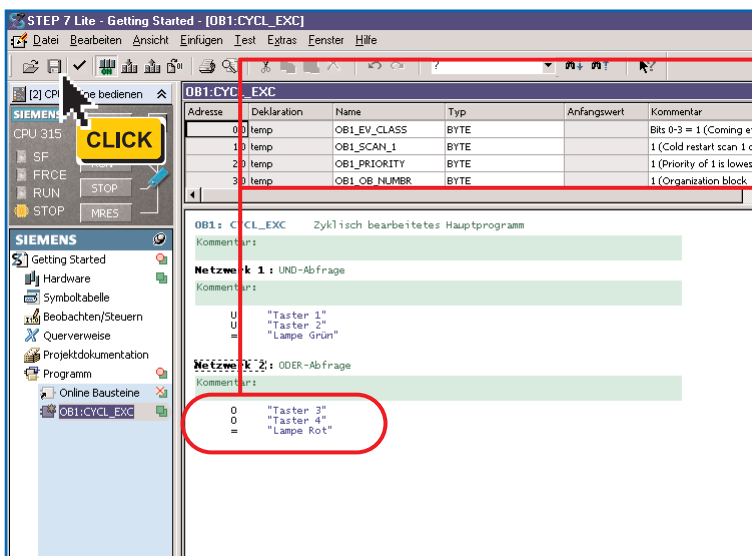


## ODER-Anweisung in AWL programmieren

1 Markieren Sie das Netzwerk 1.

2 Fügen Sie ein neues Netzwerk ein. Das geht alternativ auch über das Symbol in der Symbolleiste, über die rechte Maustaste oder CTRL R.

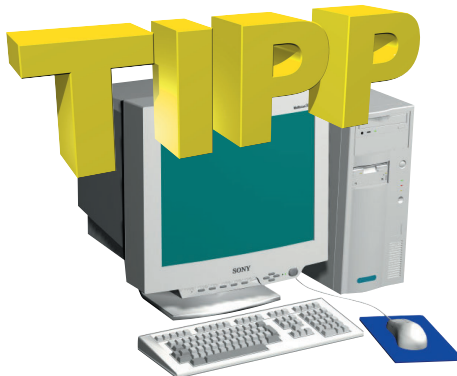
3 Klicken Sie wieder in den Eingabebereich.



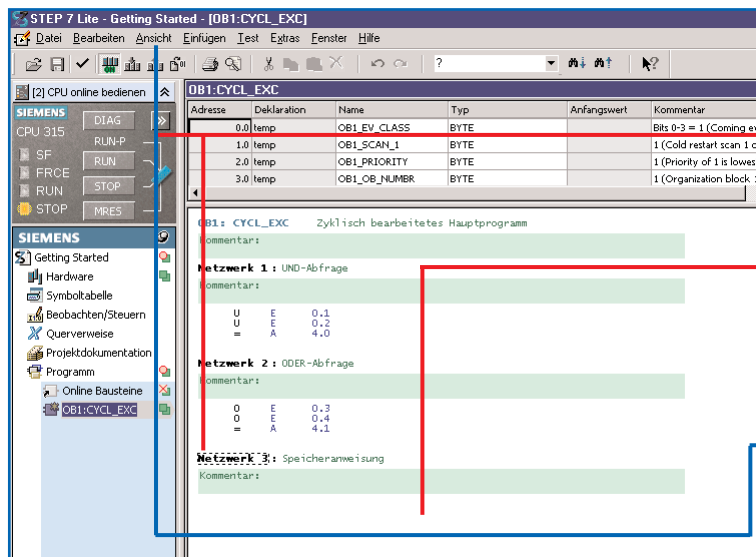
4 Tragen Sie ein "O" (ODER) und das Symbol "Taster 3" (analog zum UND) ein.

5 Vervollständigen Sie die ODER-Anweisung und speichern Sie Ihre Eingaben.

6.15



Vergeben Sie prägnante, kurze Namen für die Netzwerke. Das erleichtert die Suche beim Scrollen mit dem rechten Scrollbalken in umfangreichen Programmen. Beim Scrollen werden die Namen angezeigt.

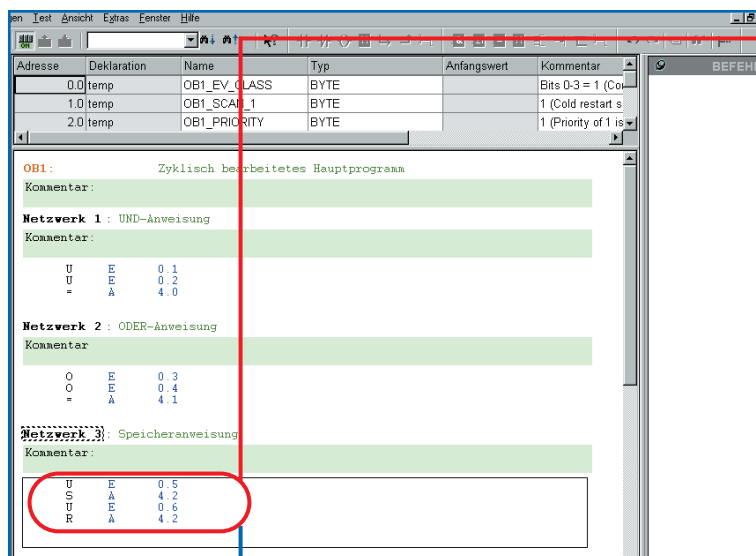


## Speicheranweisung in AWL programmieren

1 Markieren Sie das Netzwerk 2 und fügen Sie ein weiteres Netzwerk ein.

2 Klicken Sie wieder in den Eingabebereich.

3 Über **Ansicht > Anzeigen mit > Symbolischer Darstellung** können Sie wie links gezeigt von symbolischer zu absoluter Darstellung umschalten.



4 In die erste Zeile schreiben Sie die Anweisung U mit dem symbolischen Namen "Automatik Ein". Vervollständigen Sie die Speicheranweisung wie folgt:

Symbolisch:

- U "Automatik Ein"
- S "Automatikbetrieb"
- U "Hand Ein"
- R "Automatikbetrieb"

5 Speichern Sie Ihre Eingaben über **Datei > Speichern**.

6 Hatten Sie in Schritt 3 umgeschaltet, geben Sie wie folgt ein:

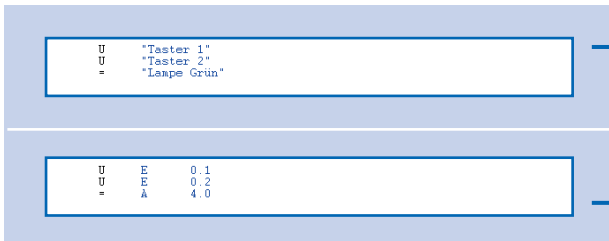
Absolut:

- U E 0.5
- S A 4.2
- U E 0.6
- R A 4.2

## Programmiersoberfläche anpassen

Über Menübefehle können Sie in STEP 7 Lite die Programmiersoberfläche individuell einrichten.

Menü **Ansicht** - Beispiele:



1

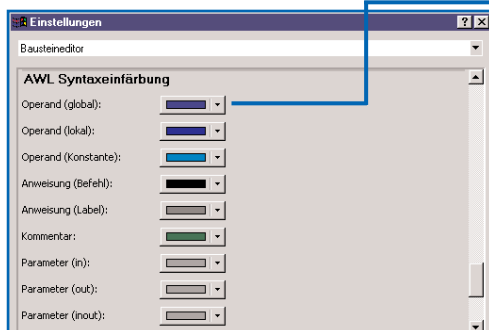
Symbolische Adressierung in AWL:  
**Ansicht > Anzeigen mit > Symbolischer Darstellung** aktivieren

2

Absolute Adressierung in AWL:  
**Ansicht > Anzeigen mit > Symbolischer Darstellung** deaktivieren

Programmiersprache wechseln  
**Ansicht > KOP/FUP/AWL**

Menübefehl **Extras > Einstellungen** - Beispiel:



3

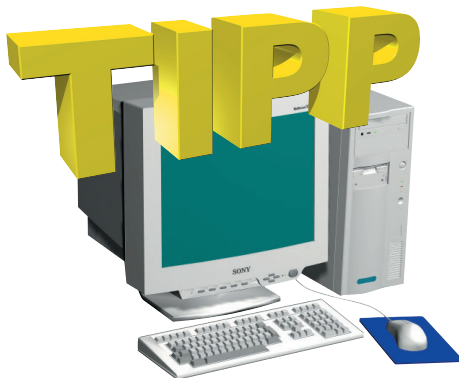
Farbdarstellungen der Anweisungen ändern:

**Extras > Einstellungen > AWL > AWL Syntaxeinfärbung**

7

Schließen Sie den Baustein über das Fenster-Symbol **Schließen**

6.17



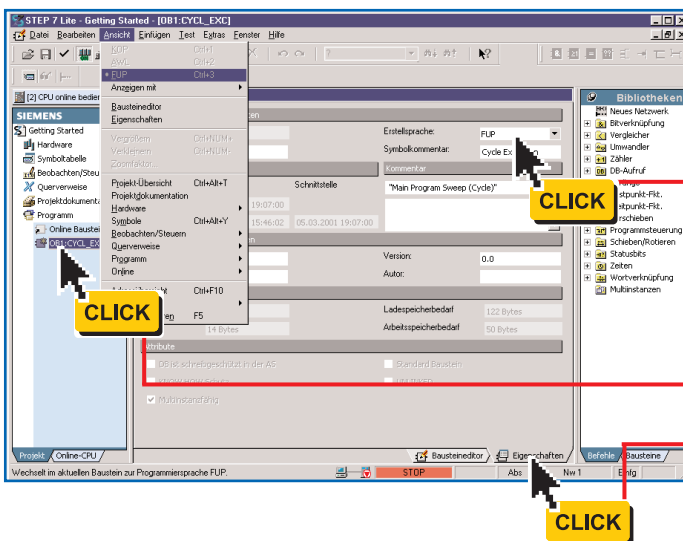
Insbesondere der Menübefehl **Extras > Einstellungen** bietet viele Möglichkeiten, das Erscheinungsbild von STEP 7 Lite mit Farben, Schriftarten usw. zu ändern.



## OB 1 mit FUP programmieren



6.18



Im Folgenden werden Sie eine UND-Funktion, eine ODER-Funktion und eine Speicherfunktion in FUP (Funktionsplan) programmieren.

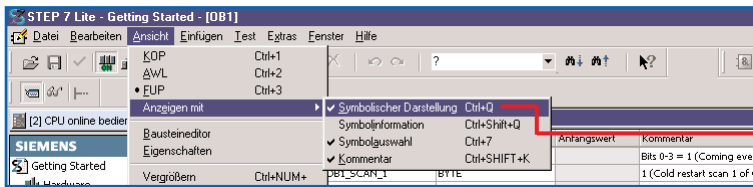
Legen Sie die Programmiersprache fest, in der der OB 1 programmiert und zukünftig geöffnet werden soll:

Doppelklicken Sie auf **OB 1**.

Klicken Sie auf **Eigenschaften**.

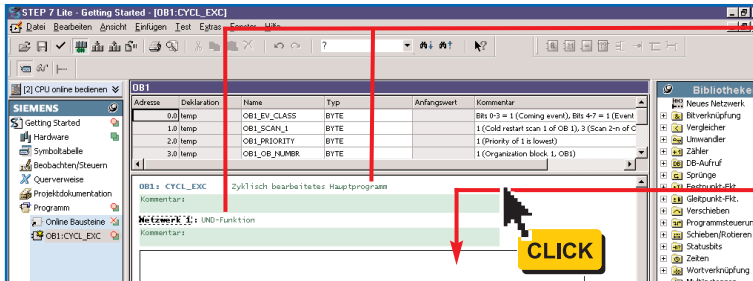
Wählen Sie **FUP**. Der OB 1 wird zukünftig in FUP geöffnet.

Verlassen Sie das Dialogfeld **Eigenschaften**. Im Menü **Ansicht** ist ebenfalls **FUP** markiert.



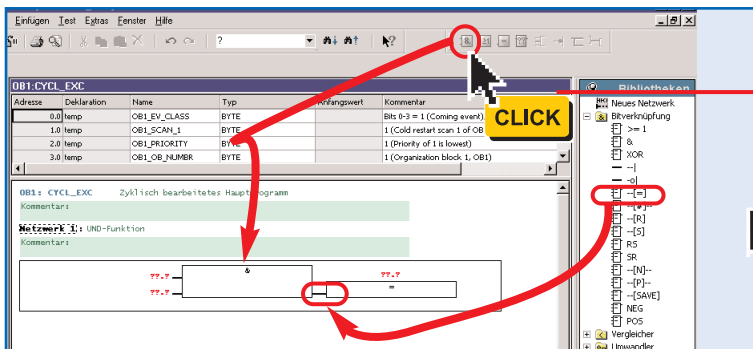
## UND-Funktion in FUP programmieren

Wählen Sie im Menü **Ansicht** die symbolische Darstellung aus.



Geben Sie bei **OB 1** "Zyklisch bearbeitetes Hauptprogramm" ein. Geben Sie bei **Netzwerk 1** "Und-Funktion" ein.

Klicken Sie in den Eingabebereich.

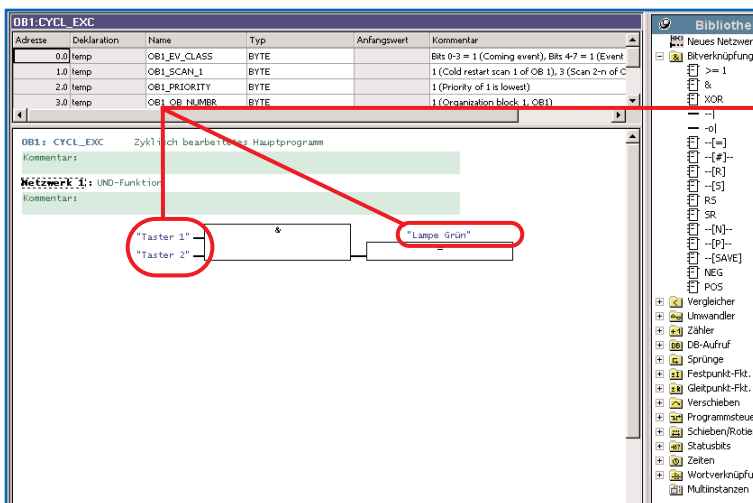


Fügen Sie zwei Programmelemente auf verschiedene Art ein:

Klicken Sie auf das Symbol für das UND-Gatter. Es wird sofort eingefügt.

Ziehen Sie die Zuweisung mit Drag&Drop auf den in der Grafik umrandeten Bereich. Treffen Sie nicht, steht die Zuweisung unter der UND-Box.

Alternativ zum Einfügen per Drag& Drop: Markieren Sie dem umrandeten Bereich und doppelklicken Sie auf die Zuweisung.

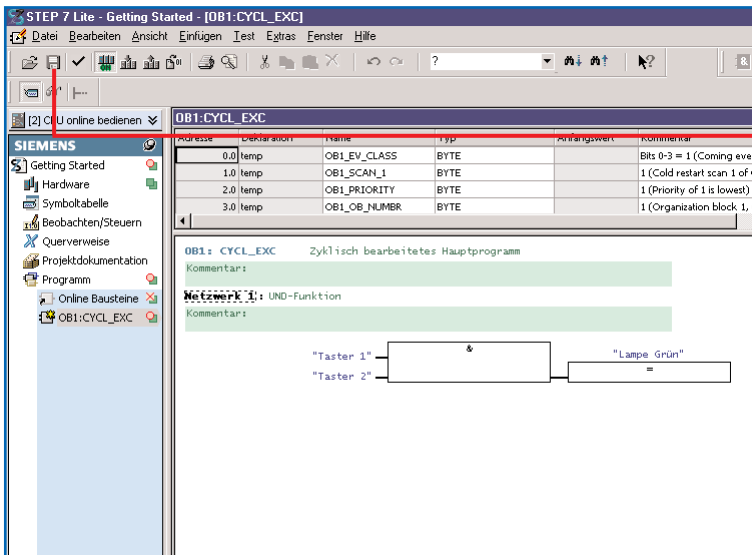


In der UND-Funktion fehlt noch die Adressierung:

Klicken Sie auf **???** und tragen Sie den symbolischen Namen "Taster 1" ein (mit Anführungszeichen). Oder klicken Sie auf die Fragezeichen, wählen Sie über die rechte Maustaste **Symbol einfügen** und fügen Sie den Namen aus der Liste ein.

Tragen Sie für den 2. Eingang der UND-Box den Namen "Taster 2" ein.

Tragen Sie für die Zuweisung den Namen "Lampe Grün" ein.



12

Die UND-Funktion ist vollständig programmiert.

Speichern Sie Ihre Eingaben, falls keine Symbole mehr rot gekennzeichnet sind, durch Klick auf das Symbol Diskette.

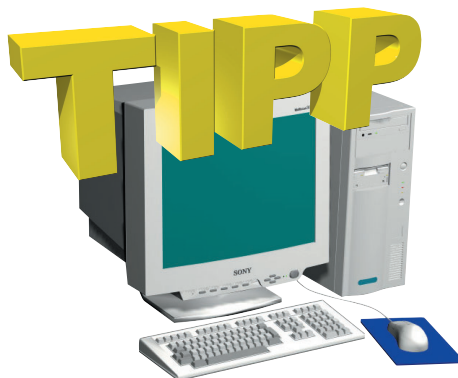
Nicht nur Ihre Eingaben in OB 1, sondern alle Elemente des Projekts werden gespeichert.

Wählen Sie den Menübefehl **Bearbeiten > Übernehmen**, werden Ihre Eingaben (immer der Inhalt des aktiven Fensters) in einer temporären Datei zwischengespeichert.

Diese Art, Daten zu sichern, ist immer dann zu empfehlen, wenn Sie Änderungen eventuell wieder rückgängig machen wollen.

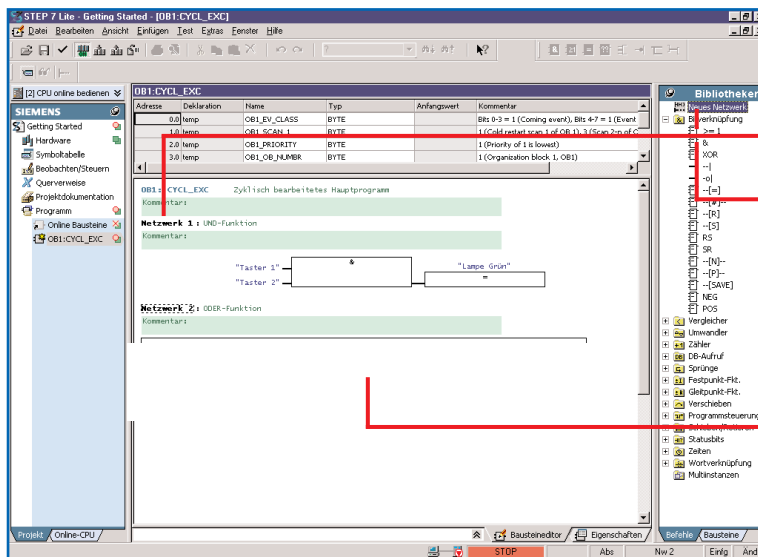
Schließen Sie Ihr Projekt nachdem Sie immer nur Daten übernommen haben, werden Sie gefragt, ob Sie Ihre Änderungen speichern möchten.

6.20



Symbole werden rot gekennzeichnet, wenn z. B. das Symbol nicht in der Symboltabelle enthalten ist oder ein Syntaxfehler vorliegt.

Speichern ist dann nicht möglich und am unteren Rand des Editorfensters erscheint eine Klartextfehlermeldung, die Hinweise für die richtige Vorgehensweise enthält.

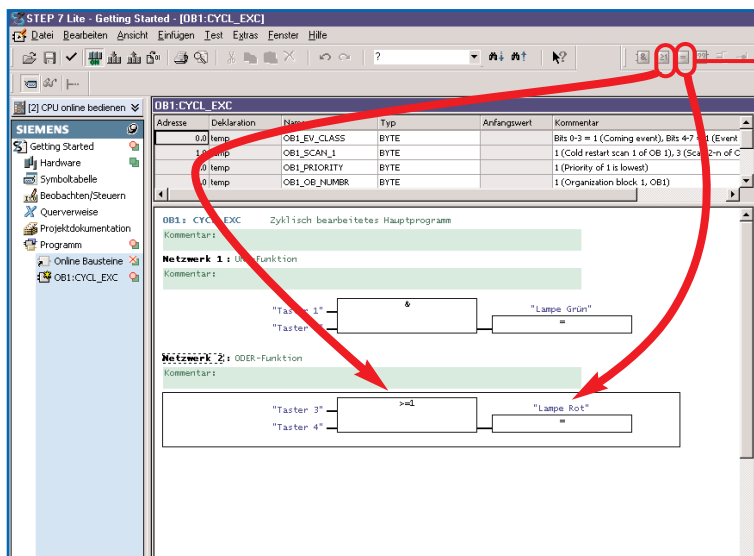


## ODER-Funktion in FUP programmieren

- 1 Markieren Sie das Netzwerk 1.
- 2 Fügen Sie ein neues Netzwerk ein.

Das geht alternativ auch über das Symbol in der Symbolleiste, über die rechte Maustaste oder CTRL R.

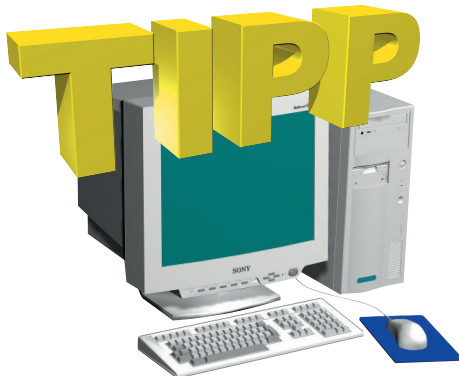
- 3 Markieren Sie wieder den Eingabebereich.



- 4 Fügen Sie eine ODER-Funktion und eine Zuweisung ein.

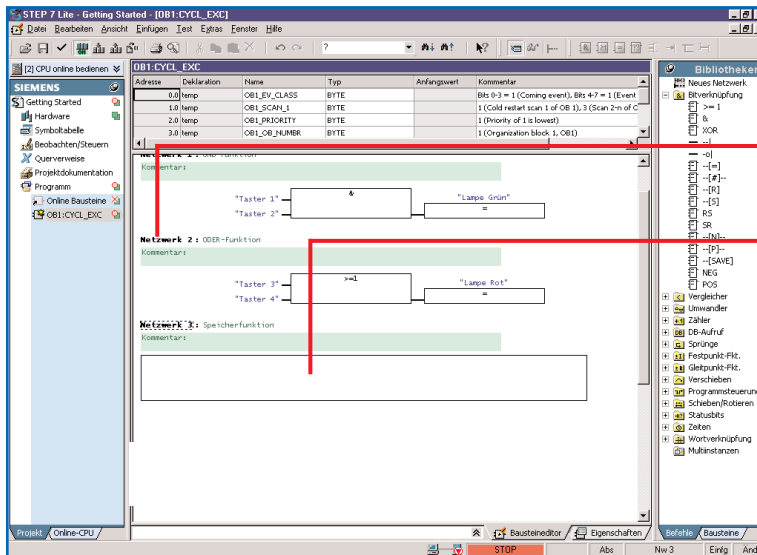
- 5 Es fehlt nur noch die Adressierung. Geben Sie die Namen wie in der Abbildung links ein. Speichern Sie Ihre Eingaben.

6.21



Vergeben Sie prägnante, kurze Namen für die Netzwerke. Das erleichtert die Suche beim Scrollen mit dem Scrollbalken rechts in umfangreichen Programmen. Beim Scrollen werden die Namen angezeigt.

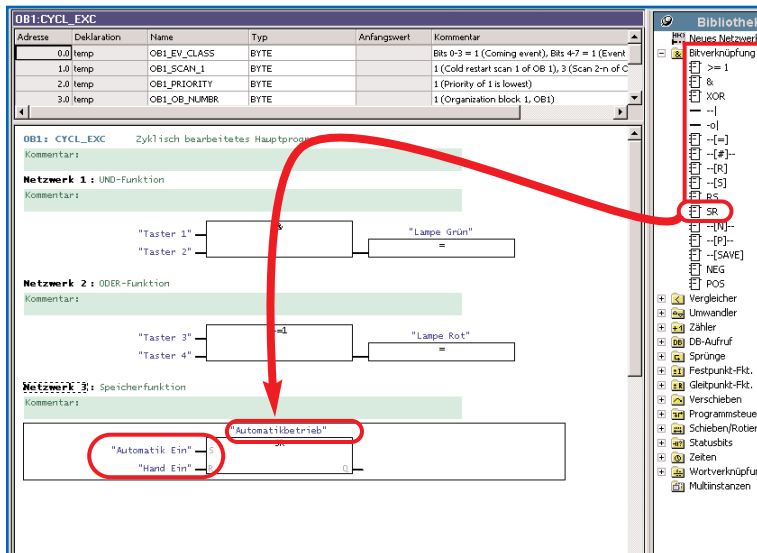
# Erste Programmierschritte



## Speicherfunktion in FUP programmieren

1 Markieren Sie das Netzwerk 2, und fügen Sie ein weiteres Netzwerk ein.

2 Markieren Sie wieder den Eingabebereich.



3 Navigieren Sie in der Befehlsübersicht über **Bitverknüpfungen** zum **SR-Element**.

Fügen Sie das Element ein.

4 Tragen Sie folgende symbolische Namen ein:

- Setzen "Automatik Ein",
- Rücksetzen "Hand Ein",
- Flipflop setzen rücksetzen "Automatikbetrieb".

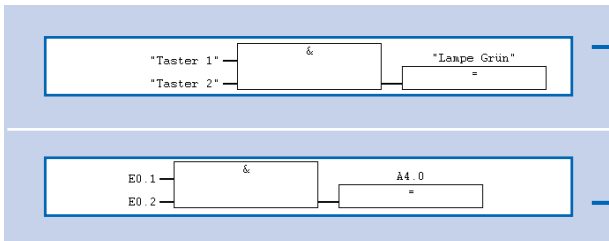
5 Speichern Sie Ihre Eingaben über **Datei > Speichern**.

6.22

## Programmiersoberfläche anpassen

Über Menübefehle können Sie in STEP 7 Lite die Programmiersoberfläche individuell einrichten.

Menü **Ansicht** – Beispiele:



1

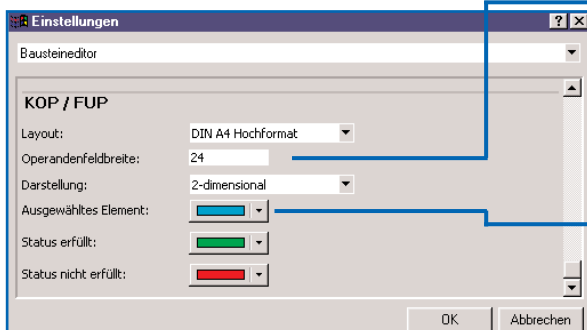
Symbolische Adressierung in FUP:  
**Ansicht > Anzeigen mit > Symbolischer Darstellung** aktivieren

2

Absolute Adressierung in KOP:  
**Ansicht > Anzeigen mit > Symbolischer Darstellung** deaktivieren

Programmiersprache wechseln  
**Ansicht > KOP/FUP/AWL**

Menü **Extras > Einstellungen** – Beispiele:



3

Zeilenumbruch der symbolischen Adressierung zwischen dem 10. und 24. Zeichen einstellen:  
**Extras > Einstellungen > KOP/FUP > Operandenfeldbreite**

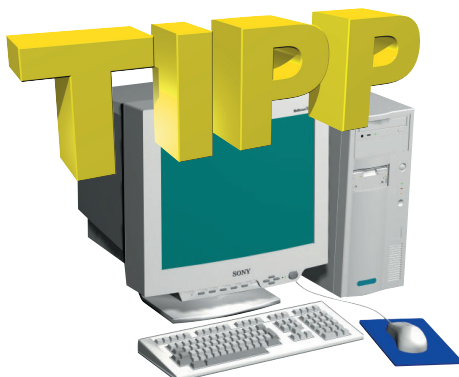
4

Farbdarstellungen ändern:  
**Extras > Einstellungen > KOP/FUP > Ausgewähltes Element**

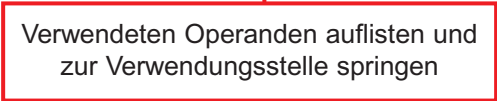
6

Schließen Sie den Baustein über das Fenster-Symbol **Schließen**.

6.23



Insbesondere das Menü **Extras** bietet viele Möglichkeiten, das Erscheinungsbild von STEP 7 Lite mit Farben, Schriftarten, Operandenfeldbreite usw. zu ändern.



Nutzen Sie die Sichten "Querverweisliste", "Verwendete Operanden" und "Programmstruktur", um einen Überblick über die Nutzung und Verwendung von Operanden, Speicherbereichen, Bausteinen etc. zu erhalten. Zu den Querverweisen gelangen Sie durch Doppelklicken des Symbols "Querverweise" im Projektfenster.

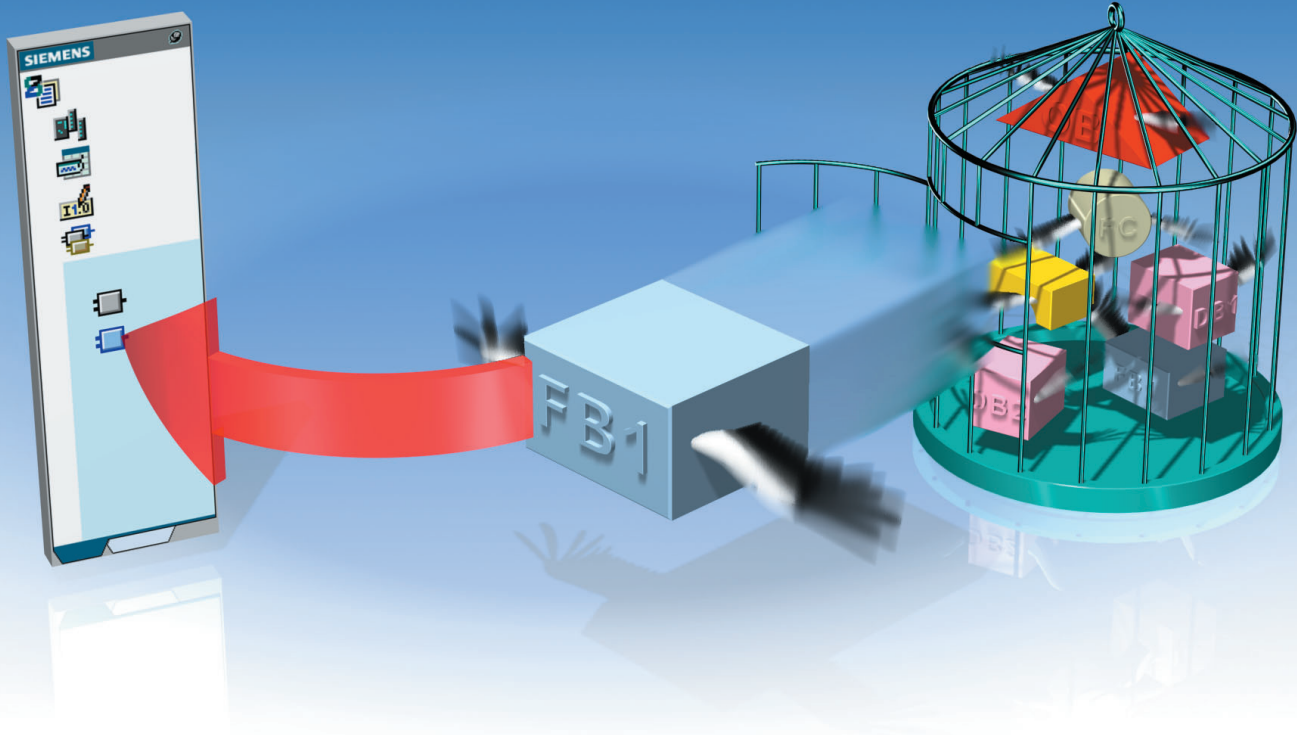


# 7

## Verwenden von Funktions- bausteinen



# Funktionsbaustein (FB) anlegen und öffnen



7.2

Funktionsbausteine werden verwendet, wenn zur Programmierung einer Funktion Zwischenergebnisse oder Betriebs-einstellungen und Betriebszustände bis zum nächsten Aufruf gespeichert werden müssen. Man nennt sie deshalb auch "Bausteine mit Gedächtnis."

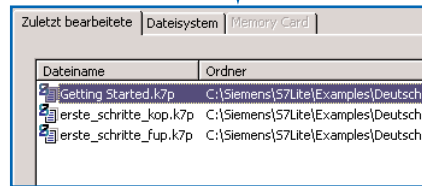
Sie programmieren in Ihrem Beispielprojekt den Funktionsbaustein FB 1 mit dem symbolischen Namen „Motor“. Verwenden Sie die Programmiersprache, in der Sie auch den OB 1 programmiert haben.

Für dieses Kapitel sollten Sie die Symboltabelle in Ihr Projekt „Getting Started“ kopiert haben (siehe Seite 5.5).

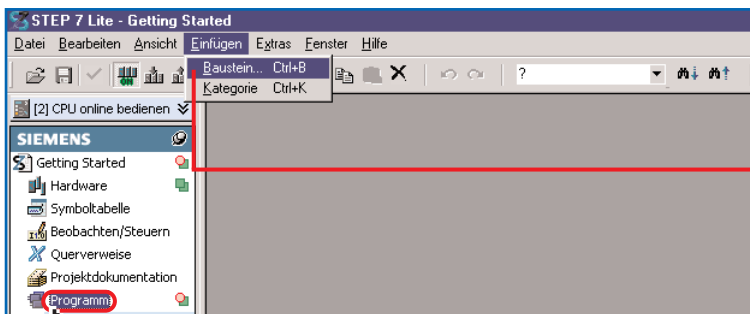


Das Dialogfeld für die Projektauswahl wird geöffnet.

1 Wenn notwendig öffnen Sie STEP 7 Lite.



2 Doppelklicken Sie im Dialogfeld **Projekt öffnen** auf Ihr Projekt "Getting started".

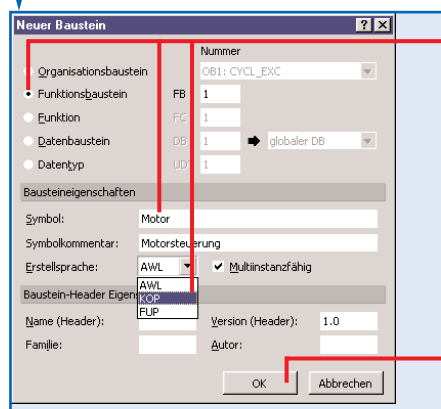


3 Klicken Sie im Projektfenster auf Programm.

4 Wählen Sie den Menübefehl **Einfügen > Baustein** oder über die rechte Maustaste den Kontextmenübefehl **Neu > Baustein**.

CLICK

Das Dialogfeld zum Anlegen neuer Bausteine wird geöffnet.

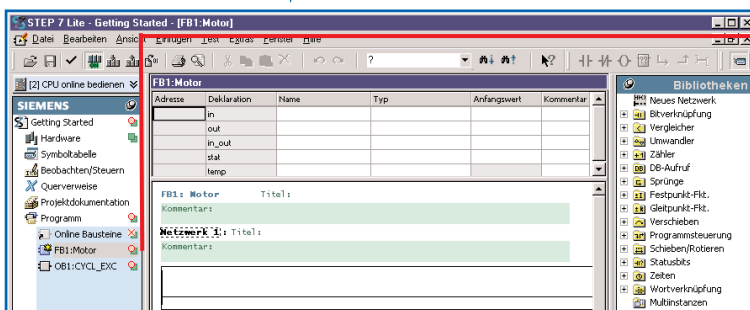


5 Markieren Sie **Funktionsbaustein**.

Wählen Sie im Feld **Erstellsprache** die Sprache, mit der Sie den Baustein programmieren wollen.

6 Bestätigen Sie mit **OK**.

Der Baustein wird eingefügt und sofort geöffnet.



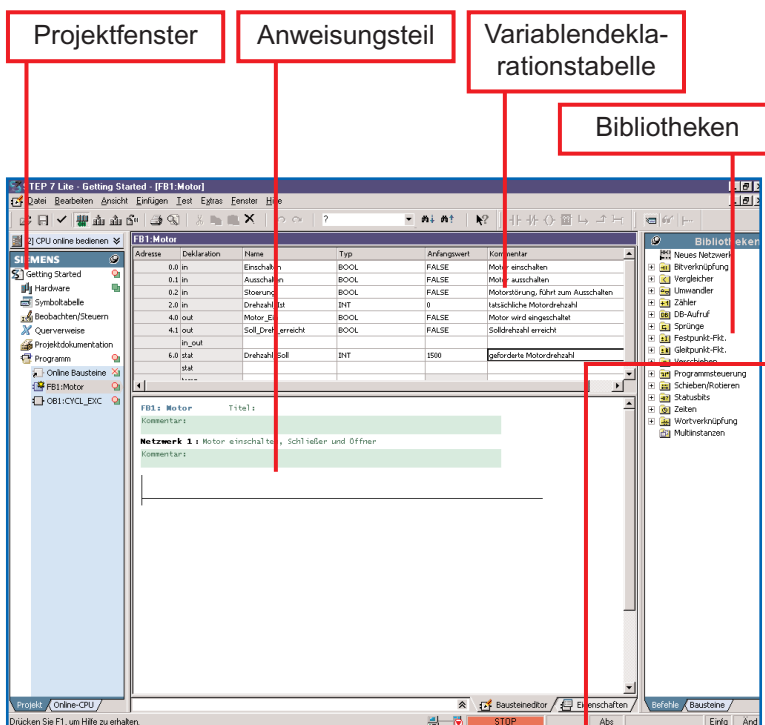
7 Der neue Baustein wird in das Projektfenster eingefügt und sofort geöffnet.

## Variablendeklarationstabelle ausfüllen

Wir zeigen Ihnen, wie Sie einen Funktionsbaustein programmieren, der mit je einem Datenbaustein einen Benzin- und einen Dieselmotor steuert und überwacht.

Alle "motorspezifischen" Signale werden dabei als Bausteinparameter zwischen Organisationsbaustein und Funktionsbaustein übergeben und müssen deshalb in der Variablendeklarationstabelle als Ein- und Ausgangsparameter (Deklaration "in" und "out") aufgeführt sein. Dadurch legen Sie die "Schnittstelle" für den Aufruf des FB im Programm fest.

Bevor Sie mit der Programmeingabe im Anweisungsteil beginnen, füllen Sie zuerst die Variablendeklarationstabelle des FBs aus.



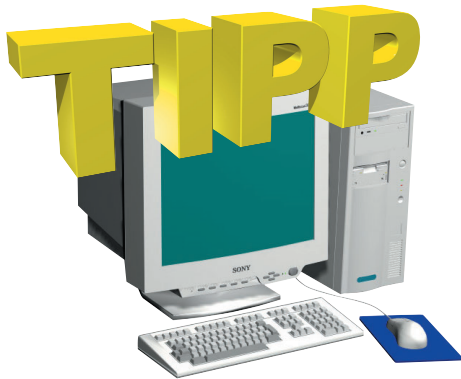
Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
0.0	in	Einschalten	BOOL	FALSE	Motor einschalten
0.1	in	Ausschalten	BOOL	FALSE	Motor ausschalten
0.2	in	Stoerung	BOOL	FALSE	Motorstörung, führt zum Ausschalten
2.0	in	Drehzahl_Ist	INT	0	tatsächliche Motordrehzahl
4.0	out	Motor_Ein	BOOL	FALSE	Motor wird eingeschaltet
4.1	out	Soll_Dreh_erreicht	BOOL	FALSE	Solldrehzahl erreicht
in_out					
6.0	stat	Drehzahl_Soll	INT	1500	geforderte Motordrehzahl
stat					

Geben Sie die Variablen aus der Abbildung in die Variablendeklarationstabelle ein.

Klicken Sie hierzu in eine Zelle, und übernehmen Sie den entsprechenden Namen sowie den Kommentar aus der Abbildung.

Den **Typ** wählen Sie aus dem Kontextmenü **Elementare Typen** über die rechte Maustaste aus.

Mit Return springt der Cursor in die nächste Spalte bzw. es wird eine neue Zeile eingefügt.



### 1. Editieren der Variablendeklarationstabelle

Für die Namen der Bausteinparameter in der Variablendeklarationstabelle sind nur Buchstaben, Ziffern und der Unterstrich zugelassen.

### 2. Hilfe zur Variablendeklarationstabelle?

Mehr Informationen über **F1 > Inhalt > Programmieren von Bausteinen > Erstellen von Codebausteinen** und **Editieren der Variablendeklarationstabelle**.

### 3. Tipps für die nachfolgenden Kapitel

In den folgenden Kapiteln programmieren Sie einen Ein- und Ausschaltvorgang und eine Drehzahlüberwachung.

Wann wird der Motor ein- und ausgeschaltet?

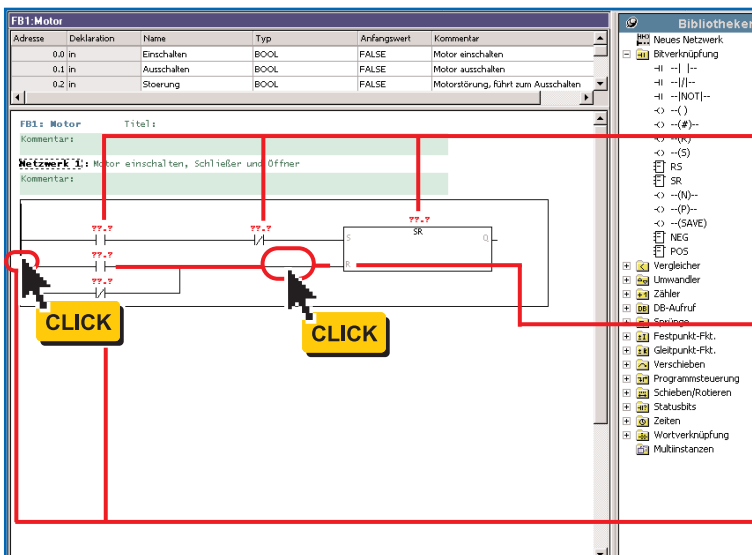
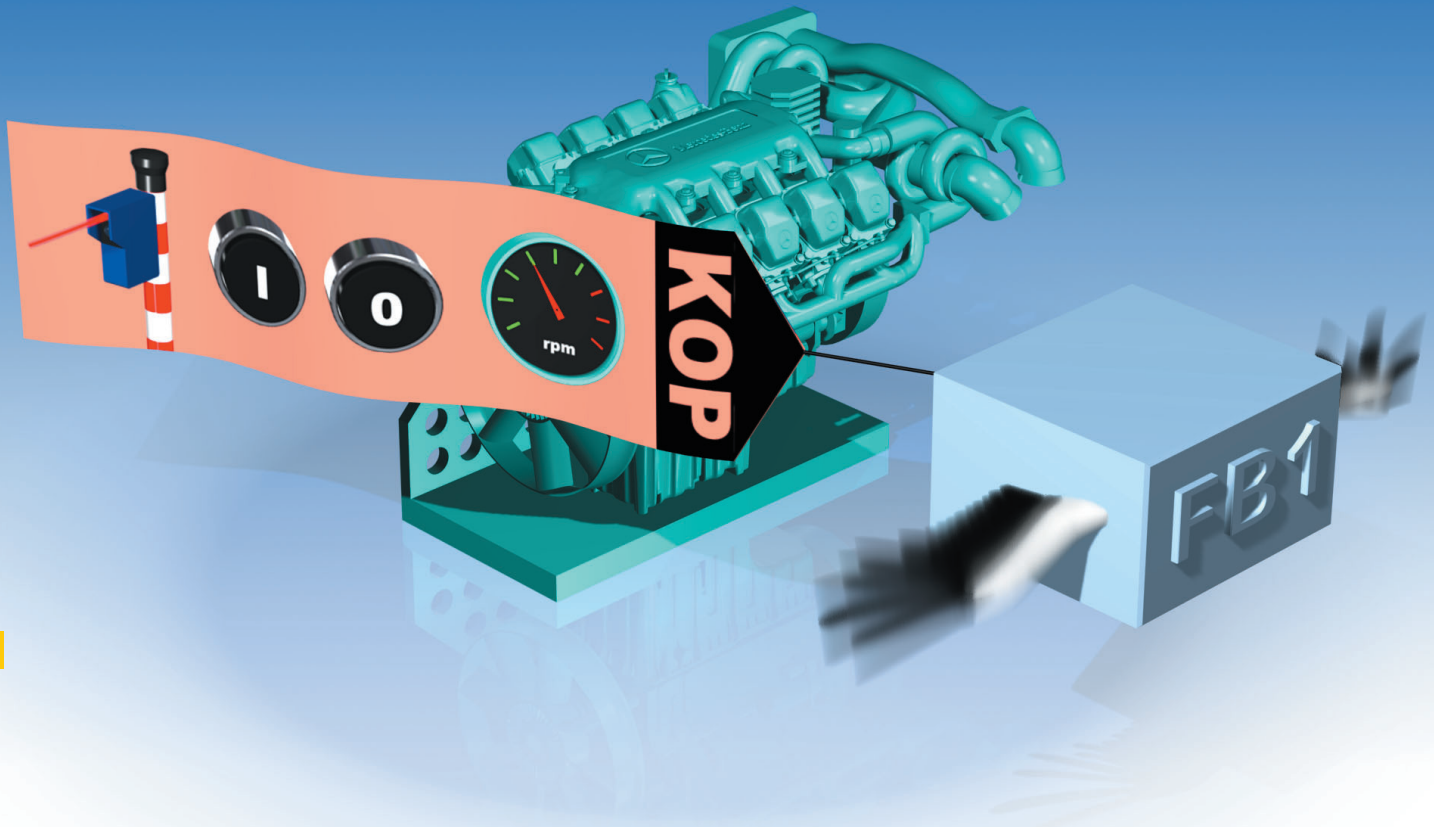
- Wenn die Variable #Einschalten den Signalzustand "1" und die Variable "Automatikbetrieb" den Signalzustand "0" führen, wird der Motor eingeschaltet.
- Wenn die Variable #Ausschalten den Signalzustand "1" oder die Variable #Stoerung den Signalzustand "0" führen, wird der Motor ausgeschaltet.

Wie überwacht der Vergleicher die Motordrehzahl?

- Über den Vergleicher werden die Variablen #Drehzahl\_Ist und #Drehzahl\_Soll verglichen und das Ergebnis der Variablen #Soll\_Drehzahl\_erreicht zugewiesen (Signalzustand 1).

Je nachdem, welche Programmiersprache Sie für Ihren OB 1 gewählt hatten, lesen Sie bitte ab jetzt weiter für KOP auf den Seiten 7.6 bis 7.7, für AWL auf den Seiten 7.8 bis 7.9, für FUP auf den Seiten 7.10 bis 7.11.

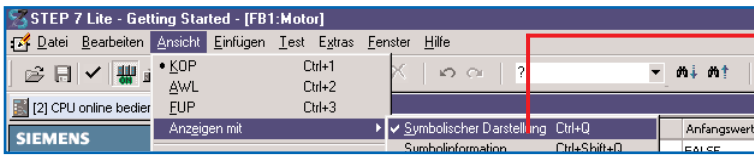
## FB in KOP programmieren



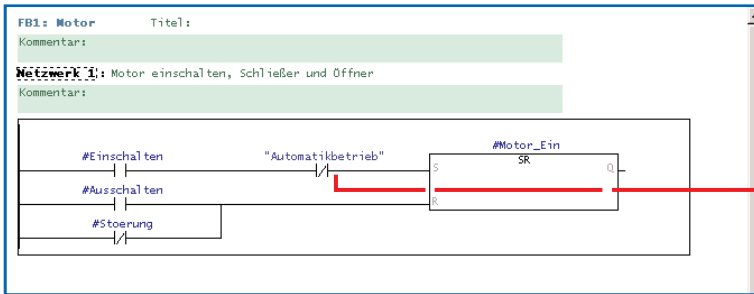
### Ein- und Ausschaltvorgang des Motors programmieren

- 1 Fügen Sie im Netzwerk 1 über **Bibliotheken > Befehle** einen Schließer, einen Öffner und ein SR-Element in Reihe ein.
- 2 Markieren Sie anschließend den Strompfad vor dem Eingang R und fügen Sie einen weiteren Schließer ein.
- 3 Markieren Sie die linke Stromschiene vor dem Schließer. Fügen Sie parallel zum Schließer einen Öffner ein.





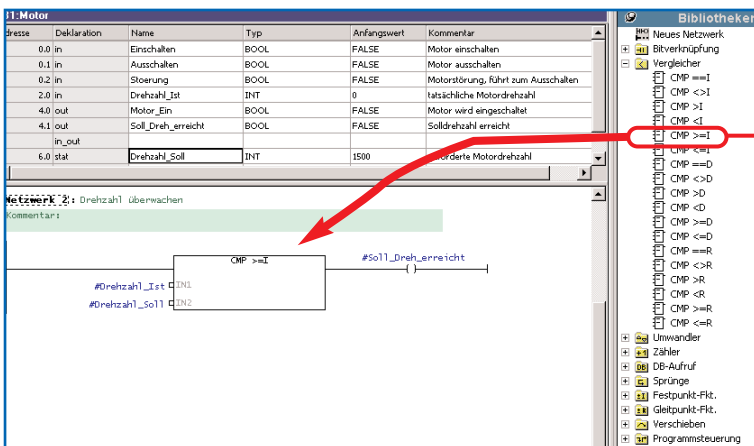
- 4 Prüfen Sie, ob die Symbolische Darstellung aktiviert ist.



- 5 Markieren Sie alle ???.?, und tragen Sie die entsprechenden Namen der Variablen Deklarationstabelle ein (# wird automatisch vergeben).

- 6 Tragen Sie für den Öffner der Reihenschaltung den symbolischen Namen "Automatikbetrieb" ein.

## Drehzahlüberwachung programmieren

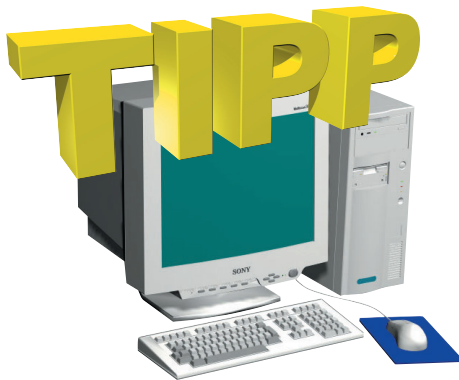


- 7 Fügen Sie ein neues Netzwerk ein und markieren Sie den Strompfad.

- 8 Navigieren Sie in der Befehlsübersicht zum Vergleich und fügen Sie **CMP >=I** ein. Fügen Sie außerdem im Strompfad eine Spule ein.

- 9 Markieren Sie wieder die Fragezeichen, und beschriften Sie die Spule und den Vergleich mit den Namen aus der Variablen Deklarationstabelle.

Speichern Sie Ihre Eingaben.

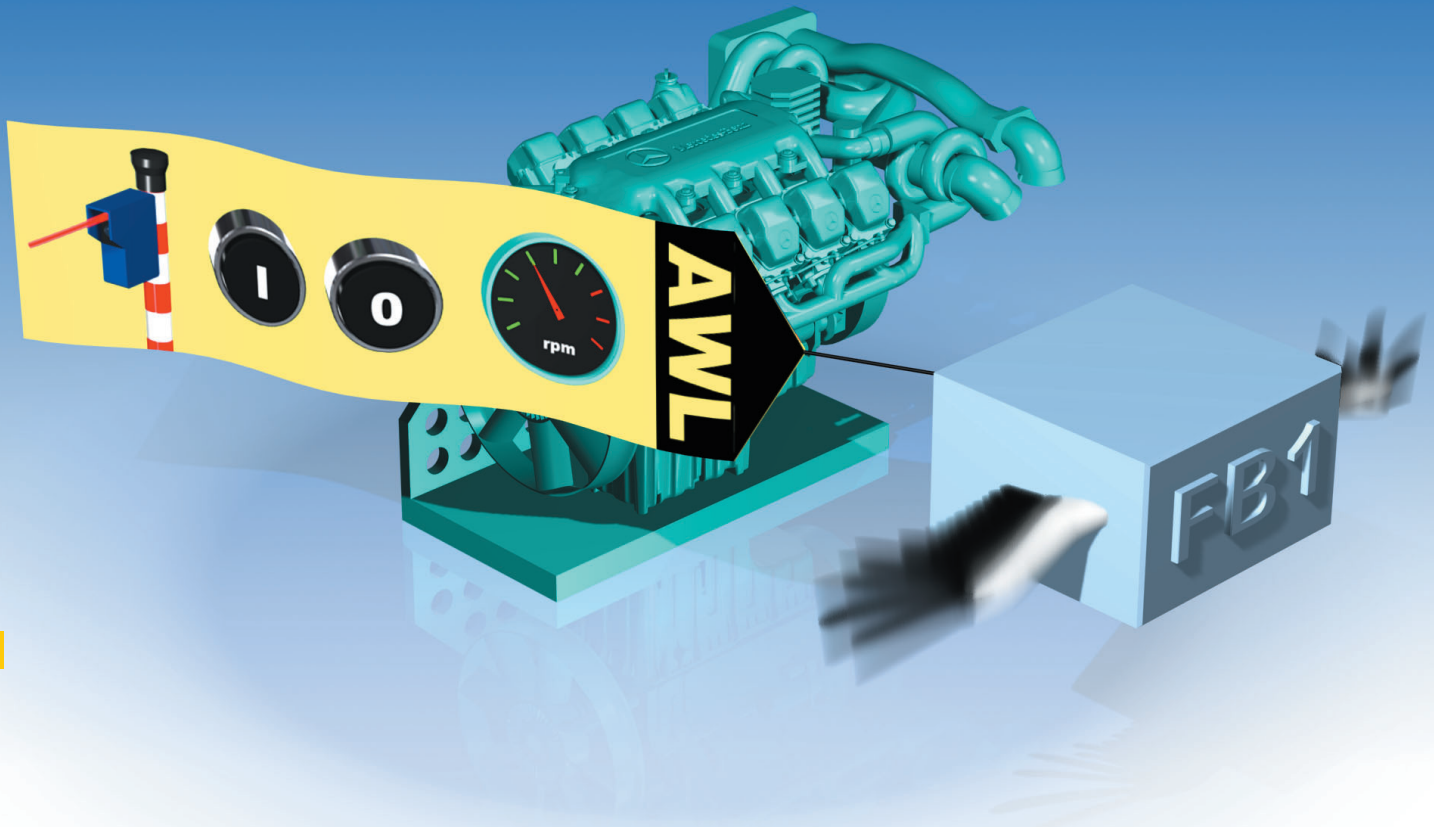


Bausteinlokale Variablen werden mit # gekennzeichnet und sind nur im Baustein gültig. Globale Variablen stehen in Anführungszeichen. Sie werden in der Symboltabelle definiert und sind im gesamten Programm gültig. Der Signalzustand "Automatikbetrieb" wird im OB1 (Netzwerk 3, vgl. Seite 6-10) durch ein anderes SR-Element festgelegt und jetzt im FB1 abgefragt.

Mehr Informationen über **F1 > Inhalt > Programmieren von Bausteinen > Erstellen von Codebausteinen** und **Editieren von KOP-Anweisungen**.

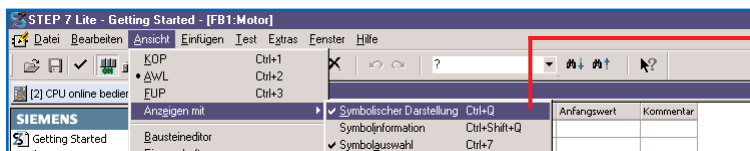


## FB in AWL programmieren



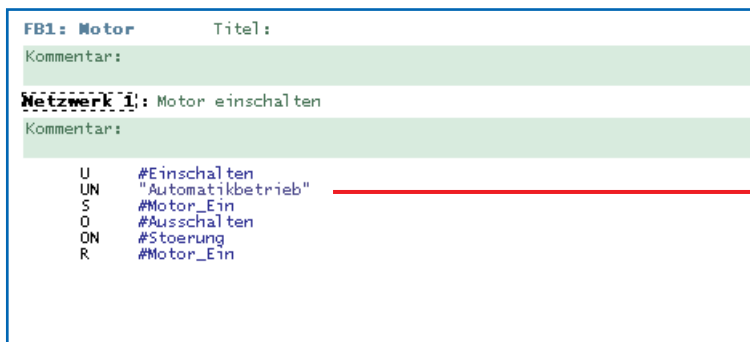
7.8

### Ein- und Ausschaltvorgang des Motors programmieren



1

Prüfen Sie, ob die Symbolische Darstellung aktiviert ist.



2

Tragen Sie im Netzwerk 1 diese AWL-Anweisungen ein.

## Drehzahlüberwachung programmieren

```
FB1: Motor      Titel:
Kommentar:

Netzwerk 1: Motor einschalten
Kommentar:

U      #Einschalten
UN     "Automatikbetrieb"
S      #Motor_Ein
O      #Ausschalten
ON     #Steuerung
R      #Motor_Ein

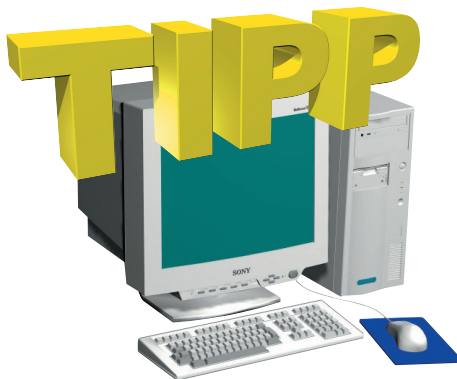
Netzwerk 2: Drehzahl überwachen
Kommentar:

L      #Drehzahl_Ist
L      #Drehzahl_Soll
>=I
=      #Soll_Drehzahl_erreicht
```

3

Fügen Sie ein neues Netzwerk ein, und tragen Sie diese AWL-Anweisungen ein.

Speichern Sie Ihre Eingaben.

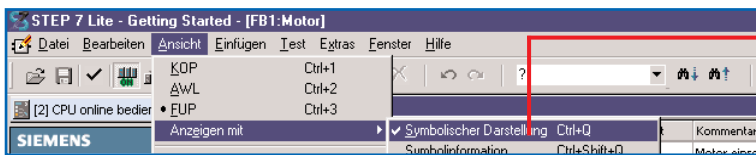


Bausteinlokale Variablen werden mit # gekennzeichnet und sind nur im Baustein gültig. Globale Variablen stehen in Anführungszeichen. Sie werden in der Symboltabelle definiert und sind im gesamten Programm gültig. Der Signalzustand "Automatikbetrieb" wird im OB1 (Netzwerk 3, vgl. Seite 6-16) durch ein anderes SR-Element festgelegt und jetzt im FB1 abgefragt.

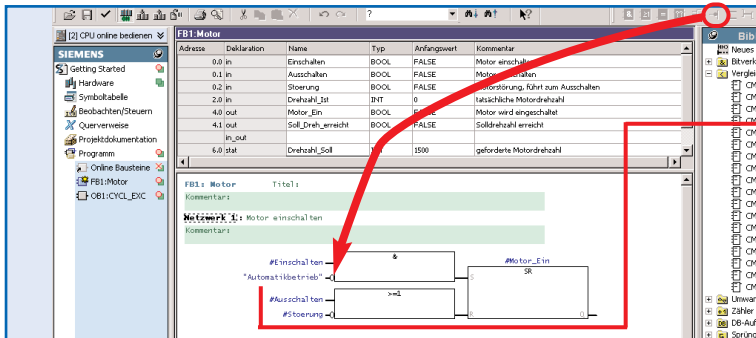
Mehr Informationen über **F1 > Inhalt > Programmieren von Bausteinen > Erstellen von Codebausteinen** und **Editieren von AWL-Anweisungen**.



Den Eingang S (Setzen) belegen Sie mit einer UND-Box. Den Eingang R (Rücksetzen) belegen Sie mit einer ODER-Box.



4 Prüfen Sie, ob die Symbolische Darstellung aktiviert ist.



5 Markieren Sie alle ???.?, und tragen Sie die entsprechenden Namen der Variablen deklaratstabelle ein (# wird automatisch vergeben).

6 Adressieren Sie einen Eingang der UND-Funktion mit dem symbolischen Namen "Automatikbetrieb". Negieren Sie noch die Eingänge "Automatikbetrieb" und #Stoerung mit dem entsprechenden Symbol aus der Funktionsleiste.

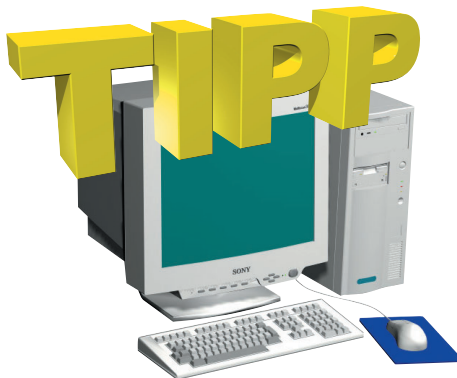
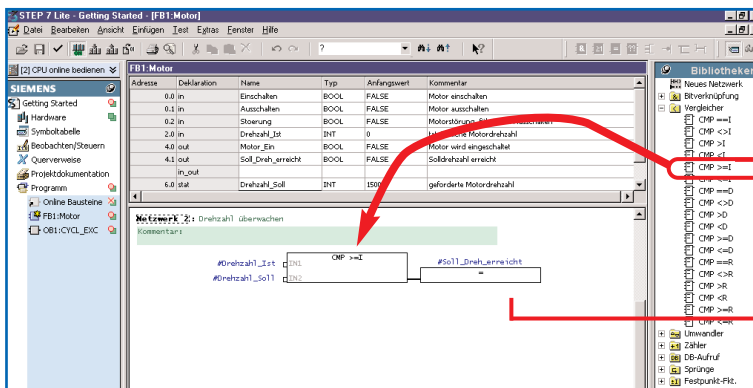
## Drehzahlüberwachung programmieren

Fügen Sie ein neues Netzwerk ein und markieren Sie den Strompfad.

Navigieren Sie in der Befehlsübersicht zum Vergleich, und fügen Sie ein **CMP>=I** ein. Adressieren Sie die Eingänge mit den Namen aus der Variablen deklaratstabelle.

8 Fügen Sie eine Ausgangszuweisung an den Vergleich an, und adressieren Sie die Zuweisung mit den Namen aus der Variablen deklaratstabelle.

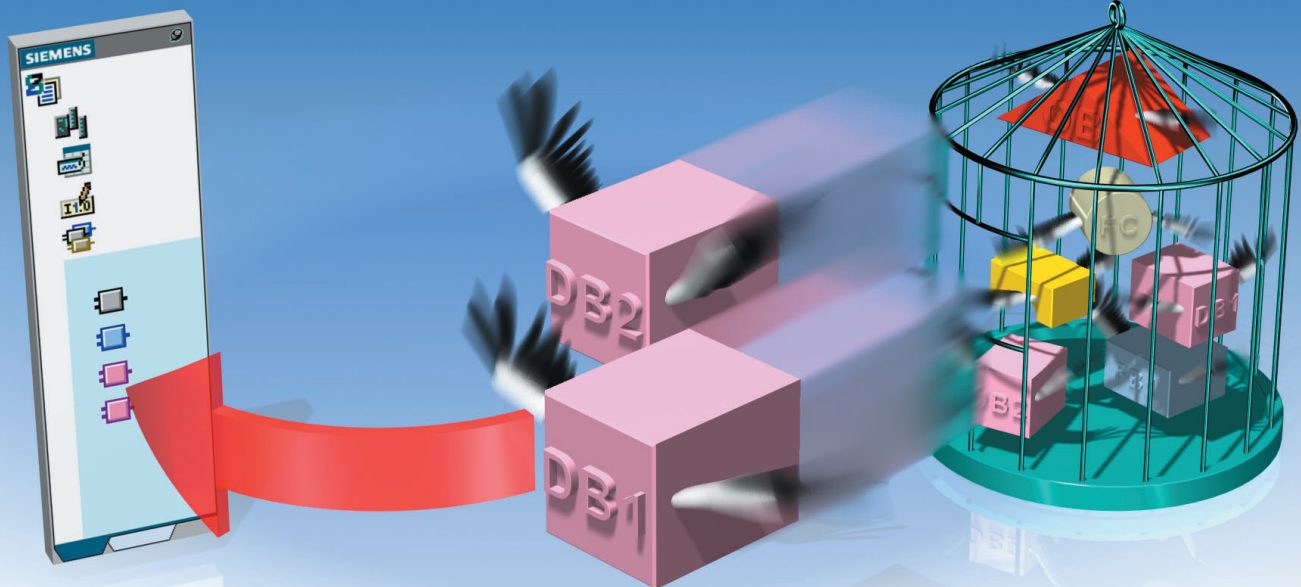
Speichern Sie Ihre Eingaben.



Bausteinlokale Variablen werden mit # gekennzeichnet und sind nur im Baustein gültig. Globale Variablen stehen in Anführungszeichen. Sie werden in der Symboltabelle definiert und sind im gesamten Programm gültig. Der Signalzustand "Automatikbetrieb" wird im OB1 (Netzwerk 3, vgl. Seite 6-22) durch ein anderes SR-Element festgelegt und jetzt im FB1 abgefragt.

Mehr Informationen über **F1 > Inhalt > Programmieren von Bausteinen > Erstellen von Codebausteinen** und **Editieren von FUP-Anweisungen**.

# Instanz-Datenbausteine erzeugen und Aktualwerte ändern



7.12

### Datenbaustein einfügen

Um später den Aufruf (CALL) des FB1 im OB1 programmieren zu können, müssen Sie noch den zugehörigen Datenbaustein erzeugen.

Der FB soll einen Benzin- bzw. Dieselmotor steuern und überwachen. Die unterschiedlichen Soll-Drehzahlen der Motoren werden dabei in zwei separaten DBs hinterlegt, indem jeweils der Aktualwert (#Drehzahl\_ Soll) geändert wird.

Indem Sie den Funktionsbaustein nur einmal zentral programmieren, reduzieren Sie Ihren Programmieraufwand.

**Neuer Baustein**

Organisationsbaustein  
Funktionsbaustein  
Funktion  
Datenbaustein

Numerus  
OB1: CYCL\_EXC  
FB 2  
FC 1  
DB 1 → FB1: Motor  
UDT 1

Datentyp

Bausteineigenschaften

Symbol: Benzin  
Symbolkommentar: Daten für Benzinmotor  
Erstelsprache: DB ☒ Multistanzenfähig

Baustein-Header Eigenschaften

Name (Header):  
Version (Header): 1.0  
Familie:  
Autor:

OK Abbrechen

- 1 Klicken Sie mit der rechten Maustaste in das Projektfenster und wählen Sie aus dem Kontextmenü **Neu > Baustein**. Das Dialogfeld **Neuer Baustein** wird geöffnet.
- 2 Markieren Sie hier **Datenbaustein** und tragen Sie als zugeordneten FB den FB 1 ein (wie im Bild dargestellt).
- 3 Bestätigen Sie mit **OK**. Der DB1 wird im Projekt "Getting Started" eingefügt und sofort geöffnet.

Der Baustein wird eingefügt.

**STEP 7 Lite - Getting Started - [DB1: Benzin]**

Ansicht: Eingaben Test Extras Fenster Hilfe

SIEMENS

Getting Started

Hardware

Symboltabelle

Querverweise

Projektdokumentation

Programme

Online Bausteine

DB1: Benzin

Beobachten/Steuern

Name	Typ	Anfangswert	Aktualwert	Kommentar
inschalten	BOOL	FALSE	FALSE	Motor einschalten
auschalten	BOOL	FALSE	FALSE	Motor ausschalten
Störung	BOOL	FALSE	FALSE	Motorstörung, führt zum Aus
rehzahl_ist	INT	0	0	tatsächliche Motordrehzahl
Motor_Ein	BOOL	FALSE	FALSE	Motor wird eingeschaltet
oll_Dreh_reicht	BOOL	FALSE	FALSE	Soll-Drehzahl erreicht
rehzahl_Soll	INT	1500	1500	geforderte Motordrehzahl

- 4 Aktivieren Sie **Datensicht**. Nur in dieser Sicht können Sie den DB ändern.
- 5 Überzeugen Sie sich, dass für den Benzinmotor in der Spalte **Aktualwert** "1500" eingetragen ist.

Der Baustein wird eingefügt.

**STEP 7 Lite - Getting Started - [DB2: Diesel]**

Ansicht: Eingaben Test Extras Fenster Hilfe

SIEMENS

Getting Started

Hardware

Symboltabelle

Querverweise

Projektdokumentation

Programme

Online Bausteine

DB2: Diesel

Beobachten/Steuern

Name	Typ	Anfangswert	Aktualwert	Kommentar
inschalten	BOOL	FALSE	FALSE	Motor einschalten
auschalten	BOOL	FALSE	FALSE	Motor ausschalten
Störung	BOOL	FALSE	FALSE	Motorstörung, führt zum Aus
Drehzahl_ist	INT	0	0	tatsächliche Motordrehzahl
Motor_Ein	BOOL	FALSE	FALSE	Motor wird eingeschaltet
Soll_Dreh_reicht	BOOL	FALSE	FALSE	Soll-Drehzahl erreicht
Drehzahl_Soll	INT	1500	1200	geforderte Motordrehzahl

- 6 Fügen Sie analog einen DB "Diesel" ein.
- 7 Tragen Sie für den Dieselmotor in Spalte **Aktualwert** "1200" ein. Speichern Sie Ihre Eingaben.

7.13

Um nun den Aufruf des FBs im OB1 zu programmieren, lesen Sie bitte entsprechend der bisher von Ihnen verwendeten Programmiersprache in einem der folgenden Kapitel zu KOP, FUP oder AWL weiter.



Mit der Einstellung der Aktualwerte sind die Vorbereitungen abgeschlossen, um mit nur einem Funktionsbaustein zwei Motoren zu steuern. Um weitere Motoren zu steuern, müssten lediglich weitere Datenbausteine erzeugt werden.

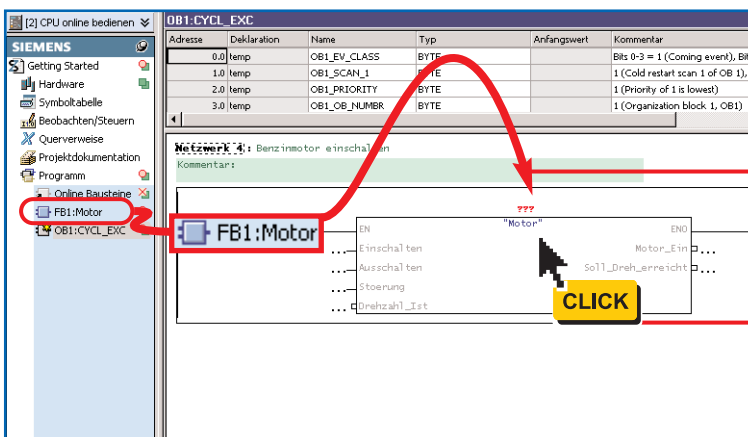
Mehr Informationen finden Sie über **F1 > Inhalt > Programmieren von Bausteinen > Erstellen von Datenbausteinen**.



# Bausteinaufruf in KOP programmieren

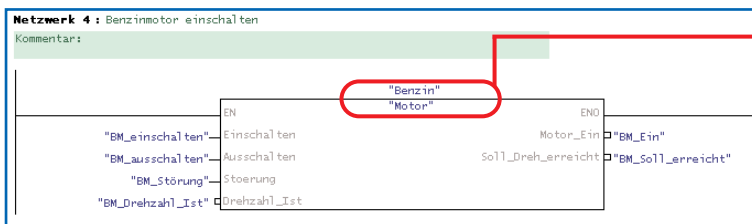


Die gesamte Programmierung eines FBs ist ohne Aufruf im OB1 unwirksam. Pro Aufruf des FBs wird je ein Datenbaustein benutzt und somit beide Motoren gesteuert.



- 1 Öffnen Sie den OB1 und fügen Sie Netzwerk 4 ein.
- 2 Ziehen Sie den **FB 1** aus dem Projektfenster per Drag&Drop in Netzwerk 4 hinein. Alle motorspezifischen Variablen werden angezeigt.
- 3 Klicken Sie auf **???**. Die Symbolauswahlliste wird geöffnet.





4

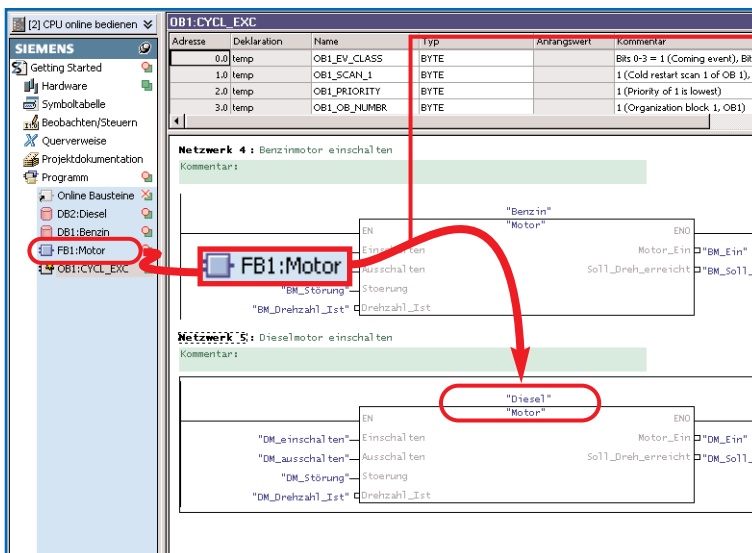
Wählen Sie aus der Symbolauswahl-  
liste den Datenbaustein "Benzin".

5

Adressieren Sie analog alle weiteren  
Parameter des Funktionsbausteins mit  
den entsprechenden symbolischen  
Namen.

Die motorspezifischen Ein- und Aus-  
gangsvariablen (Deklaration "in" und  
"out") werden im FB "Motor" angezeigt.  
Den Variablen wird je ein Signal  
"BM\_xxx" für den Benzinmotor zugewie-  
sen.

Der Aufruf für den Diesel-  
motor fehlt noch.



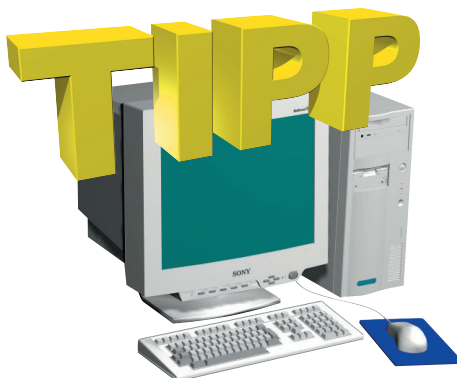
6

Fügen Sie Netzwerk 5 ein, ziehen Sie  
den FB 1 erneut aus dem Projekt-  
fenster in das Netzwerk und program-  
mieren Sie analog den Aufruf des  
Funktionsbausteins "Motor (FB1)" mit  
dem Datenbaustein "Diesel" (DB 2).

Den Variablen wird je ein Signal  
"DM\_xxx" für den Dieselmotor zuge-  
wiesen.

Speichern Sie Ihre Eingaben und  
schließen Sie den Baustein.

7.15



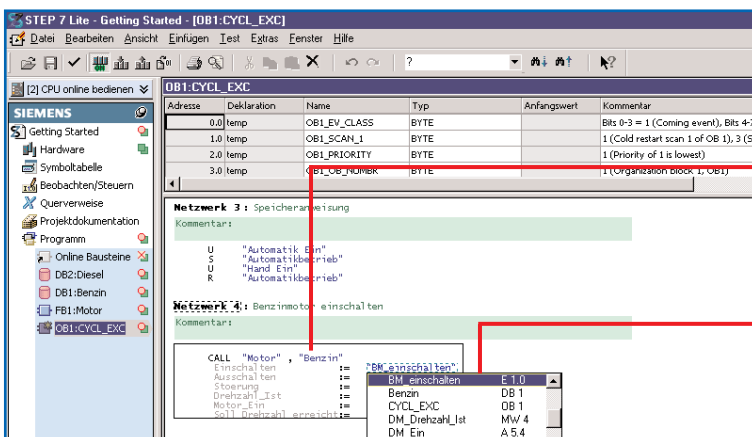
Wenn Sie Programmstrukturen mit OBs, FBs und  
DBs anlegen, so müssen Sie den Aufruf eines  
untergeordneten Bausteins (z.B. FB1) im überge-  
ordneten Baustein (z.B. OB1) programmieren. Die  
Vorgehensweise ist dabei immer identisch. Sie  
können in der Symboltabelle auch den verschie-  
denen Bausteinen symbolische Namen geben  
(z.B. FB1 hat den Namen "Motor" und DB1 den  
Namen "Benzin").

Sie können jederzeit die programmierten Bauste-  
ne ausdrucken mit **Datei > Drucken**. Mehr Infor-  
mationen zum Drucken finden Sie über **F1 >**  
**Inhalt > Drucken der Projektdokumentation**.

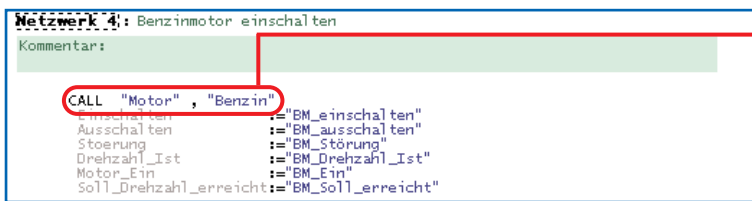
## Bausteinaufruf in AWL programmieren



Die gesamte Programmierung eines FBs ist ohne Aufruf im OB1 unwirksam. Pro Aufruf des FBs wird je ein Datenbaustein benutzt und somit beide Motoren gesteuert.



- 1 Öffnen Sie den OB1 und fügen Sie Netzwerk 4 ein.
- 2 Schreiben Sie in den Anweisungsteil **CALL "Motor", "Benzin"** und drücken Sie Return. Alle motorspezifischen Variablen werden angezeigt.
- 3 Wählen Sie nach einem Rechtsklick hinter := den Kontextmenübefehl Symbol einfügen. Die Symbolauswahlliste wird geöffnet.

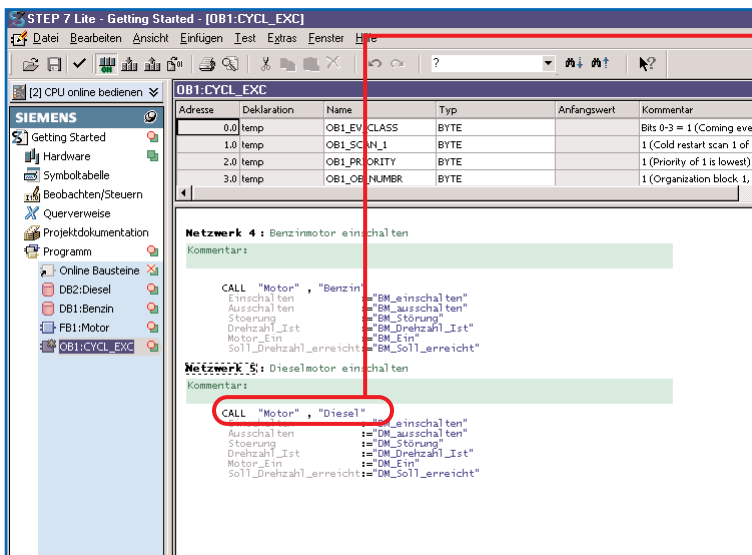


4

Adressieren Sie analog alle weiteren Parameter des Funktionsbausteins mit den entsprechenden symbolischen Namen.

Die motorspezifischen Ein- und Ausgangsvariablen (Deklaration "in" und "out") werden im FB "Motor" angezeigt. Den Variablen wird je ein Signal "BM\_xxx" für den Benzinmotor zugewiesen.

Der Aufruf für den Dieselmotor fehlt noch.

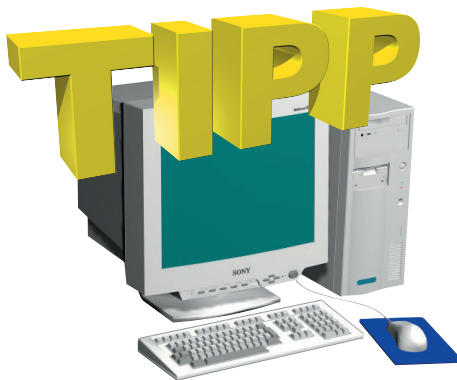


6

Fügen Sie Netzwerk 5 ein und programmieren Sie analog den Aufruf des Funktionsbausteins "Motor (FB1) mit dem Datenbaustein "Diesel" (DB 2).

Den Variablen wird je ein Signal "DM\_xxx" für den Dieselmotor zugewiesen.

Speichern Sie Ihre Eingaben und schließen Sie den Baustein.



Wenn Sie Programmstrukturen mit OBs, FBs und DBs anlegen, so müssen Sie den Aufruf eines untergeordneten Bausteins (z.B. FB1) im übergeordneten Baustein (z.B. OB1) programmieren. Die Vorgehensweise ist dabei immer identisch. Sie können in der Symboltabelle auch den verschiedenen Bausteinen symbolische Namen geben (z.B. FB1 hat den Namen "Motor" und DB1 den Namen "Benzin").

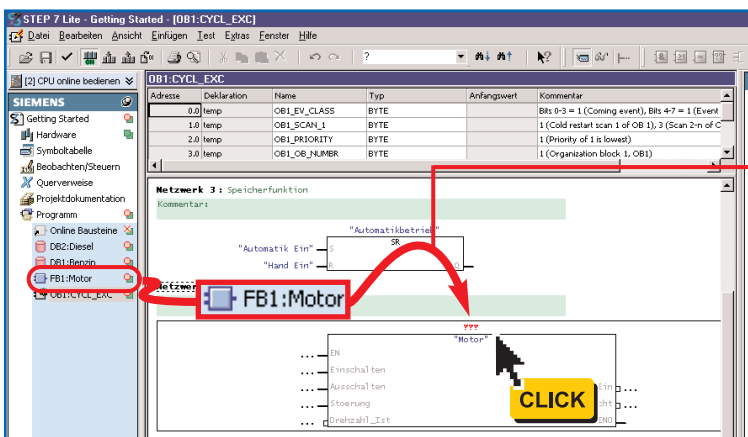
Sie können jederzeit die programmierten Bausteine ausdrucken mit **Datei > Drucken**. Mehr Informationen zum Drucken finden Sie über **F1 > Inhalt > Drucken der Projektdokumentation**.

## Bausteinaufruf in FUP programmieren

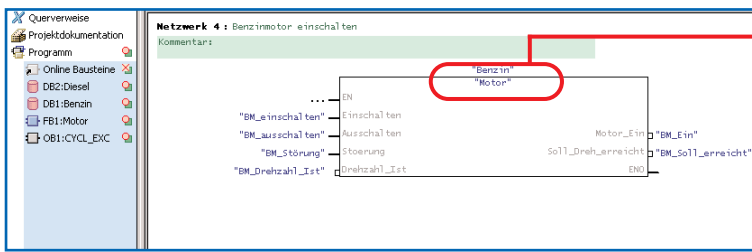


7.18

Die gesamte Programmierung eines FBs ist ohne Aufruf im OB1 unwirksam. Pro Aufruf des FBs wird je ein Datenbaustein benutzt und somit beide Motoren gesteuert.



- 1 Öffnen Sie den OB1 und fügen Sie Netzwerk 4 ein.
- 2 Ziehen Sie den **FB 1** aus dem Projektfenster per Drag&Drop in Netzwerk 4 hinein. Alle motorspezifischen Variablen werden angezeigt.
- 3 Klicken Sie auf **???**. Die Symbolauswahlliste wird geöffnet.



4

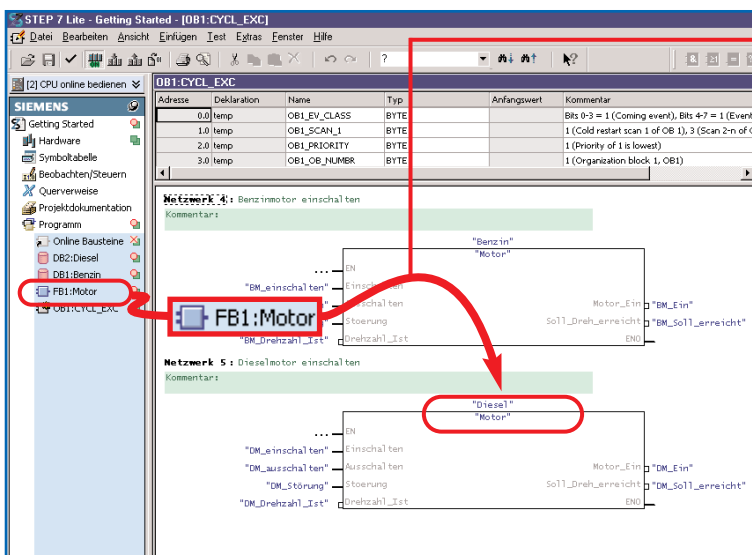
Wählen Sie aus der Symbolauswahl-  
liste den Datenbaustein "Benzin".

5

Adressieren Sie analog alle weiteren  
Parameter des Funktionsbausteins mit  
den entsprechenden symbolischen  
Namen.

Die motorspezifischen Ein- und Aus-  
gangsvariablen (Deklaration "in" und  
"out") werden im FB "Motor" angezeigt.  
Den Variablen wird je ein Signal  
"BM\_xxx" für den Benzinmotor zugewie-  
sen.

Der Aufruf für den Diesel-  
motor fehlt noch.



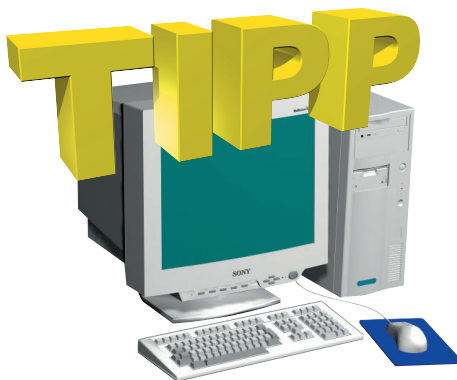
6

Fügen Sie Netzwerk 5 ein, ziehen Sie  
den FB 1 erneut aus dem Projekt-  
fenster in das Netzwerk und program-  
mieren Sie analog den Aufruf des  
Funktionsbausteins "Motor (FB1)" mit  
dem Datenbaustein "Diesel" (DB 2).

Den Variablen wird je ein Signal  
"DM\_xxx" für den Dieselmotor zugewie-  
sen.

Speichern Sie Ihre Eingaben und  
schließen Sie den Baustein.

7.19



Wenn Sie Programmstrukturen mit OBs, FBs und DBs anlegen, so müssen Sie den Aufruf eines untergeordneten Bausteins (z.B. FB1) im übergeordneten Baustein (z.B. OB1) programmieren. Die Vorgehensweise ist dabei immer identisch. Sie können in der Symboltabelle auch den verschiedenen Bausteinen symbolische Namen geben (z.B. FB1 hat den Namen "Motor" und DB1 den Namen "Benzin").

Sie können jederzeit die programmierten Bausteine ausdrucken mit **Datei > Drucken**. Mehr Informationen zum Drucken finden Sie über **F1 > Inhalt > Drucken der Projektdokumentation**.





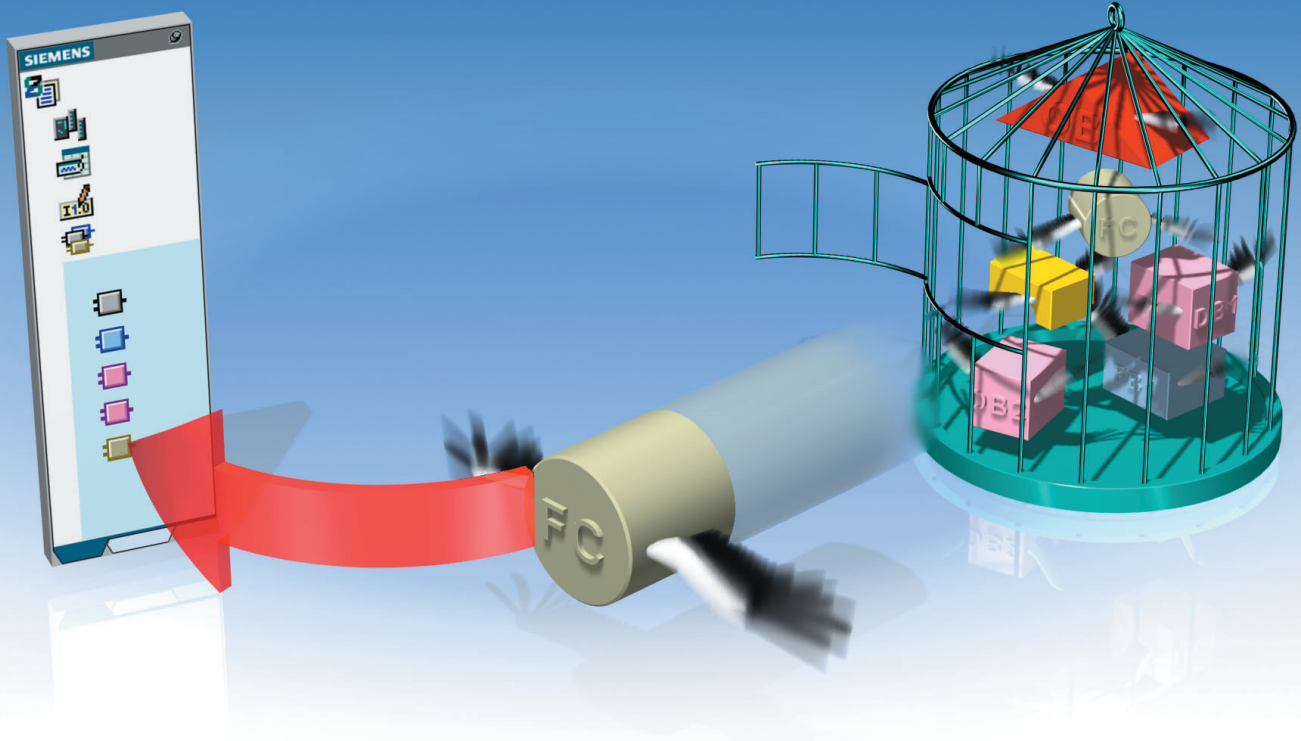
# 8

## Verwenden von Funktionen





# Funktion (FC) anlegen und öffnen



8.2

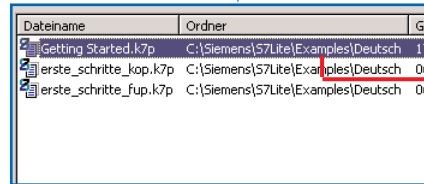
Funktionen werden verwendet, wenn zur Programmierung einer Funktion keine Zwischenergebnisse oder Betriebseinstellungen und Betriebszustände bis zum nächsten Aufruf gespeichert werden müssen. Man nennt sie deshalb auch "Bausteine ohne Gedächtnis."

Für dieses Kapitel sollten Sie die Symboltabelle in Ihr Projekt "Getting Started" kopiert haben (siehe Seite 5.5).

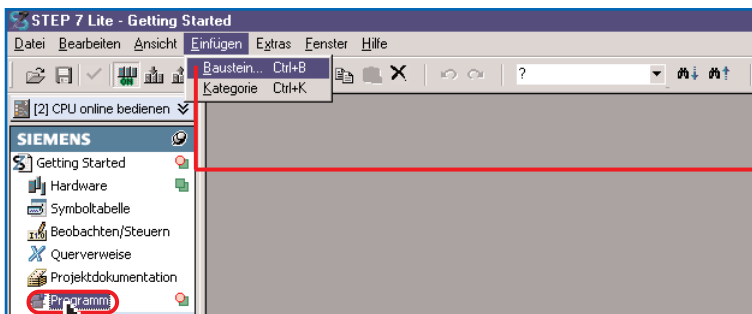


Das Dialogfeld für die Projektauswahl wird geöffnet.

1 Wenn notwendig, öffnen Sie STEP 7 Lite.



2 Klicken Sie im Dialogfeld **Projekt öffnen** auf Ihr Projekt "Getting Started".

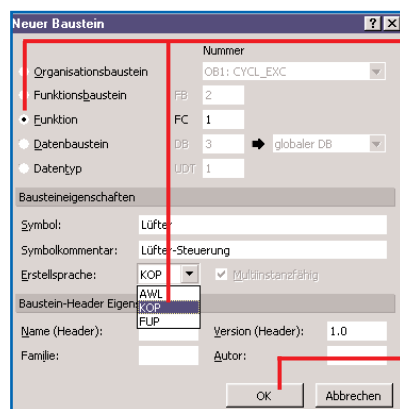


3 Klicken Sie im Projektfenster auf Programm.

4 Wählen Sie den Menübefehl **Einfügen > Baustein** oder über die rechte Maustaste den Kontextmenübefehl **Neu > Baustein**.

CLICK

Das Dialogfeld zum Anlegen neuer Bausteine wird geöffnet.

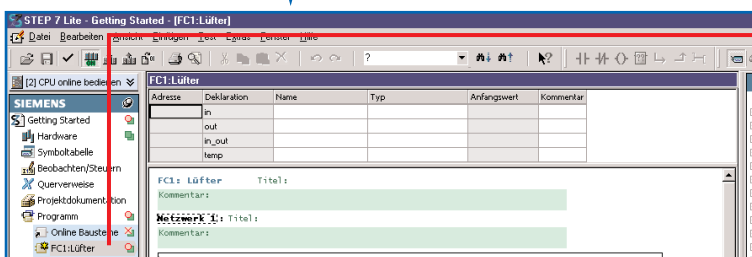


5 Markieren Sie **Funktion**.

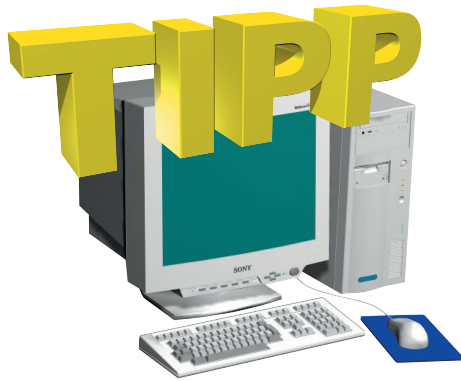
Wählen Sie im Feld **Erstellsprache** die Sprache, mit der Sie auch Ihr Projekt "Getting Started" erstellt haben.

6 Bestätigen Sie mit **OK**.

Der Baustein wird eingefügt und sofort geöffnet.

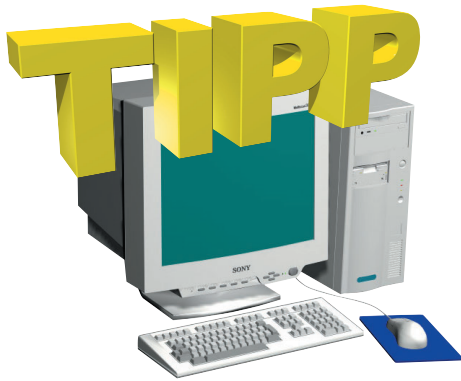


7 Der neue Baustein wird in das Projektfenster eingefügt und sofort geöffnet.



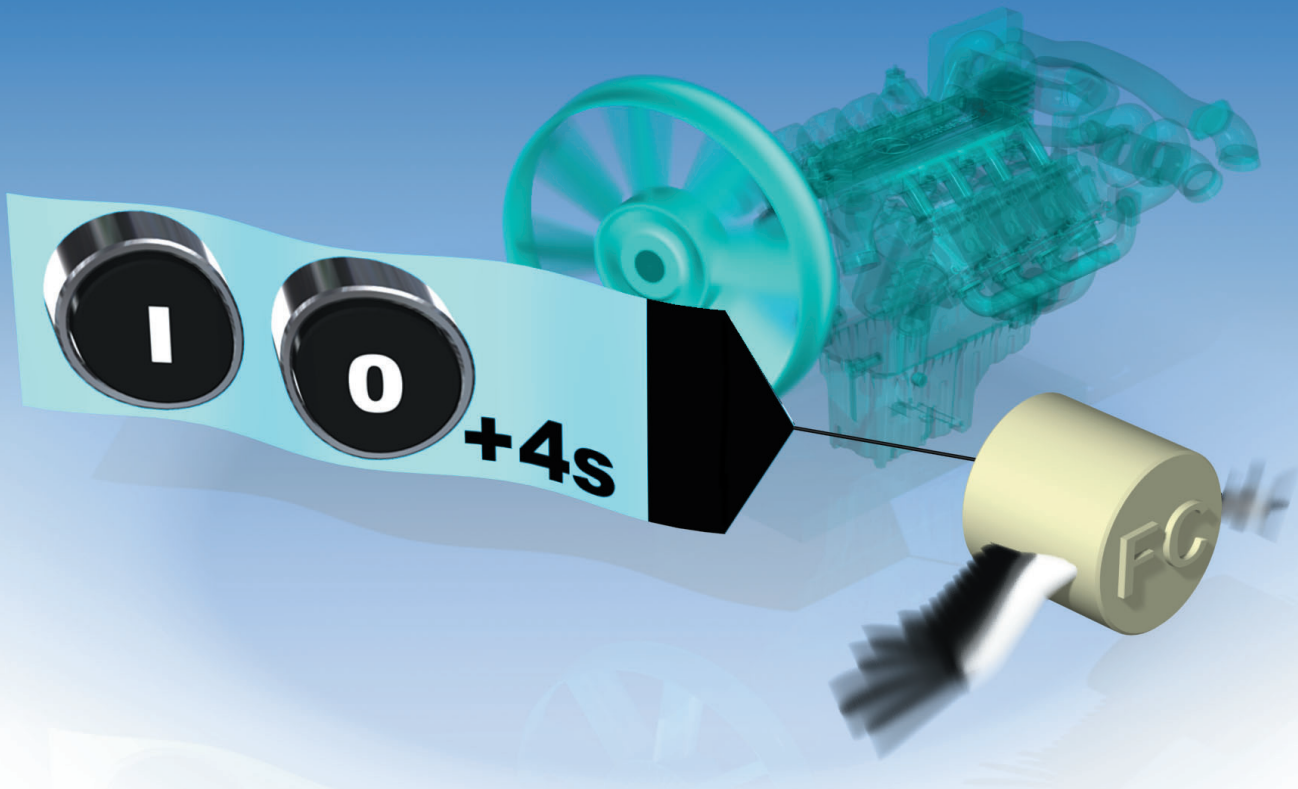
Innerhalb einer Funktion können im Gegensatz zum Funktionsbaustein keine statischen Daten in der Variablendeklarationstabelle definiert werden.

Für das Programmieren der Funktion können Sie wie gewohnt auf die symbolischen Namen aus der Symboltabelle zurückgreifen.



Mehr Information über **F1 > Inhalt > Grundlagen zum Entwerfen eines Programms > Bausteine im Anwenderprogramm.**

# Funktion programmieren



8.6

Im folgenden Beispiel programmieren Sie eine Zeitfunktion. Die Zeitfunktion bewirkt, dass sich beim Einschalten eines Motors gleichzeitig ein Lüfter einschaltet, der nach dem Ausschalten des Motors noch vier Sekunden nachläuft (Ausschaltverzögerung).

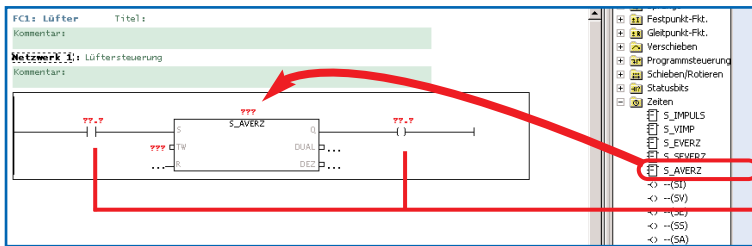
Deklaration	Name	Typ	Anfangswert	Kommentar
in	Motor_Ein	BOOL		Signal für das Einschalten des Motors
in	Zeitfunktion	TIMER		verwendete Zeitfunktion für die Ausschaltverzögerung
out	Luefter_Ein	BOOL		Signal für das Einschalten des Lüfters
in_out				
temp				

1

## Variablendeklarationstabelle

Analog zum Funktionsbaustein tragen Sie zunächst die Ein- und Ausgangsparameter der Funktion (Deklaration "in" und "out") in der Variablendeklarationstabelle ein.

Den Typ TIMER wählen Sie aus dem Kontextmenü **Parametertypen** über die rechte Maustaste aus.



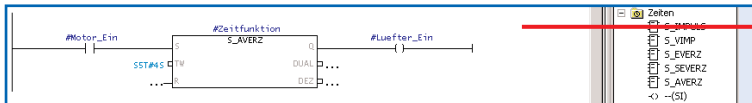
2

## Zeitfunktion programmieren in KOP

Fügen Sie aus **Bibliotheken > Befehle > Zeiten** das Element **S\_AVERZ** in das Netzwerk 1 ein.

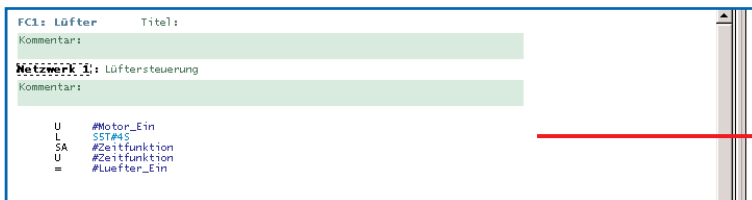
3

Fügen Sie zusätzlich einen Schließer und eine Spule ein.



4

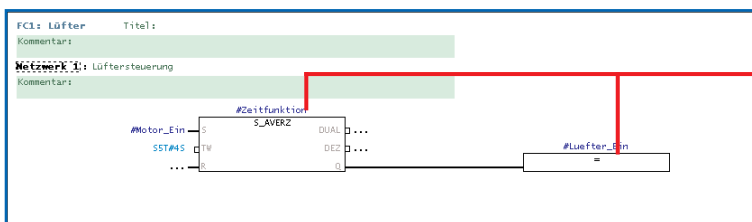
Tragen Sie für die **???** die Namen aus der Variablendeklarationstabelle ein. Sie werden automatisch mit # gekennzeichnet. Geben Sie für den Zeitwert TW S5T#4s ein. Speichern Sie Ihre Eingaben. Schließen Sie den Baustein.



2

## Zeitfunktion programmieren in AWL

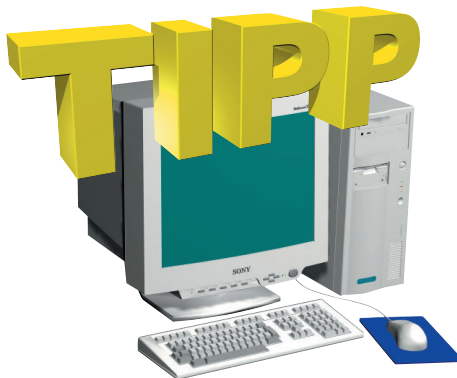
Geben Sie die nebenstehenden Anweisungen ein. Speichern Sie Ihre Eingaben. Schließen Sie den Baustein.



2

## Zeitfunktion programmieren in FUP

Analog zu KOP: Fügen Sie nebenstehende Befehle aus der Bibliothek in das Netzwerk ein, füllen Sie alle **???** aus und geben Sie die Konstante für den Zeitwert ein. Speichern Sie Ihre Eingaben. Schließen Sie den Baustein.



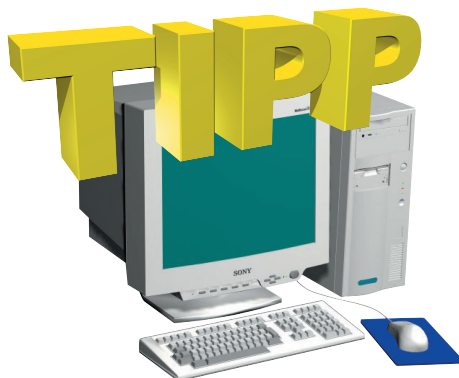
Mit dem Eingangsparameter **"#Motor\_Ein"** wird die **"#Zeitfunktion"** gestartet. Sie wird später beim Aufruf im OB 1 einmal mit den Parametern für den Benzinmotor und einmal mit den Parametern für den Dieselmotor (z. B. T1 für **"BM\_Nachlauf"**) versorgt.

## Funktion im OB 1 aufrufen



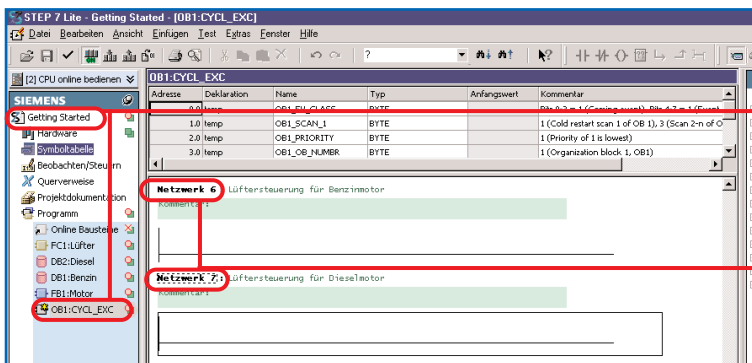
8.8

Der Aufruf der Funktion FC1 erfolgt analog zum Aufruf des Funktionsbausteins im OB 1. Alle Parameter der Funktion werden im OB 1 mit den entsprechenden Operanden des Benzin- bzw. des Dieselmotors versorgt.



Ein Operand ist der Teil einer STEP 7 Lite-Anweisung, der aussagt, womit die CPU etwas tun soll. Er kann absolut oder symbolisch adressiert werden.

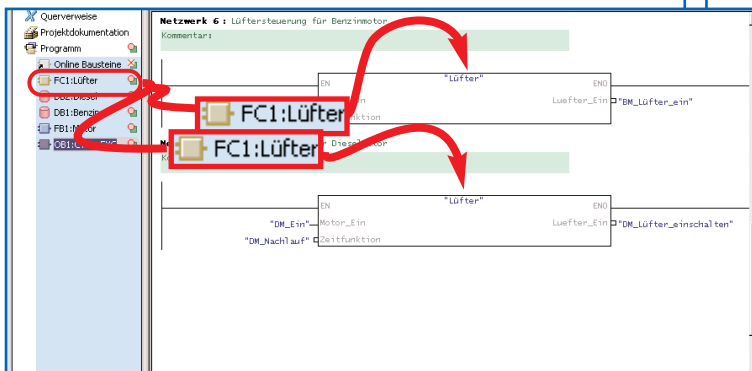




## OB 1 öffnen

- 1 Öffnen Sie das von Ihnen in KOP, FUP oder AWL erstellte Projekt "Getting Started" und öffnen Sie den OB 1.
- 2 Fügen Sie **Netzwerk 6** für den Benzinmotor und **Netzwerk 7** für den Dieselmotor ein.

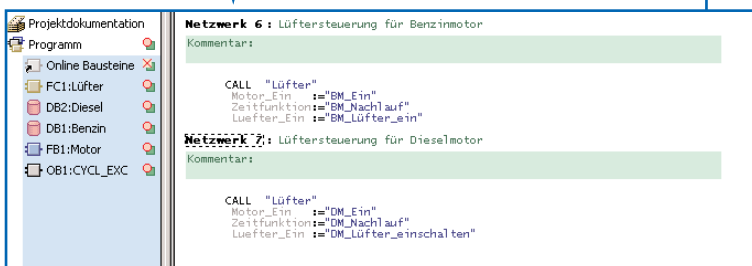
So sieht es nach der Programmierung aus in KOP, ...



## Bausteinanruf programmieren mit KOP

- 3 Ziehen Sie den FC1 per Drag&Drop in Netzwerk 6 und in Netzwerk 7.
- 4 Alle ??? wie nebenstehend ausfüllen. Speichern und Baustein schließen.

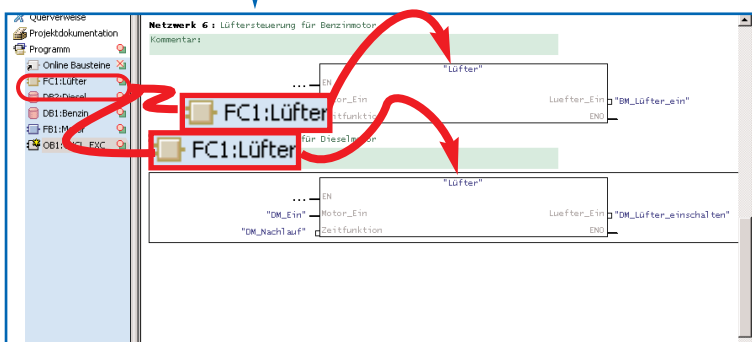
... so in AWL,



## Bausteinanruf programmieren mit AWL

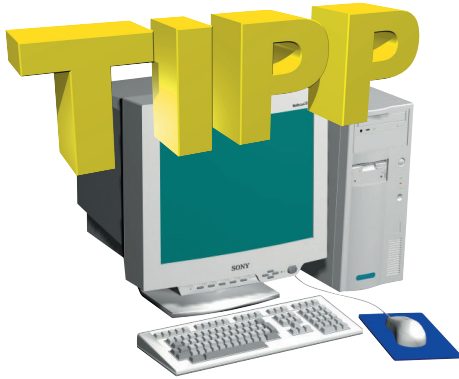
- 3 Nebenstehende AWL-Anweisungen eingeben.
- 4 Speichern und Baustein schließen.

... so in FUP.



## Bausteinanruf programmieren mit FUP

- 3 FC1 per Drag&Drop in Netzwerk 6 Benzinmotor ziehen. FC1 per Drag&Drop in Netzwerk 7 Dieselmotor ziehen.
- 4 Alle ??? wie nebenstehend ausfüllen. Speichern und Baustein schließen.



## 1. Ihr Bildschirm sieht anders aus als in unseren Abbildungen?

Stellen Sie die Symbolische Programmierung ein mit **Ansicht > Anzeigen mit > Symbolischer Darstellung**.

## 2. Sie wollen mehr Bildschirminformation?

Aktivieren Sie **Ansicht > Anzeigen mit > Symbolinformation**, um in jedem Netzwerk Informationen einzelner Adressen zu erhalten.

Um mehrere Netzwerke auf einem Bildschirm darzustellen, deaktivieren Sie **Ansicht > Anzeigen mit > Kommentar** und gegebenenfalls **Ansicht > Anzeigen mit > Symbolinformation**. Mit **Ansicht > Zoomfaktor** können Sie die Darstellungsgröße der Netzwerke verstellen.

## 3. Sie brauchen Informationen zu den Programmiersprachen KOP, AWL oder FUP?

Mehr Informationen finden Sie über **F1 > Inhalt > Aufruf von Referenzhilfen > Sprachbeschreibungen und Bausteinhilfen**.

## 4. Sie wollen die Funktion nicht immer aufrufen?

Der Aufruf der Funktionen wurde in unserem Beispiel als unbedingter Aufruf programmiert, d. h. die Funktion wird immer bearbeitet. Abhängig von den Anforderungen Ihrer Automatisierungsaufgabe können Sie den Aufruf von FCs oder FBs auch mit gewissen Bedingungen verknüpfen: z. B. an einen Eingang oder an eine Vorverschaltung. Zur Programmierung von Bedingungen steht der EN-Eingang bzw. der ENO-Ausgang der Box zur Verfügung.

# 9

## Verwenden von globalen Datenbau- steinen



# Globalen Datenbaustein (DB) anlegen und öffnen



9.2

Falls die Anzahl der internen Merker einer CPU (Speicherzellen) nicht mehr ausreicht, den Datenbestand aufzunehmen, können ausgesuchte Daten in einem globalen Datenbaustein abgelegt werden.

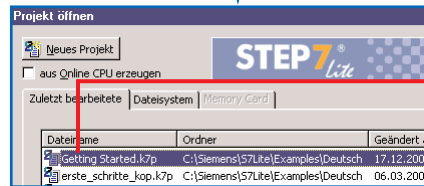
Die Daten des globalen Datenbausteins stehen jedem anderen Baustein zur Verfügung. Ein Instanz-Datenbaustein hingegen ist einem bestimmten Funktionsbaustein zugeordnet, seine Daten sind nur lokal in diesem Funktionsbaustein verfügbar (vgl. Kap 7, Instanz-Datenbausteine erzeugen und Aktualwerte ändern).

Für dieses Kapitel sollten Sie die Symboltabelle in Ihr Projekt "Getting Started" kopiert haben (siehe Seite 5.5).

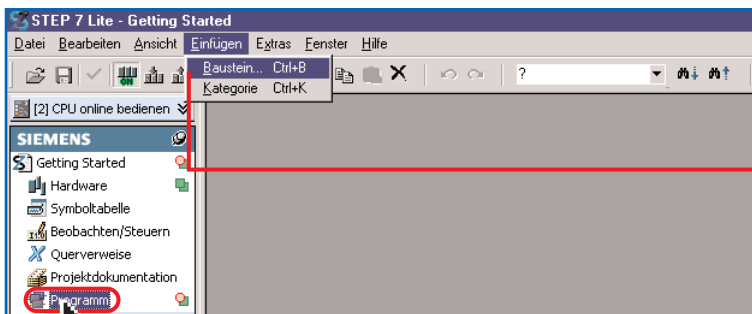


Das Dialogfeld für die Projektauswahl wird geöffnet.

1 Wenn notwendig öffnen Sie STEP 7 Lite.



2 Doppelklicken im Dialogfeld **Projekt öffnen** auf Ihr Projekt "Getting Started".

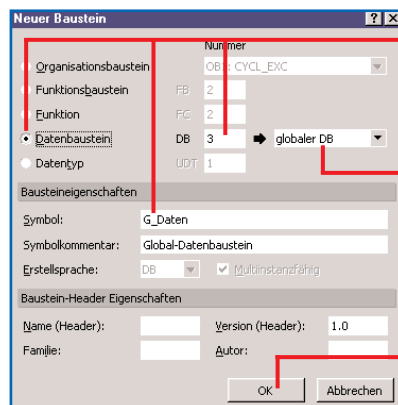


3 Klicken Sie im Projektfenster auf Programm.

4 Wählen Sie den Menübefehl **Einfügen > Baustein** oder über die rechte Maustaste den Kontextmenübefehl **Neu > Baustein**.

CLICK

Das Dialogfeld zum Anlegen neuer Bausteine wird geöffnet.

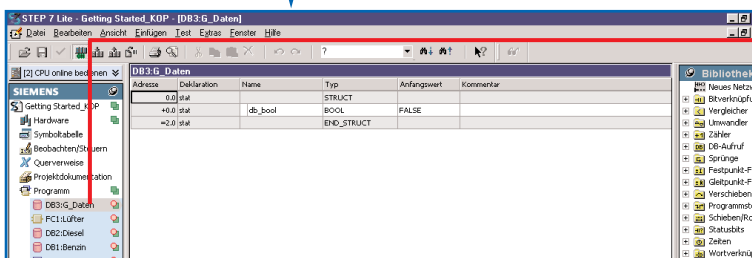


5 Markieren Sie **Datenbaustein**. Im Feld **DB** wird automatisch eine "3" eingetragen und im Feld **Symbol** "G\_Daten".

6 Wählen Sie hier "globaler DB" aus.

7 Bestätigen Sie mit **OK**.

Der Baustein wird eingefügt und sofort geöffnet.



8 Der neue Baustein wird in das Projektfenster eingefügt und sofort geöffnet.

## Variablen im DB programmieren

Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
0.0	stat		STRUCT		
+0.0	stat	BM_Ist_Drehzahl	INT	0	Ist-Drehzahl für Benzinmotor
+2.0	stat	DM_Ist_Drehzahl	INT	0	Ist-Drehzahl für Dieselmotor
+4.0	stat	Soll_erreicht	BOOL	FALSE	beide Motoren haben Soll-Drehzahl erreicht
=6.0	stat		END_STRUCT		

9.4

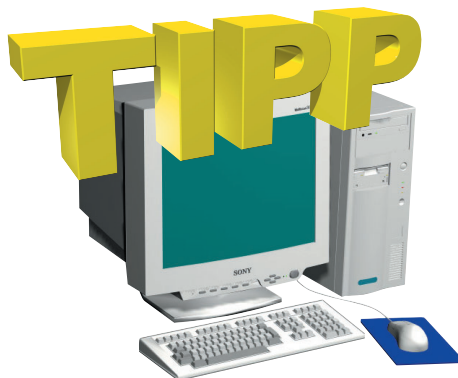
1

Tragen Sie in Spalte **Name** "BM\_Ist\_Drehzahl" ein.

2

Wählen Sie für **Typ** mit der rechten Maustaste über das Kontextmenü **Elementare Typen > INT** aus. Vervollständigen Sie die Liste, wie im Bild dargestellt.

Speichern Sie Ihre Eingaben und schließen Sie den Baustein.



Mehr Informationen finden Sie über **F1 > Inhalt Programmieren von Bausteinen > Erstellen von Datenbausteinen**.



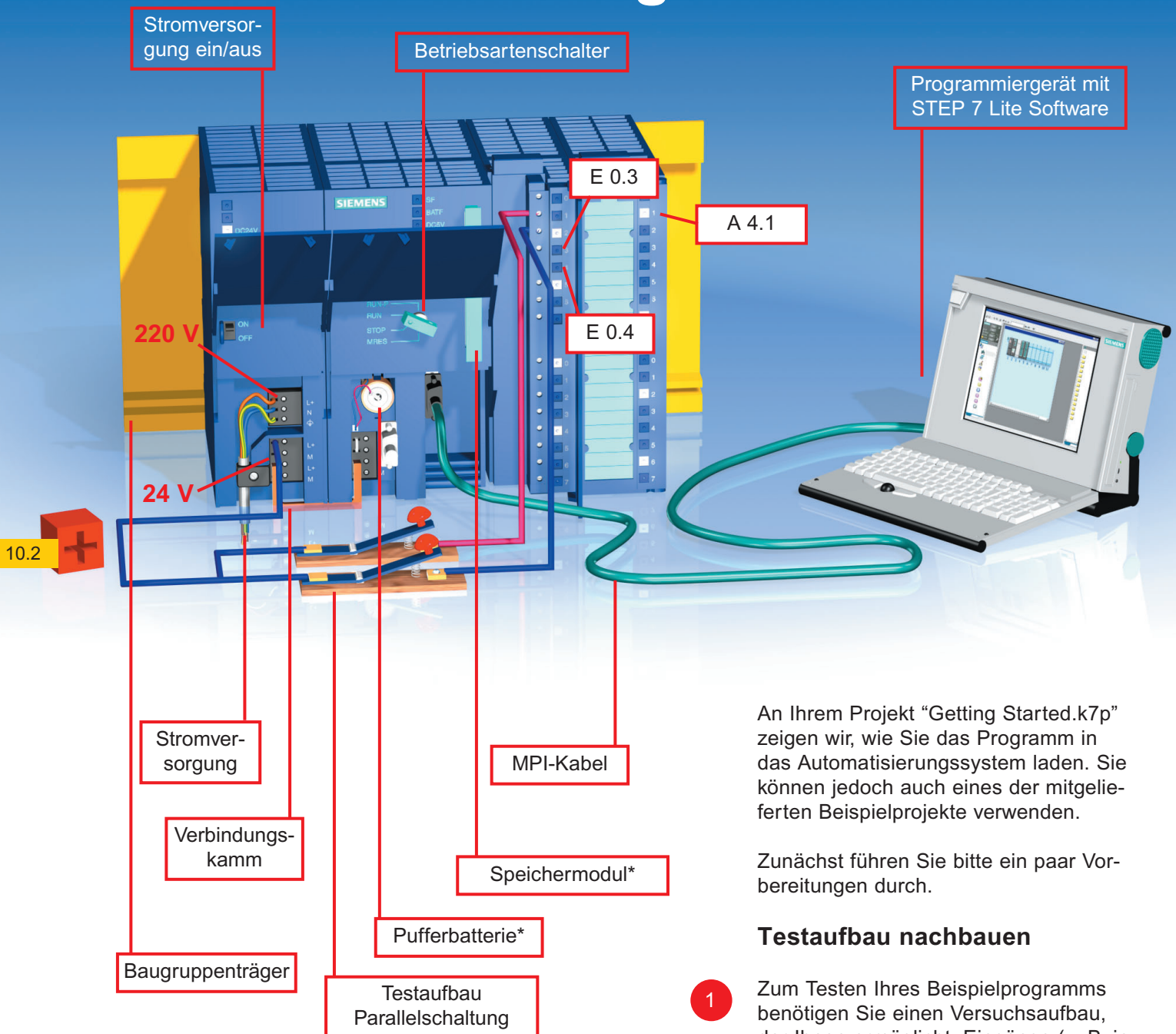
# 10

Laden des  
Programms in  
die CPU





## Online-Verbindung aufbauen und Online gehen



An Ihrem Projekt "Getting Started.k7p" zeigen wir, wie Sie das Programm in das Automatisierungssystem laden. Sie können jedoch auch eines der mitgelieferten Beispielprojekte verwenden.

Zunächst führen Sie bitte ein paar Vorbereitungen durch.

### Testaufbau nachbauen

- 1 Zum Testen Ihres Beispielprogramms benötigen Sie einen Versuchsaufbau, der Ihnen ermöglicht, Eingänge (z. B. in Form von Tastern) zu setzen.

\*= Nicht zwingend erforderliches Zubehör

2

## Programm-Check durchführen

Verwenden Sie das von Ihnen erstellte Projekt "Getting Started", dann sollten Sie zumindest die Hardware konfiguriert (Kapitel 4) und die Parallelschaltung programmiert haben (Kapitel 6).

3

## Hardware-Check durchführen

Bauen Sie die Hardware auf und prüfen Sie noch einmal:

- Busverbinder in Baugruppen gesteckt?
- Baugruppen in Profilschiene eingehängt, nach unten geschwenkt und festgeschraubt?
- Stromversorgung 220 V angeschlossen?
- Verbindungskamm eingesetzt?
- Sofern vorhanden, Pufferbatterie und Speichermodul eingesetzt?

10.3

## Online verbinden

4

Online verbinden bedeutet, eine Verbindung zwischen CPU und Programmiergerät herstellen.

- Stecken Sie das MPI-Kabel am Programmiergerät und an der CPU.

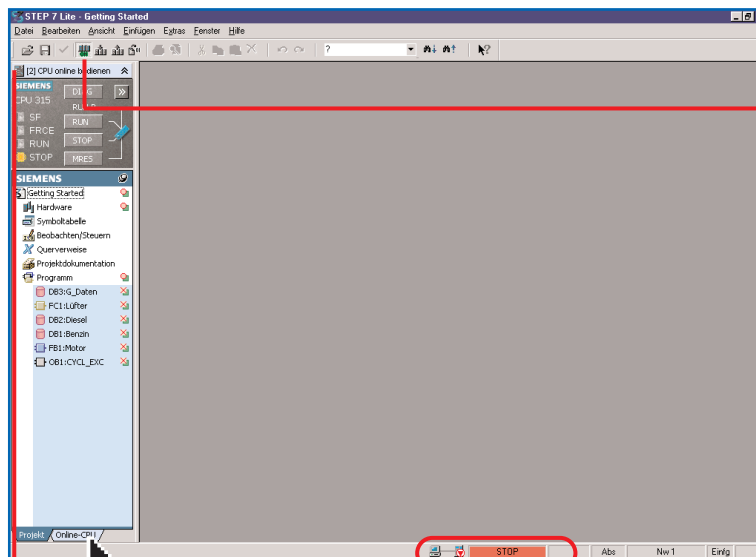
An der CPU:

- Schalten Sie die Stromversorgung **ein**.
- Drehen Sie den Betriebsartenschalter auf **STOP**.



Am Programmiergerät:

- Schalten Sie den Hauptschalter **ein**.
- Starten Sie STEP 7 Lite.
- Öffnen Sie "Getting Started.k7p" oder eines der Beispielprojekte.



## Online-Verbindung herstellen

5

STEP 7 Lite versucht beim Start sofort, eine Online-Verbindung herzustellen.

Die Schaltfläche **Online verbinden** muß grün leuchten und in der Statusleiste wird zuerst der Verbindungsstatus **Online** und dann der Betriebszustand der CPU **STOP** angezeigt.

Wenn keine CPU erreicht werden kann, bleibt STEP 7 Lite Offline und in der Statuszeile wird der Verbindungsstatus **Getrennt** angezeigt.

Beseitigen Sie in diesem Fall die Ursache für den misslungenen Aufbau der Online-Verbindung (z. B. Kabel ist nicht richtig gesteckt oder CPU ist ausgeschaltet) und klicken Sie auf die Schaltfläche **Online verbinden**.

Mit dieser Schaltfläche können Sie die Online-Verbindung jederzeit ein-/aus-schalten.

6

Prüfen Sie, ob **CPU Online bedienen** angezeigt wird. Die Schaltflächen des CPU-Bedienpanels sind jetzt aktiv.

7

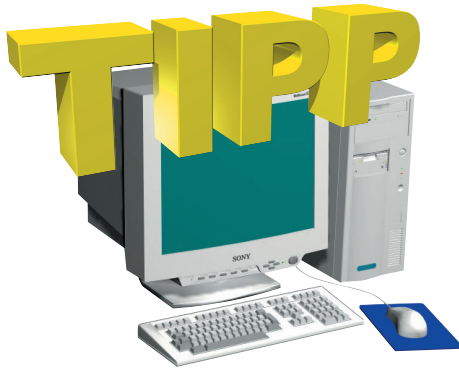
Prüfen Sie: In der Fußzeile wird jetzt der aktuelle Betriebszustand der CPU angezeigt.

8

Probieren Sie aus: Sie können jetzt zwischen der Offline-Sicht **Projekt** und der Online-Sicht **Online-CPU** umschalten.

Im Register **Online-CPU** werden alle Bausteine angezeigt, die sich auf der CPU befinden.

Solange Sie Ihr Projekt noch nicht auf die CPU geladen haben, weisen Symbole im Projektfenster darauf hin, dass die Daten auf dem Programmiergerät und der CPU noch nicht übereinstimmen.

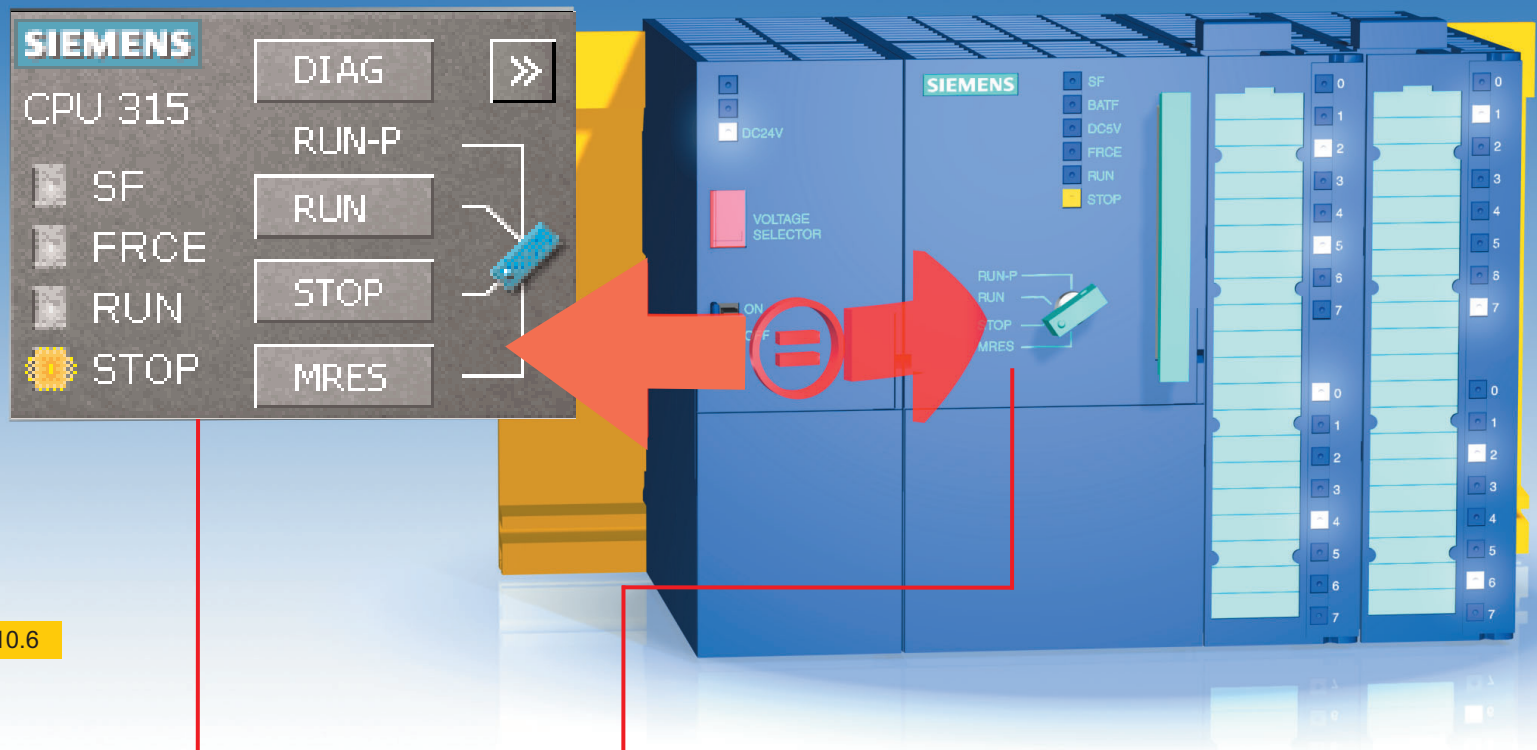


Online verbinden und Testen (Kapitel 11) ist auch durchführbar, wenn Sie eine andere Hardware verwenden als auf Seite 10.2 abgebildet. Halten Sie lediglich die Adressierung der Ein- und Ausgänge ein.

Mehr Informationen zum Aufbauen der zentralen Baugruppen finden Sie in den Handbüchern "S7-300 – Aufbauen, CPU-Daten".

Mehr zum Online-Verbinden enthält die Online-Hilfe über **F1 > Inhalt > Aufbau der Online-Verbindung**.

## CPU urlöschen und Programm übertragen



CPU-Bedienpanel  
in STEP 7 Lite

CPU-Bedienpanel  
an der CPU

### Das CPU-Bedienpanel

Bevor Sie Ihr Programm übertragen müssen Sie die CPU urlöschen. Das machen Sie entweder mit dem CPU-Bedienpanel an der CPU oder mit dem CPU-Bedienpanel in STEP 7 Lite.

Mit dem Bedienpanel stellen Sie auch die Betriebszustände der CPU ein.

Allerdings sind aus Sicherheitsgründen über das Bedienfeld von STEP 7 Lite nur die Schaltflächen aktivierbar, die für den an der CPU eingestellten Betriebszustand angewählt werden dürfen.

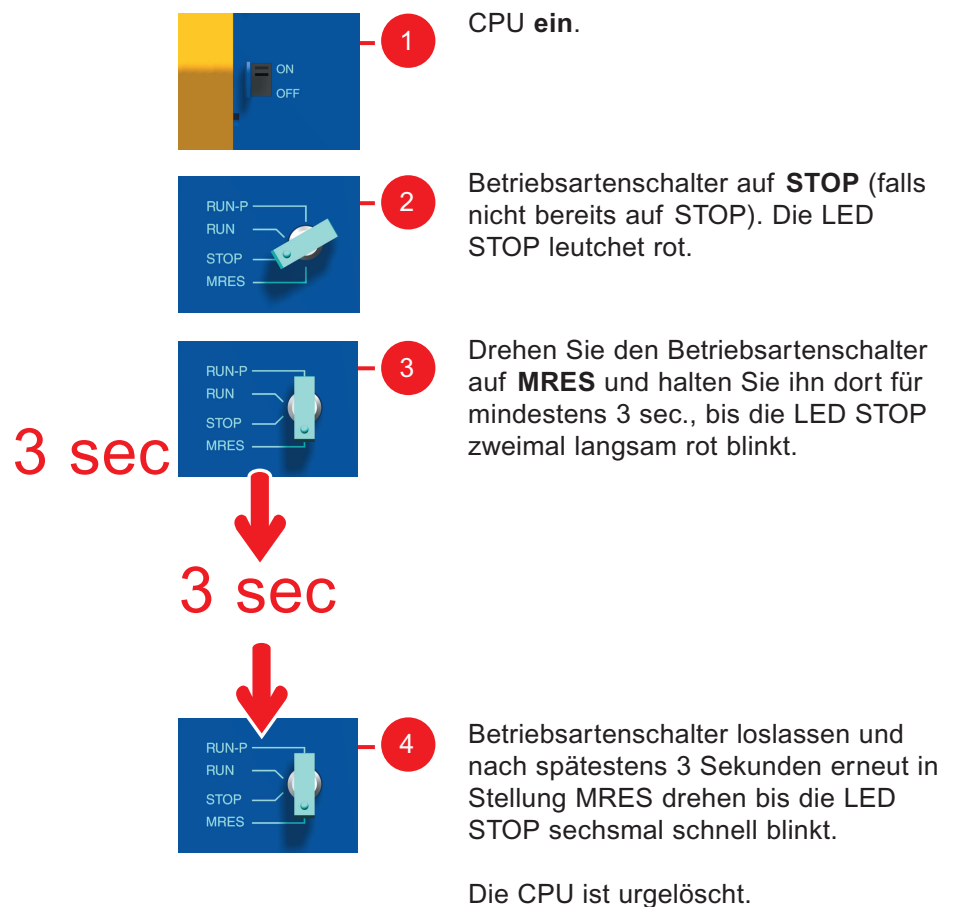
Beispiel:

- An der CPU: Schalter auf **RUN** -  
Am Bedienpanel von STEP 7 Lite: **STOP** ist einstellbar.
- An der CPU: Schalter auf **STOP** -  
Am Bedienpanel von STEP 7 Lite: **RUN** ist nicht einstellbar.

**In einer Gefahrensituation setzen Sie die CPU auch über STEP 7 Lite auf STOP.**

## CPU urlöschen an der CPU

Bevor Sie Ihr Programm auf die CPU übertragen, löschen Sie alle alten Daten und Programme auf der CPU mit dem Befehl urlöschen. Hierzu:



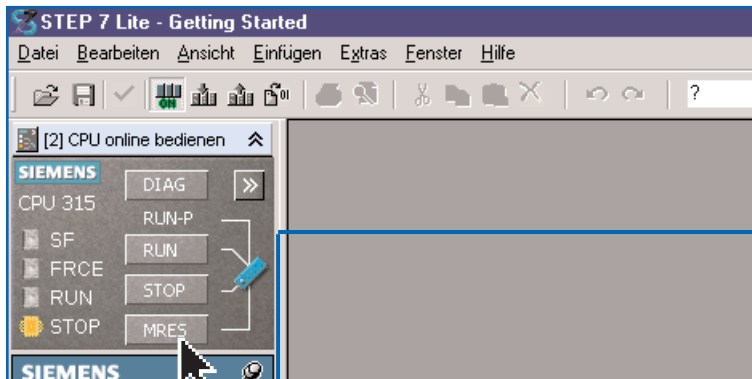
## CPU urlöschen mit STEP 7 Lite

Alternativ können Sie die CPU auch über STEP 7 Lite urlöschen.



1

An der CPU: Stellen Sie den Betriebsartenschalter auf **STOP**.

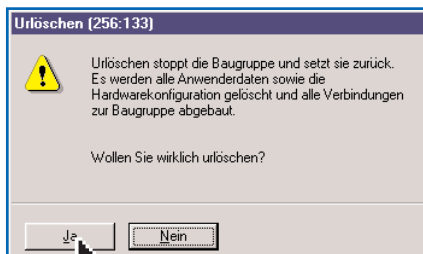


2

In STEP 7 Lite: Klicken Sie auf die Schaltfläche **MRES**.

CLICK

Eine Sicherheitsabfrage erscheint.



CLICK

3

Bestätigen Sie die Sicherheitsabfrage mit **Ja**. Die CPU ist urlöscht.



Hinweise zu den Betriebszuständen finden Sie in der Hilfe zu STEP 7 Lite über **F1 > Inhalt > Anhang > Betriebszustände**.

Beachten Sie:  
Das Wissen über die Betriebszustände der CPUs ist nützlich für die Programmierung des Anlaufs, den Test der Steuerung sowie für die Fehlerdiagnose.

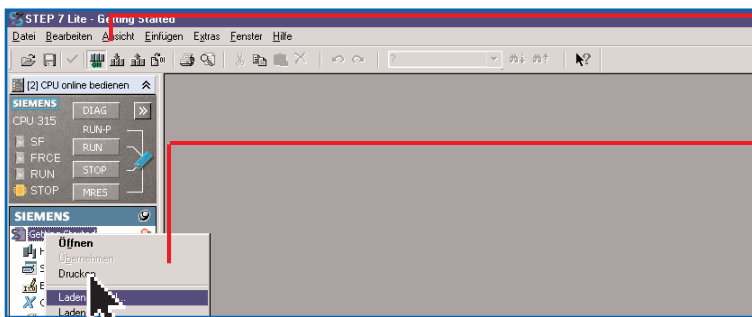


## Programm in CPU laden



1

Prüfen Sie: Beim Laden des Programms müssen die Betriebsarten-schalter an CPU und in STEP 7 Lite auf **Stop** stehen.



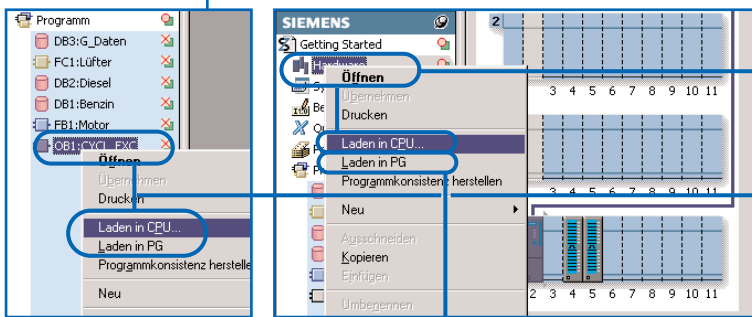
2

Gesamtes Projekt auf die CPU laden: Markieren Sie im Projektfenster das Projekt "Getting Started".

3

Klicken Sie mit der rechten Maustaste auf das Projekt und wählen Sie im Kontextmenü **Laden in CPU**.

Das gesamte Projekt einschließlich der Hardwarekonfiguration wird auf die CPU geladen.



4

In diesem Beispiel ist die Hardware-Konfiguration für das Laden markiert.

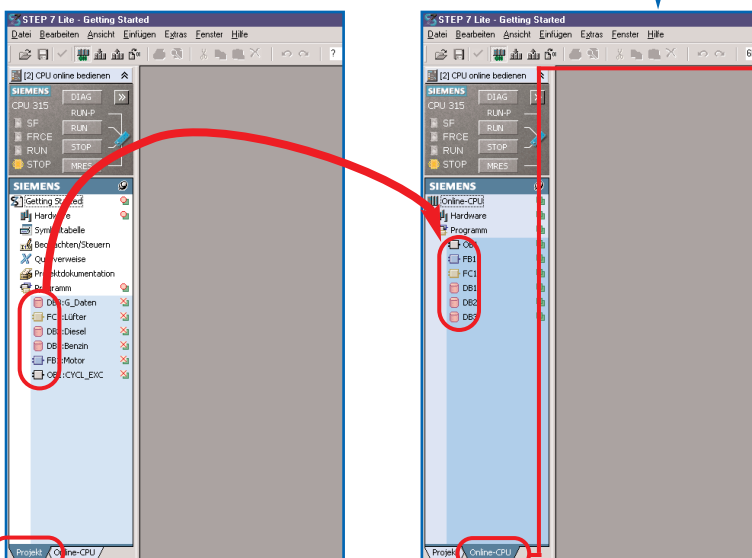
5

In diesem Beispiel ist ein Baustein für das Laden markiert.

5

Abhängig von den markierten Elementen, bietet Ihnen STEP 7 Lite **Laden in PG** (Programmiergerät) an.

Die Sicht "Online-CPU" hat sich geändert.



6

Klicken Sie auf **Online-CPU**.

Hier werden alle Objekte angezeigt, die sich auf der CPU befinden.

- **Projekt** (Offline-Sicht)  
= Objekte auf dem Programmiergerät
- **Online-CPU** (Online-Sicht)  
= Objekte auf der CPU

## CPU einschalten und Betriebszustand prüfen



1

Drehen Sie den Betriebsartenschalter Auf **RUN-P**. Die LED **RUN** leuchtet grün und die LED **STOP** rot erlischt. Die CPU ist betriebsbereit.

Die zyklische Abarbeitung des Programms wird angezeigt.

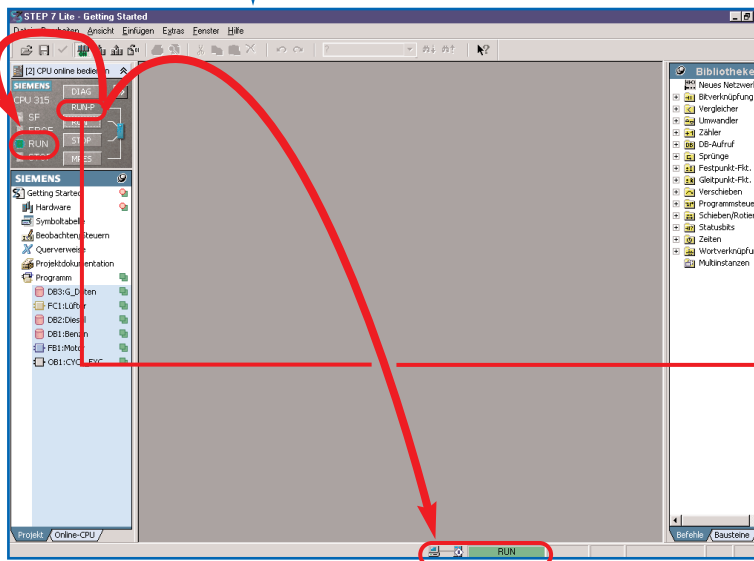
2

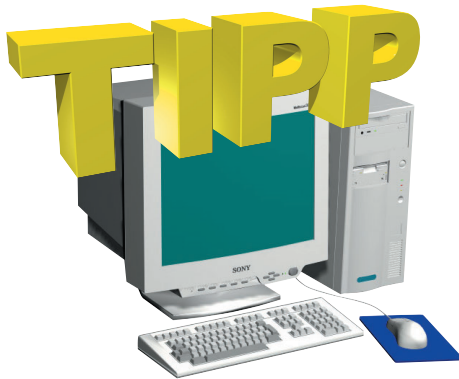
Prüfen Sie an der CPU:  
Wenn die LED grün leuchtet, können Sie mit dem Testen des Programms beginnen.

Leuchtet die LED weiterhin rot, liegt ein Fehler vor. Zur Fehlerdiagnose können Sie durch Klicken auf die Schaltfläche "DIAG" den Diagnosepuffer auswerten (siehe auch Abschnitt "Baugruppenzustand und Fehlerhistorie", S. 12.5).

3

Prüfen Sie in STEP 7 Lite:  
Mit dem Umschalten an der CPU wurde auch hier auf **RUN-P** umgeschaltet und die zyklische Abarbeitung des Programms durch grüne Hinterlegung angezeigt.





#### **Urlöschen:**

Trotz Urlöschen befinden sich in der CPU die Systemfunktionsbausteine (SFBs) und Systemfunktionen (SFCs). Diese Funktionen des Betriebssystems stellt die CPU bereit. Sie müssen nicht geladen werden, können jedoch auch nicht gelöscht werden.

#### **Laden einzelner Bausteine:**

Um in der Praxis schnell auf Fehler reagieren zu können, lassen sich Bausteine einzeln auf die CPU übertragen. Beim Laden von Bausteinen muß der Betriebsartenschalter an der CPU entweder auf "RUN-P" oder "STOP" stehen. Im Betriebszustand "RUN-P" geladene Bausteine werden sofort aktiviert. Sie sollten dabei bedenken:



Werden fehlerfreie Bausteine durch fehlerhafte Bausteine überschrieben, hat dies eine Fehlfunktion Ihrer Anlage zur Folge.



Wurde die Reihenfolge für das Laden der Bausteine nicht beachtet (z. B. ein Baustein wird im OB 1 aufgerufen, ist aber noch gar nicht auf der CPU vorhanden), geht die CPU in den Betriebszustand "STOP" über.

10.11

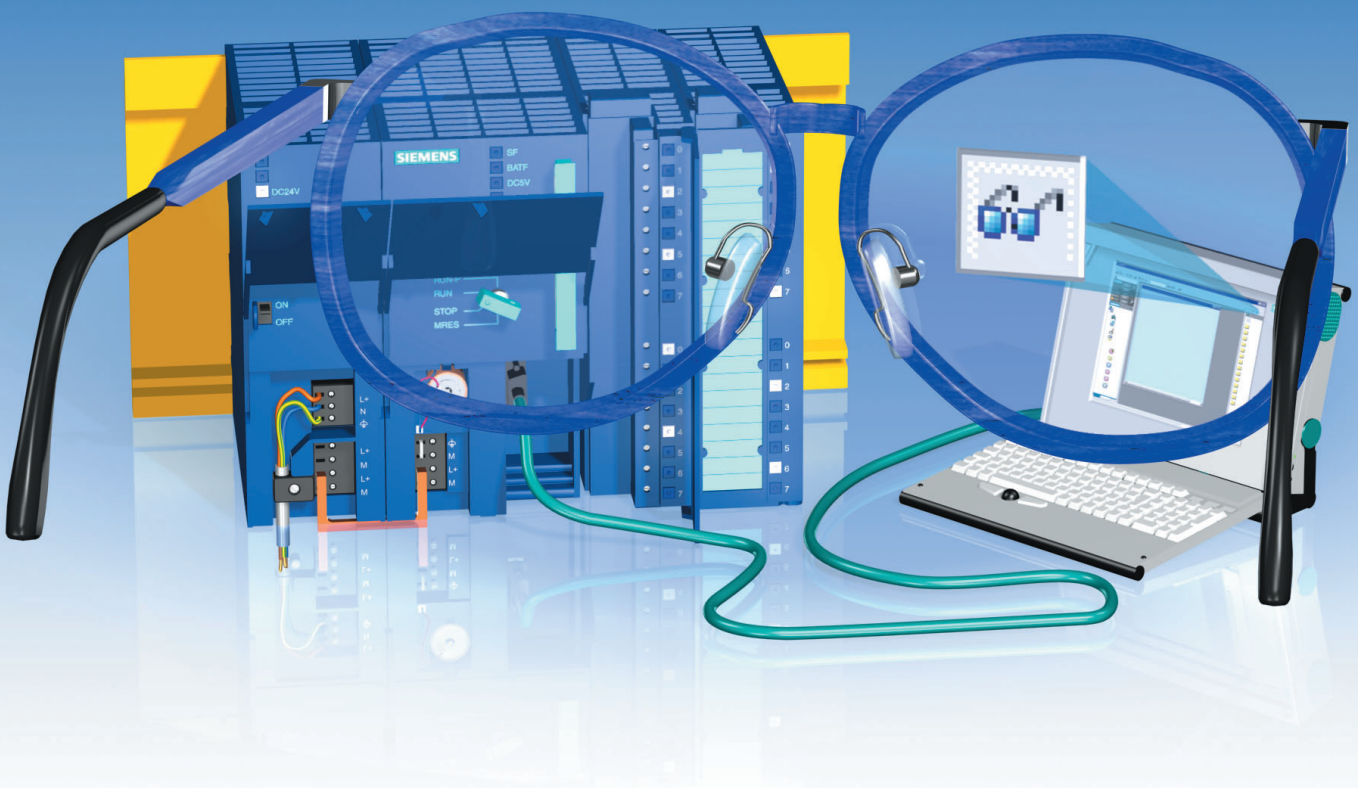
#### **CPU 31xC:**

Bei einer CPU 31xC ist der Betriebsartenschalter nicht als Drehschalter, sondern als Kippschalter ausgeführt und es gibt keine Betriebsart RUN-P. Die Vorgehensweise beim Urlöschen ist aber gleich. Informationen zu Micro Memory Cards finden Sie über **F1 > Index > Micro Memory Card**.

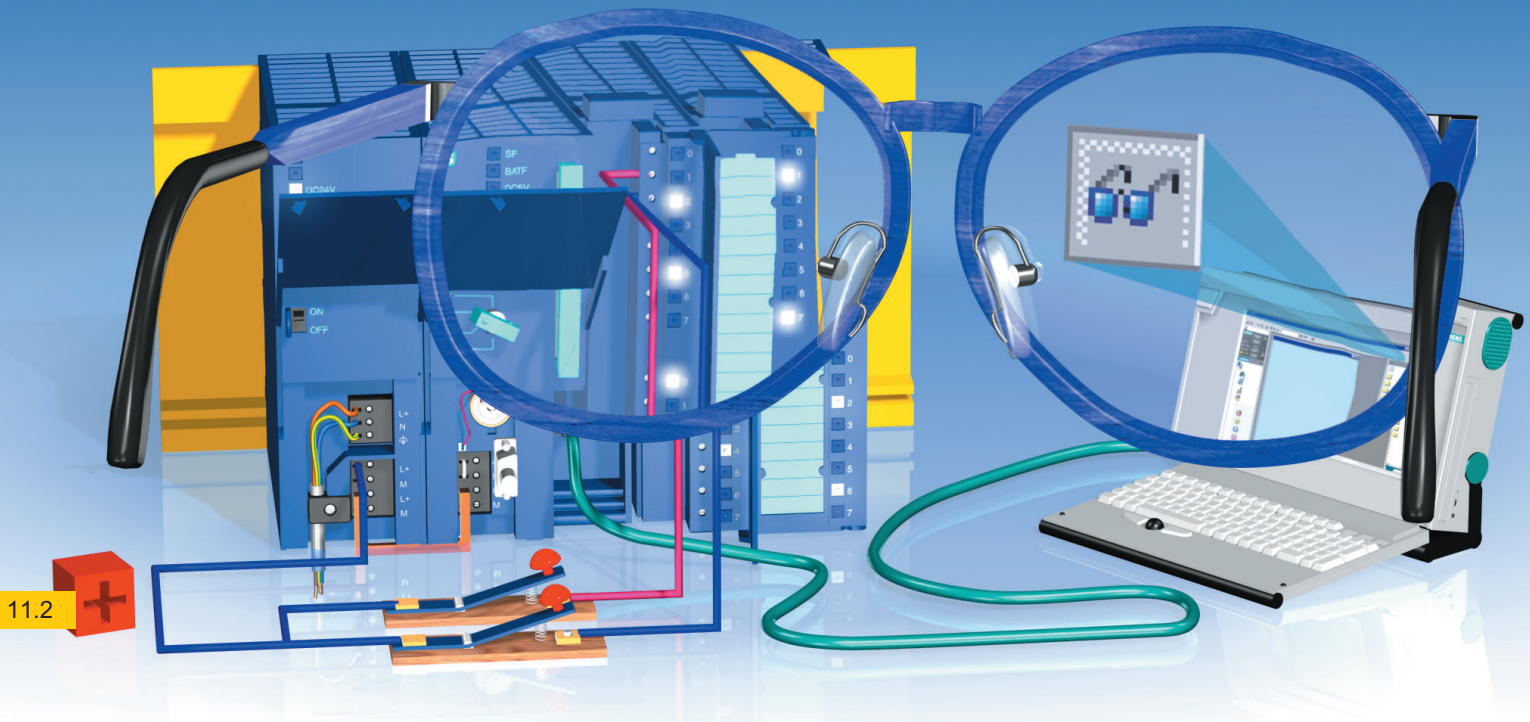


# 11

## Testen des Programms



# Programm mit Programmstatus testen



11.2

STEP 7 Lite bietet die Möglichkeit, den Ablauf Ihres Programms auf der SPS-Station zu testen. Folgende Testmöglichkeiten stehen Ihnen zur Verfügung:

1. Testen mit dem Programmstatus – zur Verfolgung des Programmablaufs (siehe S. 11.3 bis 11.5).
2. Testen mit Variablentabelle – zum Beobachten und Beeinflussen von Operanden, z. B. Eingängen, Ausgängen, Merkern (siehe S. 11.6 bis 11.11).

Voraussetzung für das Testen mit Programmstatus: Das gesamte Projekt muss in die CPU geladen worden sein.

## Vorbereitungen

1

Online-Verbindung herstellen.



2

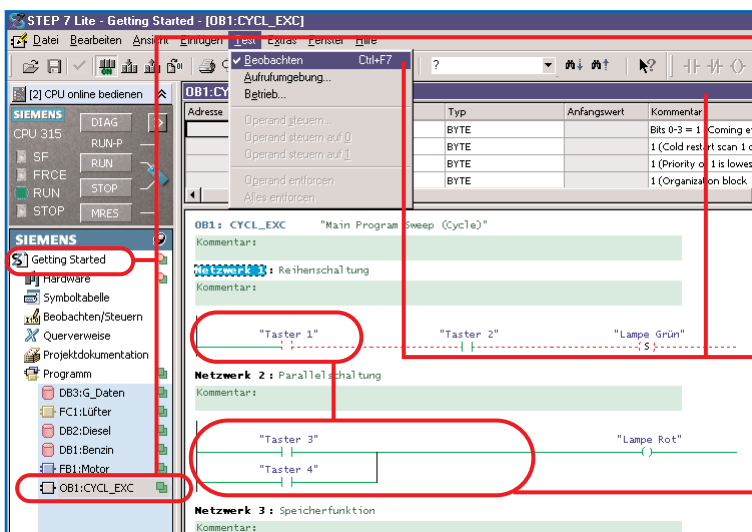
Schlüsselschalter an der CPU auf RUN oder RUN-P stellen.

3

Für Netzwerk 1: Serienschaltung mit Kabeln aufbauen. Für Netzwerk 2: Parallelschaltung mit Kabeln aufbauen (siehe Grafik)

4

Das Projekt "Getting Started" oder eines der Beispielprojekte, das auf die CPU geladen wurde, öffnen. OB 1 öffnen.



5

Funktion Beobachten per **Test > Beobachten** aufrufen. Die Funktion ist nur nach "Online verbinden" aktivierbar.

6

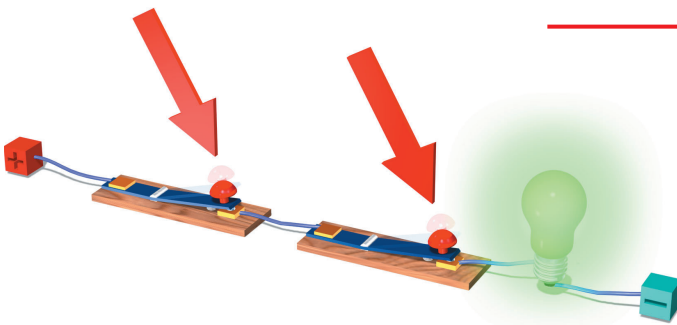
Die schwarzen Strompfade in den Netzwerken werden jetzt farbig dargestellt.

Strompfad **GRÜN**: Strom fließt.  
Strompfad **ROT**: Strom fließt nicht.

7

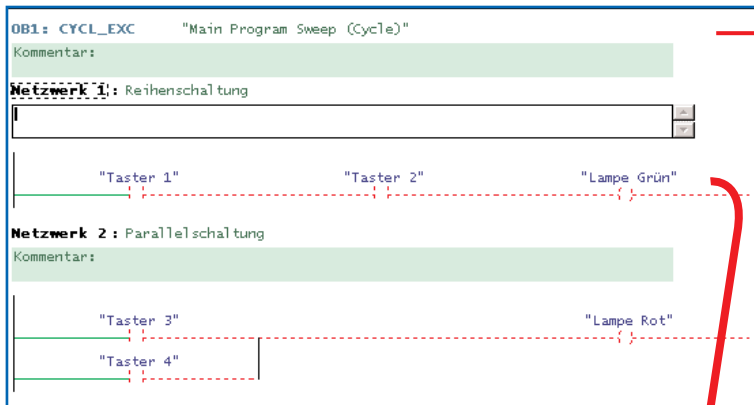
In Ihrem Versuchsaufbau schließen Sie jetzt nacheinander die Taster und beobachten hierbei:

- In STEP 7 Lite: Wie sich die Farben der Strompfade ändern.
- An den Baugruppen: Wie die LEDs an den Ein- und Ausgangsbaugruppen leuchten und erlöschen.





## Testen mit KOP

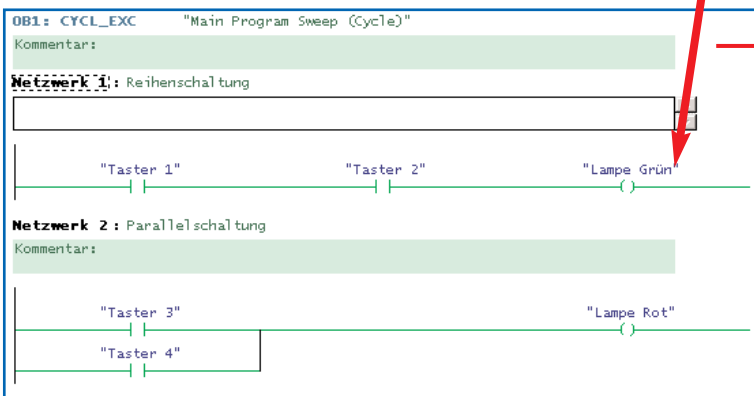


1

Sie lassen alle Taster geöffnet.

In den Netzwerken 1 und 2 liegt Spannung bis zu den Tastern 1, 3, 4 an. Die Strompfade sind hier grün. Ab den Tastern 1, 3, 4 fließt noch kein Strom, die Strompfade sind rot.

Der Farbumschlag symbolisiert, dass das Verknüpfungsergebnis bis zu dieser Stelle erfüllt ist.

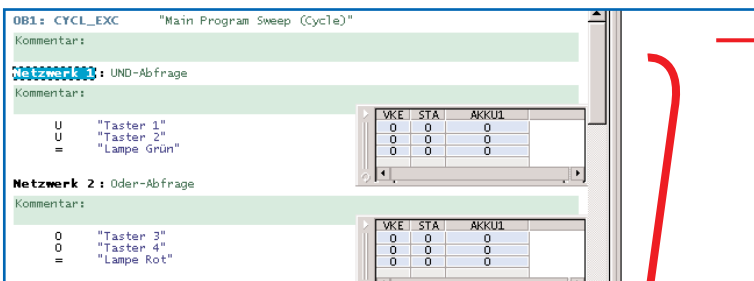


2

Jetzt schließen Sie die Taster 1, 2, 3, 4. In alle Pfade fließt jetzt Strom.

Haben Sie eines unserer Beispielprojekte geöffnet, können Sie in den Kommentaren mitverfolgen, welche Dioden an den Ein- und Ausgangsbaugruppen leuchten müssen.

## Testen mit AWL



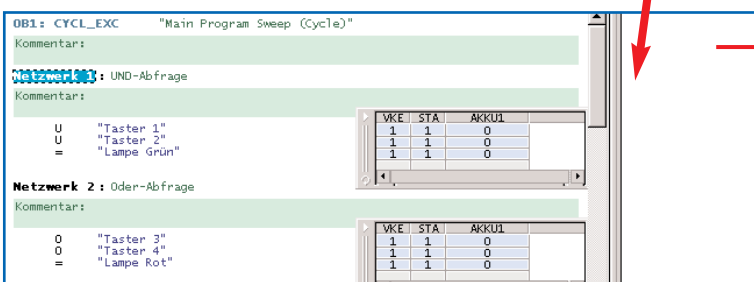
1

Sie lassen alle Taster geöffnet.

Für AWL werden

- Verknüpfungsergebnis (VKE)
- Statusbit (STA)
- Akkumulator (AKKU1)

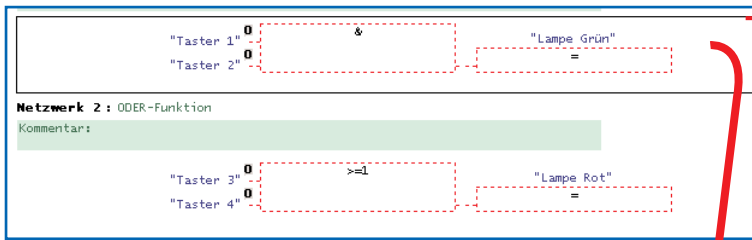
in Tabellenform angezeigt.



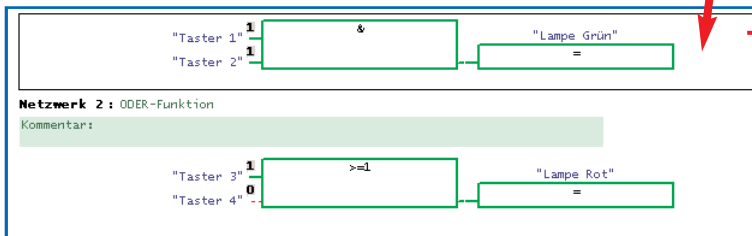
2

Jetzt schließen Sie die Taster 1, 2, 3, 4. Das Verknüpfungsergebnis ist an allen Stellen erfüllt.

## Testen mit FUP

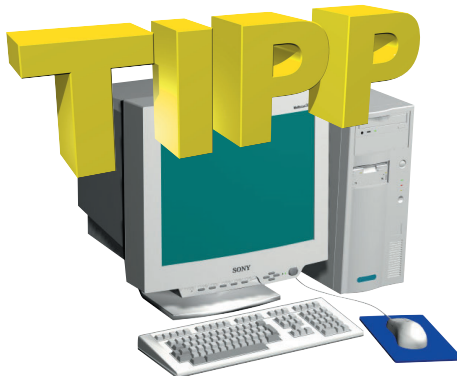


1 Sie lassen alle Taster geöffnet.



2 Jetzt schließen Sie die Taster 1, 2, 3, 4. Der Farbumschlag signalisiert, dass das Verknüpfungsergebnis an allen Stellen erfüllt ist.

3 Deaktivieren Sie **Test > Beobachten** und schließen Sie das Fenster.



Über **Extras > Einstellungen > Bausteineditor** können Sie die Darstellungsart der Testergebnisse ändern.

Mehr Information erhalten Sie über **F1 > Inhalt > Testen > Testen mit Programmstatus**.

## Variablen beobachten und steuern

Beobachten starten

Operanden beobachten

Operanden steuern

Beobachten/Steuern aufrufen

Eingabefelder zum Erstellen der Variablentabelle

Anzeigefelder für Statuswert z.B. "true" oder "false"

Eingabefeld für Steuerwert

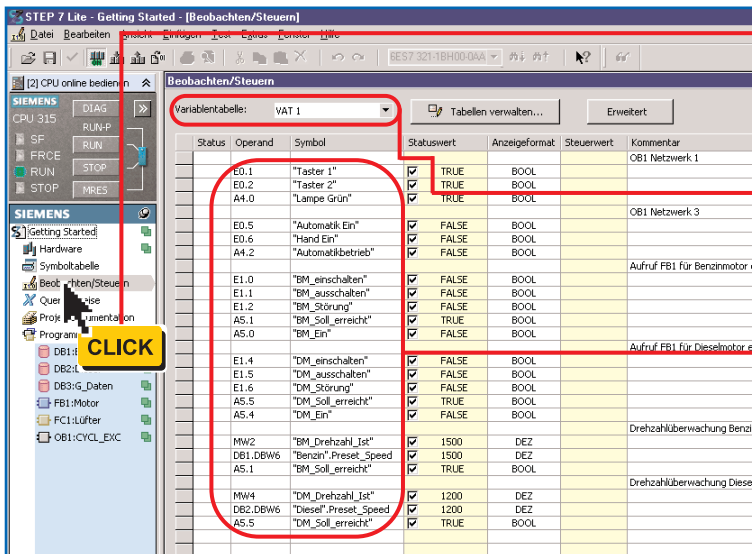
Beobachten/Steuern erweitern oder reduzieren

Beobachten/Steuern

Status	Operand	Symbol	Statuswert	Anzeigeformat	Steuerwert	Kommentar
	E0.1	"Taster 1"	<input checked="" type="checkbox"/> TRUE	BOOL		OB1 Netzwerk 1
	E0.2	"Taster 2"	<input checked="" type="checkbox"/> TRUE	BOOL		
	A4.0	"Lampe Grün"	<input checked="" type="checkbox"/> TRUE	BOOL		
	E0.5	"Automatik Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		OB1 Netzwerk 3
	E0.6	"Hand Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		
	A4.2	"Automatikbetrieb"	<input checked="" type="checkbox"/> FALSE	BOOL		
	E1.0	"BM_einschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		Aufruf FB1 für Benzinmotor einschalten
	E1.1	"BM_ausschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
	E1.2	"BM_Störung"	<input checked="" type="checkbox"/> FALSE	BOOL		
	A5.1	"BM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		
	A5.0	"BM_Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		Aufruf FB1 für Dieselmotor einschalten
	E1.4	"DM_einschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
	E1.5	"DM_ausschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
	E1.6	"DM_Störung"	<input checked="" type="checkbox"/> FALSE	BOOL		
	A5.5	"DM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		
	A5.4	"DM_Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		
	MW2	"BM_Drehzahl_Ist"	<input checked="" type="checkbox"/> 1500	DEZ		Drehzahlüberwachung Benzinmotor
	DB1.DBW6	"Benzin".Preset_Speed	<input checked="" type="checkbox"/> 1500	DEZ		
	A5.1	"BM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		Drehzahlüberwachung Dieselmotor
	MW4	"DM_Drehzahl_Ist"	<input checked="" type="checkbox"/> 1200	DEZ		
	DB2.DBW6	"Diesel".Preset_Speed	<input checked="" type="checkbox"/> 1200	DEZ		
	A5.5	"DM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		

Sie können Operanden beobachten, in dem Sie diese in die Variablentabelle eintragen. Voraussetzung ist eine bestehende Online-Verbindung zur CPU.

Sie können Operanden steuern in dem sie einen Steuerwert angeben und den Haken setzen. Voraussetzung ist zusätzlich zur Online-Verbindung, dass sich die CPU im Betriebszustand RUN-P befindet.



## Variablentabelle erstellen

Öffnen Sie Ihr Projekt "Getting Started". Doppelklicken Sie auf **Beobachten/Steuern**.

Legen Sie in Feld **Variablentabelle** eine neue Tabelle mit dem Namen "VAT 1" an.

Tragen Sie für das Beispiel "Getting Started" alle Variablen ein, bzw. nur die Variablen, die Sie steuern möchten.

Hierzu:

Geben Sie in Spalte **Operand** "E0.1" ein. Nach Return wird "Taster 1" automatisch aus der Symboltabelle übernommen. Vervollständigen Sie die Tabelle gemäß unserer Abbildung. Sie können auch den Cursor in die Spalte Operand platzieren und mit **Strg + j** die Operanden aus der angebotenen Liste auswählen.

Alternativ kopieren Sie die Variablentabelle aus einem der Beispielprojekte.

Öffnen Sie in einer zweiten Instanz von STEP 7 Lite eines der Beispielprojekte. Klicken Sie dort ebenfalls auf **Beobachten/Steuern** und öffnen Sie die **VAT 1**.

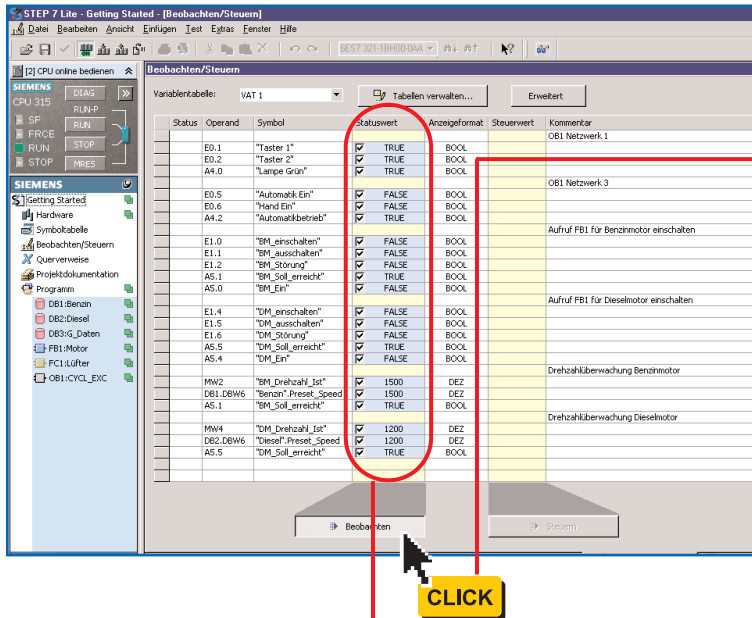
Markieren Sie mit **Strg + a** die gesamte Tabelle und kopieren Sie sie mit **Strg + c** in den Zwischenspeicher.

Wechseln Sie zum Projekt "Getting Started".

Fügen Sie mit **Strg + v** die Daten aus dem Zwischenspeicher ein.

## Variablen beobachten

Beim Beobachten der Variablen testen Sie Ihr Programm, aber auch die fehlerfreie Funktion Ihrer Hardware.



1

Schließen Sie an Ihrem Versuchsaufbau "Taster 1" und "Taster 2".

2

Klicken Sie auf **Beobachten**. Die Spalte Statuswert ist jetzt blau hinterlegt, Variablen werden beobachtet.

3

Sie können beobachten,

- wie die Anzeigen in Spalte **Statuswert** von "FALSE" nach "TRUE", wechseln und
- wie gleichzeitig die LEDs an den Ein- und Ausgangsbaugruppen aufleuchten bzw. erlöschen, wenn Sie Taster am Versuchsaufbau drücken.

4

Für Ihren Programm- bzw. Hardwaretest prüfen Sie jetzt, ob die Zustände in ihrer Kombination

- Tasterstellung auf/zu
  - LED an/aus
  - Variable true/false
- plausibel erscheinen.

Beim Steuern können Sie Variablen mit Werten vorbelegen und damit bestimmte Situationen für den Programmablauf simulieren.

Für das Steuern muss sich Ihre CPU wieder im Betriebszustand RUN-P befinden, **Beobachten** bleibt dabei aktiv.

Aktivieren Sie den Steuerwert durch Setzen eines Häkchens in dem Optionsfeld, das angezeigt wird, sobald der Steuerwert eingetragen wurde.

Beobachten Sie die Auswirkungen der gesteuerten Variablen in Spalte **Statuswert**.



# Testen des Programms

SIEMENS CPU 315

DIAG RUN-P RUN STOP MRES

SIEMENS

Getting Started Hardware Symboltabelle Beobachten/Steuern Querverweise Projektdokumentation Programm

DB1: Benzin DB2: Diesel DB3: G\_Daten FB1: Motor FC1: Lüfter OB1: CYC\_EXC

Variablenliste: VAT 1

Tabelle verwalten... Erweitert

Status	Operand	Symbol	Statuswert	Anzeigeformat	Steuerwert	Kommentar
<input checked="" type="checkbox"/>	E0.1	"Taster 1"	<input checked="" type="checkbox"/> TRUE	BOOL	<input checked="" type="checkbox"/> TRUE	OB1 Netzwerk 1
<input checked="" type="checkbox"/>	E0.2	"Taster 2"	<input checked="" type="checkbox"/> TRUE	BOOL	<input checked="" type="checkbox"/> TRUE	
<input checked="" type="checkbox"/>	A4.0	"Lampe Grün"	<input checked="" type="checkbox"/> TRUE	BOOL		OB1 Netzwerk 3
<input checked="" type="checkbox"/>	E0.5	"Automatik Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E0.6	"Hand Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	A4.2	"Automatikbetrieb"	<input checked="" type="checkbox"/> TRUE	BOOL		Aufruf FB1 für Benzinmotor ein
<input checked="" type="checkbox"/>	E1.0	"BM_einschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E1.1	"BM_ausschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E1.2	"BM_Störung"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	AS.1	"BM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		
<input checked="" type="checkbox"/>	AS.0	"BM_Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		Aufruf FB1 für Dieselmotor ein
<input checked="" type="checkbox"/>	E1.4	"DM_einschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E1.5	"DM_ausschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E1.6	"DM_Störung"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	AS.5	"DM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		
<input checked="" type="checkbox"/>	AS.4	"DM_Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		Drehzahlüberwachung Benzinmotor
<input checked="" type="checkbox"/>	MW2	"BM_Drehzahl_Ist"	<input checked="" type="checkbox"/> 1500	DEZ	<input checked="" type="checkbox"/> 1500	
<input checked="" type="checkbox"/>	DB1.DBW6	"Benzin_Preset_Speed"	<input checked="" type="checkbox"/> 1500	DEZ	<input checked="" type="checkbox"/> 1500	
<input checked="" type="checkbox"/>	AS.1	"BM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		Drehzahlüberwachung Dieselmotor
<input checked="" type="checkbox"/>	MW4	"DM_Drehzahl_Ist"	<input checked="" type="checkbox"/> 1200	DEZ	<input checked="" type="checkbox"/> 1200	
<input checked="" type="checkbox"/>	DB2.DBW6	"Diesel_Preset_Speed"	<input checked="" type="checkbox"/> 1200	DEZ	<input checked="" type="checkbox"/> 1200	
<input checked="" type="checkbox"/>	AS.5	"DM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		

Beobachten Steuern

Es lassen sich nicht nur binäre Operanden steuern. Für nicht-binäre Operanden sollten Sie zuerst das Anzeigeformat wählen und dann einen entsprechenden Steuerwert eingeben.

Wenn Sie zum Steuern der Solldrehzahl DEZ in die Spalte **Anzeigeformat** eintragen,

können Sie für die Steuerwerte die Drehzahlwerte "1200" und "1500" dezimal eintragen.

Geben Sie einen Wert ein, der nicht zum Anzeigeformat passt, wird die Zelle rot hinterlegt und in Spalte **Steuerwert** werden keine Optionsfelder mehr angezeigt.

Das Anzeigeformat einer Variable können Sie nach Klick auf den Typ in der Spalte **Anzeigeformat** ändern.

8

## Erweiterte Sicht Beobachten/Steuern

Beobachten/Steuern

Variablenliste: VAT 1

Tabelle verwalten... Erweitert PA freischalten

Status	Operand	Symbol	Statuswert	Anzeigeformat	Steuerwert	Kommentar
<input checked="" type="checkbox"/>	E0.1	"Taster 1"	<input checked="" type="checkbox"/> TRUE	BOOL	<input checked="" type="checkbox"/> TRUE	OB1 Netzwerk 1
<input checked="" type="checkbox"/>	E0.2	"Taster 2"	<input checked="" type="checkbox"/> TRUE	BOOL	<input checked="" type="checkbox"/> TRUE	
<input checked="" type="checkbox"/>	A4.0	"Lampe Grün"	<input checked="" type="checkbox"/> TRUE	BOOL		OB1 Netzwerk 3
<input checked="" type="checkbox"/>	E0.5	"Automatik Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E0.6	"Hand Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	A4.2	"Automatikbetrieb"	<input checked="" type="checkbox"/> TRUE	BOOL		Aufruf FB1 für Benzinmotor einschalten
<input checked="" type="checkbox"/>	E1.0	"BM_einschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E1.1	"BM_ausschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E1.2	"BM_Störung"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	AS.1	"BM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		
<input checked="" type="checkbox"/>	AS.0	"BM_Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		Aufruf FB1 für Dieselmotor einschalten
<input checked="" type="checkbox"/>	E1.4	"DM_einschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E1.5	"DM_ausschalten"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	E1.6	"DM_Störung"	<input checked="" type="checkbox"/> FALSE	BOOL		
<input checked="" type="checkbox"/>	AS.5	"DM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		
<input checked="" type="checkbox"/>	AS.4	"DM_Ein"	<input checked="" type="checkbox"/> FALSE	BOOL		Drehzahlüberwachung Benzinmotor
<input checked="" type="checkbox"/>	MW2	"BM_Drehzahl_Ist"	<input checked="" type="checkbox"/> 1500	DEZ		
<input checked="" type="checkbox"/>	DB1.DBW6	"Benzin_Preset_Speed"	<input checked="" type="checkbox"/> 1500	DEZ	<input checked="" type="checkbox"/> 1500	
<input checked="" type="checkbox"/>	AS.1	"BM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		Drehzahlüberwachung Dieselmotor
<input checked="" type="checkbox"/>	MW4	"DM_Drehzahl_Ist"	<input checked="" type="checkbox"/> 1200	DEZ		
<input checked="" type="checkbox"/>	DB2.DBW6	"Diesel_Preset_Speed"	<input checked="" type="checkbox"/> 1200	DEZ	<input checked="" type="checkbox"/> 1200	
<input checked="" type="checkbox"/>	AS.5	"DM_Soll_erreicht"	<input checked="" type="checkbox"/> TRUE	BOOL		

Beobachtmodus Zyklusende permanent Zyklusbeginn permanent Steuermodus

Beobachten sofort Beobachten Steuern Steuern sofort

In der erweiterten Sicht Beobachten/Steuern können Sie zwischen verschiedenen Beobacht- und Steuermodi auswählen und z. B. bestimmen, ob der Steuerwert

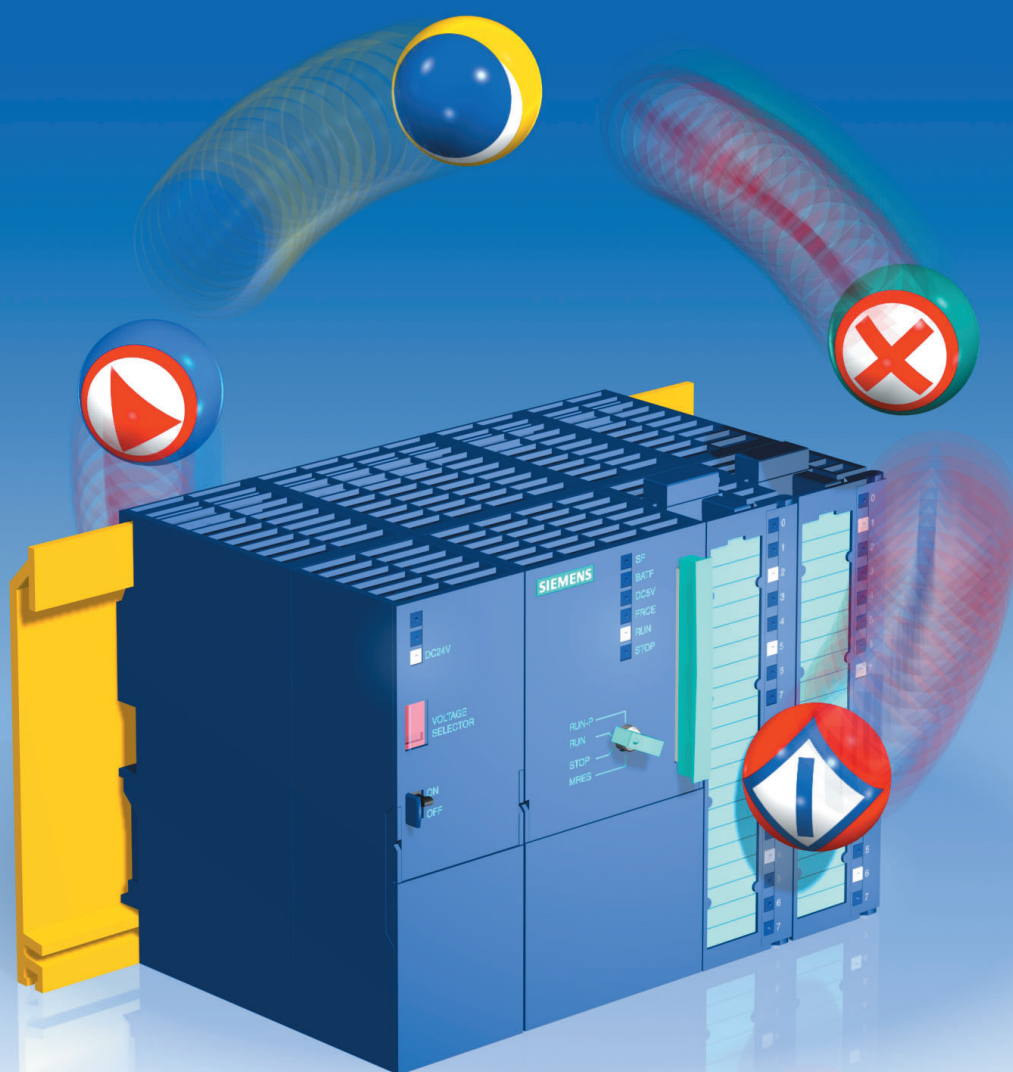
- nur bei definierten Zykluszuständen des OB 1 gesetzt werden soll, z.B. bei Zyklusbeginn, oder
- sofort, also auch mitten im Programmablauf.

9



# 12

## Fehler- diagnose



## HW-Diagnose auf einen Blick

The screenshot shows the 'Hardware' window in STEP 7 Lite. On the left, a project tree lists various components like 'CPU 315', 'SF', 'FRCE', 'RUN', 'STOP', 'MRES', 'DB3:G\_Daten', 'FC1:Lüfter', 'DB2:Diesel', 'DB1:Benzin', 'FB1:Motor', and 'OB1:CYCL\_EXC'. The main area displays a rack configuration with four racks (Rack 0 to Rack 3). Rack 0 is highlighted, showing its components: CPU 315, SM 32\* DI, and SM 32\* DO. A detailed view of the 'CPU 315' is shown on the right, including its version, order number, and status. The status is 'O.K.'. The bottom of the window shows a 'STOP' button and a status bar with 'Abs', 'Nw 1', and 'Einf'. Red lines connect numbered callouts to specific parts of the interface.

- 1 Projekt öffnen unter Datei
- 2 Online verbinden
- 3 Betriebszustand der CPU beachten
- 4 Sammelfehlermeldung Hardware beachten
- 5 Sicht: HW-Diagnose aufrufen
- 6 Diagnostizierbare Baugruppen werden angezeigt
- 7 Globale Baugruppendaten – mit Statusangaben
- 8 Detaildaten – mit näherer Fehlerspezifikation
- 9 Diagnosepuffer einblenden
- 10 Alle Angaben für eine Ersatzteilbestellung

STEP 7 Lite bietet Ihnen bei einem Hardwarefehler in der SPS-Station eine umfassende Fehlerdiagnose auf einen Blick. Die Nummern deuten Ihnen den Diagnoseweg an. Diese Sicht ist nur bei Online-Verbindung möglich.

## Fehler aufgetreten?



An einer Anlage ist die SPS-Station in STOP gegangen. Der Schlüsselschalter der CPU steht in Stellung **RUN**.

Ein Hardwarefehler liegt vor.

## Fehler beheben



Drehen Sie den Schlüsselschalter in Stellung **STOP**.

1

Öffnen Sie das zu dieser SPS-Station gehörende Projekt, in dessen Hardware-Konfiguration der Fehler aufgetreten ist.

2

Stellen Sie eine Online-Verbindung zwischen Programmiergerät und SPS-Station her (vergleiche Kapitel 10).

12.3

3

Dass die CPU in STOP gegangen ist, wissen Sie bereits. Diese Anzeigen benötigen Sie, wenn Sie z.B. während der Instandsetzungsarbeiten keinen Blickkontakt zur CPU haben.

4

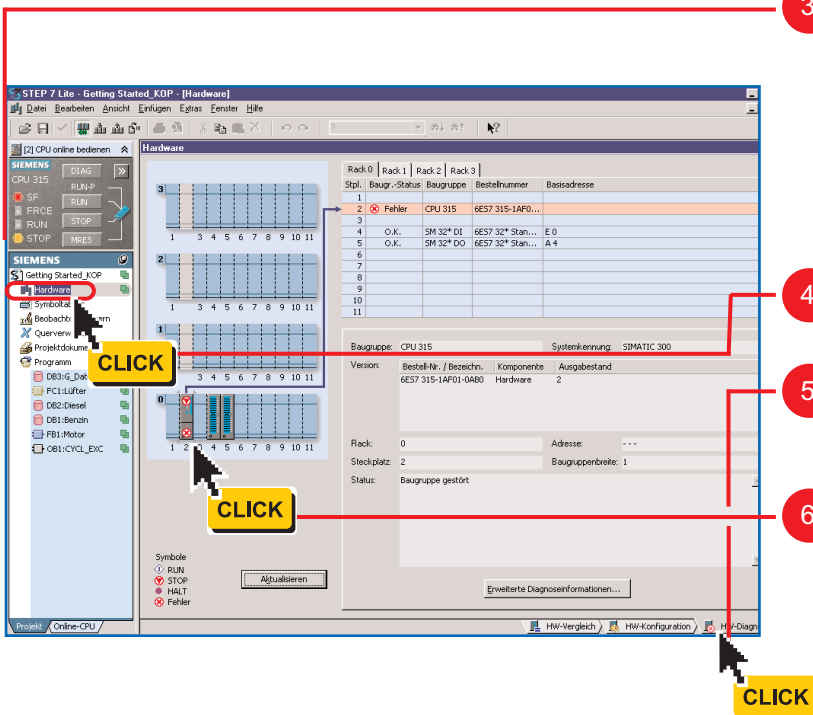
Doppelklicken Sie im Projektfenster links auf **Hardware**.

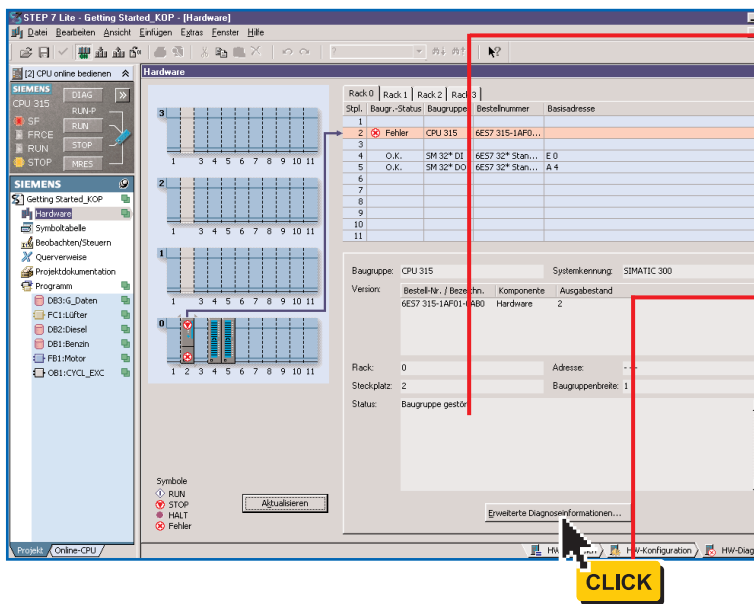
5

Öffnen Sie die Sicht **HW-Diagnose**.

6

Im Rack ist die defekte Baugruppe durch ein Piktogramm gekennzeichnet. Klicken Sie auf diese Baugruppe.





8

Hier erhalten Sie eine Klartextmeldung:

- Baugruppe O. K.
- Baugruppe gestört.

9

Benötigen Sie mehr Informationen klicken Sie bitte auf **Erweiterte Diagnoseinformationen**.

CLICK

# Baugruppenzustand und Fehlerhistorie

STOP, RUN, HALT, ...

OK, gestört, ...

Weitere Daten zur CPU aufrufen

1 Diagnosepuffer

2 Details zum markierten Ereignis im Diagnosepuffer

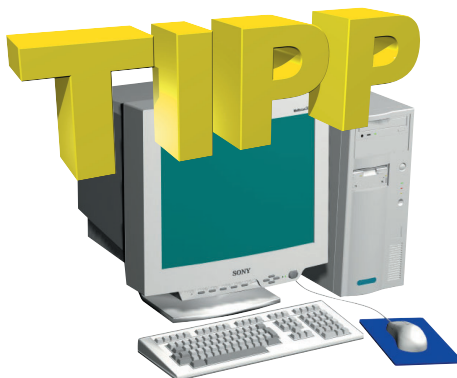
3 Hilfe zum angezeigten Ereignis

Filter für anzuzeigende Ereignisse im Diagnosepuffer

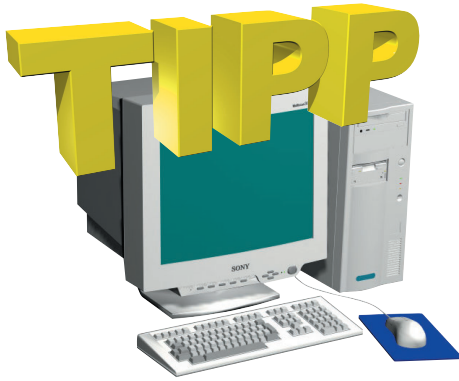
Diagnosepuffer speichern im TXT-Format

Bevor Sie eine vermeintlich defekte Baugruppe austauschen, sehen Sie hier in Funktion **Baugruppenzustand** nach (Aufruf, siehe Seite 12.4).

Der Diagnosepuffer z. B. hält alle Ereignisse in der CPU fest – nicht nur die Fehler. Diese Information benötigen Sie insbesondere dann, wenn Sie reine Folgefehler von dem zuerst aufgetretenen tatsächlichen Fehler unterscheiden müssen.



Mehr Information über **F1 > Index > Aufruf des Baugruppenzustands**.

**Bei Drahtbruch:**

Überprüfen Sie die Verkabelung oder ggf. falsch gesteckte Messbereichsmodule.

**Geht die CPU in STOP:**

Werten Sie die Meldungen aus dem Diagnosepuffer aus. Den Diagnosepuffer können Sie am schnellsten erreichen, wenn Sie im CPU-Bedienpanel auf die Schaltfläche "DIAG" klicken.

**Bei defekter Baugruppe:**

Schalten Sie vor dem Ziehen der Baugruppe die Lastspannung aus.

# 13

## Index



## A

Absolut programmieren 5.2  
Absolute Adresse 4.8, 5.2  
Adresse 0.2  
Anweisungsliste 6.2  
Ausschaltverzögerung 8.6  
Automation License Manager 1.10  
AWL 6.2

## B

Basishilfe 2.9  
Baugruppen parametrieren 4.12  
Baugruppen stecken 4.9  
Baugruppenträger 10.2  
Baugruppenzustand 12.5  
Bausteinaufruf 7.14  
Bausteinaufruf programmieren 8.9  
Bausteineditor 6.4  
Benzinmotor 3.3  
Beobachten 11.3, 11.8  
Bestellnummern 1.5  
Bibliotheken 2.5

## C

Checkliste Komponenten 1.5  
CPU-Bedienpanel 2.5  
CPU urlöschen 10.6, 10.7, 10.8

## D

Dateihandling 2.7  
Datentypen 5.6  
DB 6.5  
Diagnosepuffer 12.2  
Dieselmotor 3.3  
Direkthilfe 2.9  
Dokumentation 1.5  
Drahtbruch 12.5  
Drehzahlüberwachung programmieren 7.7

## E

Ein- und Ausschaltvorgang des Motors 7.6  
Eingang 5.4

Einsatzzweck STEP 7 Lite 0.3

## F

FB 6.5  
FC 6.5, 8.2  
Fehlerdiagnose 12.2  
Fehlerhistorie 12.5  
Fehlermeldungen 2.10  
Flipflop 1.3  
Funktion 8.2  
Funktion programmieren 8.6  
Funktionsbaustein 7.2  
Funktionsplan 6.2  
FUP 6.2

## G

Globaler Datenbaustein 9.2

## H

Hardwarekatalog 4.6  
Hilfe 2.8  
HW-Diagnose 12.2  
HW-Konfiguration 4.6  
HW-Vergleich 4.18

## I

Installation 1.8  
Instanz-Datenbausteine 7.12

## K

Konfigurationsdaten speichern 4.14  
Konfigurationsfehler 4.10  
Konfigurieren der Baugruppen 4.2  
Kontaktplan 6.2  
KOP 6.2  
Kurzhilfe 2.9

## L

Laden der Hardware-Konfiguration 4.16  
Laden des Programms 10.6  
Laden einzelner Bausteine 10.11  
License Key 1.8

## M

Motorenteststand 3.2  
MPI-Kabel 10.2

## N

Neuer Baustein 7.13  
Neues Netzwerk einfügen 6.4  
Neues Projekt 4.4

## O

Offline 4.19  
Online 4.19  
OB 6.5  
Online verbinden 10.3  
Online-Verbindung herstellen 10.4  
Operand 8.8

## P

Physik 4.19  
Parallelschaltung 1.3  
Parametrieren 4.13  
Parametrierung von Baugruppen 4.6  
Piktogramme 4.6  
Pinnadel 4.5  
Programm in CPU laden 10.9  
Programm übertragen 10.6  
Programmierbeispiele 2.3  
Programmieroberfläche anpassen 6.11  
Programmiersprache 6.2  
Programmiersprache ändern 6.4  
Programmstatus 11.2  
Projektfenster 2.4  
Pufferbatterie 10.2

## Q

Querverweise 6.24

## R

Rack 4.2  
Rechner 1.5

Referenzhilfe 2.10  
Reihenschaltung 1.3

## S

Sicherheitsanforderungen definieren 3.5  
Simulationssoftware 1.5  
Speicherglied 1.3  
Speichermodul 10.2  
Speichern 4.15, 6.8, 6.14, 6.20  
Steuern 11.9  
STOP 12.5  
Stromschiene 6.9  
Stromversorgung 10.2  
Symbole 4.20  
Symbolisch programmieren 5.4  
symbolische Darstellung 6.7  
Symboltabelle 5.5  
Symboltabelle kopieren 5.5

## T

Testaufbau 10.2  
Testen 11.4  
Testen des Programms 11.2

## U

Überblick 1.6  
Übernehmen 4.15, 6.8, 6.14, 6.20  
Umschalten symbolische/ absolute  
    Programmierung 6.4

## V

Variablen beobachten 11.6, 11.8  
Variablen steuern 11.9  
Variablendeklarationstabelle 8.6  
Variablendeklarationstabelle ausfüllen 7.4  
Variablentabelle erstellen 11.7  
Verbindungskamm 10.2  
Vergleich Online/Offline 4.17

## Z

Zeitfunktion 8.6  
Zeitfunktion programmieren 8.7







**6ES7810-3CC07-0YA05**

Order No. 6ES7810-3CC07-0YA05

**Siemens Aktiengesellschaft**

Bereich Automation and Drives  
Geschäftsgebiet Industrial Automation Systems  
Postfach 4848, D-90327 Nürnberg

[www.siemens.com/automation](http://www.siemens.com/automation)