

Vorwort

Die Komplexität heutiger Automatisierungsaufgaben ist enorm gestiegen. Es stellt sich die Frage, ob die klassischen Programmiersprachen FBS (FUP), KOP und AWL ausreichend sind, um übersichtliche, leicht wartbare Programme zu schreiben. Mit der Hochsprache SCL für «Structured Control Language» ist dies leichter möglich. Die Programmiersprache STEP 7-SCL ist eine höhere Programmiersprache. SCL basiert auf der Sprache ST (Strukturierter Text) der Norm IEC 61 131-3.

SCL enthält neben Hochsprachenelementen auch typische Elemente der SPS wie Eingänge, Ausgänge, Zeiten, Bausteinaufrufe usw. als Sprachelemente. Das SCL-Programm besteht aus einer Folge von Anweisungen, die mit einem Strichpunkt abschließen. Schleifen, wie FOR... bis REPEAT..., sowie Auswahlanweisungen, wie IF... und CASE..., sind in dieser Sprache möglich.

Bausteine, die in SCL programmiert wurden, können auch in den anderen IEC-Sprachen wie AWL, FBS (FUP) und KOP verwendet werden.

Dieses Buch wendet sich an PLC(SPS)-Programmierer sowie Schüler und Studenten an beruflichen Gymnasien, Technikerschulen und Hochschulen, die das Programmieren mit SCL im TIA-Portal erlernen wollen. Alle Anwendungsschritte im Umgang mit dem TIA-Portal werden detailliert dargestellt.

In diesem Workshop werden die Grundlagen an einfachen Beispielen aus der Praxis erarbeitet. Das Wiederverwenden der Programmbausteine steht dabei im Vordergrund. Es werden nicht nur klassische Programmentwurfsmethoden vorgestellt, sondern mit Hilfe der künstlichen Intelligenz KI – z. B. ChatGPT, Gemini, Claude oder Copilot – Programme für einige Aufgaben und Übungen entworfen.

Vorlagen und Lösungen der Aufgaben und Übungen für das jeweilige TIA-Portal Projekt finden Sie auf der immer wieder aktualisierten Webseite <https://vogel-professional-education.de> unter **InfoClick**. Die Archivdateien der Vorlagen enthalten notwendige Bausteine sowie die Variablendeklarationen. Weiterhin finden Sie in der Vorlage eine Anlagensimulation mit deren Visualisierung, um Ihr erstelltes Programm auch ohne Hardware-CPU und eine reale Anlage testen zu können. Sollten Sie eine höhere TIA-Portalversion besitzen, so können Sie die Projektvorlagen aus **InfoClick** benutzen und in Ihrem TIA-Portal beim Öffnen bzw. Dearchivieren hochrüsten. Sie finden Animationen und als Lernzielkontrolle Tests zu den einzelnen Kapiteln. Sie können sich nach Registrierung eine kostenlose Trial-Version von STEP7 Professional und PLCSIM, die zeitlich begrenzt ist (21 Tage), von der Webseite der Firma Siemens im TIA-Portal herunterladen.

Alle Aufgaben und Übungen wurden ausschließlich aus didaktischer Sicht erstellt und können so nicht als Lösungen für Anlagen verwendet werden.

Ich wünsche Ihnen viel Freude beim Erlernen der SCL-Programmierung im TIA-Portal und bei der Anwendung Ihres neu erworbenen Wissens.

Soden

Karl Schmitt

Inhaltsverzeichnis

Vorwort	5
1 Einführung	11
1.1 Das TIA-Portal	11
1.2 Speicherprogrammierbare Steuerung (SPS), Programmable Logic Controller (PLC)	11
1.3 Sprachen	12
1.4 Programmorganisationseinheiten (POEs)	12
1.5 Variable, Datentypen	12
1.6 Prozessabbild, Merker, Datenbausteine, Remanenz	13
1.7 PLC-Simulation (PLCSIM)	14
1.8 Visualisierung, HMI	14
1.9 Anlagensimulationen.	14
2 Verknüpfungssteuerung ohne Speicherverhalten	15
2.1 TIA-Projekt analysieren und testen	15
<i>Aufgabe 2.1: Funktion 2 aus 3</i>	15
2.1.1 TIA-Projekt analysieren.	16
2.1.2 TIA-Projekt testen	18
2.2 TIA-Projekt erweitern	22
<i>Aufgabe 2.2: Funktion 2 aus 3, Erweiterung</i>	22
3 Programmorganisationseinheiten, Gliederung	25
3.1 Gliederung mit OBs – Program cycle	25
<i>Aufgabe 3.1: Überwachung mit Gliederung</i>	25
3.2 Gliederung mit einem Hauptprogramm und Unterprogramme. ...	27
4 Bibliotheksfähige, parametrisierbare Funktionen erstellen	29
<i>Aufgabe 4.1: Funktion 2 aus 3</i>	29
4.1 Programmentwurf.	29
4.2 Neues Projekt erstellen	30
4.2.1 Lösungsvariante 1 der Aufgabe 4.1: Funktion mit Rückgabewert	32
4.2.2 Lösungsvariante 2 der Aufgabe 4.1: Funktion mit Ausgangsparameter	35
4.3 Bibliotheken	36
<i>Aufgabe 4.2: Neue Bibliothek.</i>	36
<i>Übung 4.1: Funktion 2 aus 3, Erweiterung.</i>	37
4.4 Programmentwurf mit Hilfe der KI.	39
<i>Aufgabe 4.3: Funktion 2 aus 3, Entwurf mit Hilfe der KI</i>	39
5 Verknüpfungssteuerungen mit Speicherverhalten	43
<i>Aufgabe 5.1: Fördereinrichtung 1</i>	43
5.1 Unterprogramm mit IF...THEN- oder IF...THEN...ELSIF- Anweisungen.	44

5.2	Remanenz	45
5.3	Unterprogrammaufruf	46
5.4	Haltepunkte	48
5.5	Unterprogramm speichernd ein/aus mit AND und OR.	49
	<i>Übung 5.1: Fördereinrichtung 2</i>	50
	<i>Übung 5.2: Erweiterung mit einer Tankanlage</i>	52
6	Parametrisierbare Funktionsbausteine (FBs) erstellen	55
6.1	Funktionsbausteine (Funktionsblöcke)	55
	<i>Aufgabe 6.1: Fördereinrichtung mit Instanzen eines FBs</i>	55
6.2	Simulation der Anlage.	56
6.3	Funktionsbaustein erstellen und Instanzen aufrufen	57
6.4	Instanz-Datenbausteine.	60
	<i>Übung 6.1: Tankanlage mit Instanzen eines FBs.</i>	61
6.5	Flankenerkennung.	62
	<i>Aufgabe 6.2: Fördereinrichtung mit Flankenerkennung</i>	62
6.6	Temporäre lokale Variable	62
	<i>Übung 6.2: Fördereinrichtung mit Drehrichtungsumkehrsteuerung</i>	64
6.7	Zustandsgraph – eine Programmentwurfsmethode.	65
	<i>Aufgabe 6.3: Verteileinheit, Zustandsgraph</i>	65
6.8	Startup[OB100]	67
	<i>Übung 6.3: Erweiterung der Verteileinheit, Zustandsgraph</i>	68
	<i>Übung 6.4: Drehrichtungsumkehrsteuerung, Zustandsgraph</i>	68
6.9	Programmentwurf mit Hilfe der KI.	69
	<i>Aufgabe 6.4: Fördereinrichtung, Entwurf mit der KI</i>	69
	<i>Übung 6.5: Wendesteuerung, Entwurf mit der KI</i>	73
7	Eine Funktion im Funktionsbaustein aufrufen	79
7.1	Funktion mit Rückgabewert im Funktionsbaustein aufrufen	79
	<i>Aufgabe 7.1: Tankanlage mit Temperaturüberwachung</i>	79
7.2	Funktion mit Ausgangsparameter im Funktionsbaustein aufrufen	81
8	Zeiten, Datentyp Time, Instanzen und Multiinstanzen	83
8.1	Zeit-Funktionsbausteine – Timer	83
8.2	Datentyp Time, LTime, Anzeigeformat	83
8.3	Unterprogramm, Timer, Instanzen und Speicherauslastung	84
	<i>Aufgabe 8.1: Fördereinrichtung mit Laufzeitüberwachung.</i>	86
8.4	Bibliotheksfähiger Funktionsbaustein, Timer und Multiinstanzen	87
	<i>Aufgabe 8.2: Zwei Fördereinrichtungen mit Laufzeitüberwachung</i>	87
	<i>Aufgabe 8.3: Zwei Fördereinrichtungen mit Laufzeitüberwachung, Taktgeber</i>	89
	<i>Übung 8.1: Funktionseinheit Bandanlage mit Ein- und Ausschaltverzögerung.</i>	91
	<i>Übung 8.2: Fördereinrichtung mit Laufzeitüberwachung und Meldung</i>	92
	<i>Übung 8.3: Zustandsgraph, Stern-Dreieck-Anlauf</i>	93
8.5	Programmentwurf mit Hilfe der KI.	93

<i>Aufgabe 8.4 Förderband mit Laufzeitüberwachung,</i>	
<i>Programmmentwurf mit der KI</i>	94
<i>Übung 8.4 Stern-Dreick-Anlauf, Programmmentwurf mit der KI...</i>	97
9 Zählen, Datentyp Integer, CASE,	
Datentypenumwandlung	101
9.1 Datentypen für ganze Zahlen	101
9.2 Zählen.	102
<i>Aufgabe 9.1: Vorwärtszähler</i>	102
<i>Übung 9.1: Vor-/Rückwärtszähler</i>	104
<i>Aufgabe 9.2: Fördereinrichtung für mehrere Paletten</i>	105
9.3 CASE – die Fallunterscheidung	106
<i>Aufgabe 9.3: Verteileinheit, Zustandsgraphen mit CASE-Anweisung</i>	107
9.4 InOut-Parameter	108
9.5 Datentypenumwandlung	109
<i>Aufgabe 9.4: Funktion 2 aus 3 mit Datentypenumwandlung</i>	109
9.6 Programmmentwurf mit Hilfe der KI	110
<i>Aufgabe 9.5: Fördereinrichtung mit Zähler, Programmmentwurf mit der KI</i>	110
<i>Übung 9.2: Stern-Dreick-Anlauf, Programmmentwurf mit der KI</i> ..	112
<i>Aufgabe 9.6: Funktion 2 aus 3, Programmmentwurf mit der KI</i> ...	115
10 Rechnen, REAL, Visualisierung, DBs, ARRAY und Schleifen	119
10.1 Datentyp REAL, LREAL	119
10.2 Datentypenumwandlung	120
10.3 Global-Datenbaustein	120
<i>Aufgabe 10.1: Mischanlage, Lösungsvariante 1</i>	121
10.4 Visualisierung	123
<i>Aufgabe 10.2: Mischanlage, Lösungsvariante 2</i>	125
10.5 Bedingter Programmaufruf	125
<i>Aufgabe 10.3: Mischanlage, bedingter Programmaufruf, Lösungsvariante 3</i>	126
<i>Übung 10.1: Mischanlage mit FB</i>	127
<i>Aufgabe 10.4: Tankanlage, Umrechnungen von °C nach °F</i>	128
10.6 ARRAY, ein zusammengesetzter Datentyp	129
10.7 Programmschleifen	129
10.8 PLC-Datentyp – ein anwenderdefinierter Datentyp	131
<i>Aufgabe 10.5: Tankanlage, Füllvolumen</i>	132
10.9 Datentyp STRUCT	134
<i>Aufgabe 10.6: Tankanlage, Füllvolumen mit FC</i>	134
<i>Übung 10.2: FC 2 aus 3 mit Erweiterung</i>	136
10.10 Datentyp DTL (Datum und Uhrzeit Date_AND_LTIME)	138
<i>Aufgabe 10.7: Arbeitszeiterfassung</i>	138
10.11 CONTINUE, EXIT	139
<i>Aufgabe 10.8: Speicher füllen</i>	139
10.12 Programmmentwurf mit Hilfe der KI	141
<i>Aufgabe 10.9 Tankanlage °C → °F, ARRAY, Programmmentwurf mit der KI</i>	141

11	Ablaufsteuerungen, GRAFCET, CASE	145
11.1	GRAFCET	145
	Aufgabe 11.1: Ablaufsteuerung, Waschanlage mit Automatikbetrieb	145
	Aufgabe 11.2: Ablaufsteuerung, Waschanlage mit Automatik- und Handbetrieb	148
	Aufgabe 11.3: Ablaufsteuerung, Waschanlage mit CASE-Anweisung	149
	Übung 11.1: Ablaufsteuerung, Waschanlage als FB, Schrittüberwachung	151
	Übung 11.2: Ablaufsteuerung, Waschanlage mit parallelen Schrittketten	153
	Aufgabe 11.4: Ablaufsteuerung, Ampel 1 mit CASE-Anweisung.	154
	Übung 11.3 Schrittkettensteuerung, Stern-Dreieck-Anlauf.	156
	Übung 11.4 Schrittkettensteuerung, Verteileinheit	156
11.2	Programmwurf mit Hilfe der KI	156
	Aufgabe 11.5 Ampelsteuerung, Programmwurf mit der KI	156
12	Analogwertverarbeitung	163
	Aufgabe 12.1: Tankanlage, Füllvolumen, AI-Baugruppe	163
	Aufgabe 12.2: Tankanlage, AQ-Baugruppe	166
12.1	OB-Baustein Cyclic interrupt, Weckalarm-OBs	168
	Aufgabe 12.3: Tankanlage, Cyclic interrupt	168
	Aufgabe 12.4: Tankanlage, gleitende Mittelwertbildung.	169
12.2	Arrays von Multiinstanzen und PLC-Variablen	172
	Aufgabe 12.5 Tankanlage mit Temperaturregelung	172
	Übung 12.1: Bandsteuerung	179
13	Byte- und Word-Verarbeitung, Strings	183
13.1	Datentyp für logische Operationen	183
	Aufgabe 13.1: Temperaturüberwachung, 2 aus 3	183
13.2	Datentyp STRING – eine Zeichenkette.	185
13.3	Slice-Zugriff	186
	Aufgabe 13.2: Ablaufsteuerung Ampel 2	190
	Aufgabe 13.3: Ablaufsteuerung, Waschanlage mit Bytezugriff	191
13.4	Schrittkeite mit Funktionsaufrufen und einem globalen Datenbaustein als Datenspeicher	193
	Aufgabe 13.4: Regallager	193
14	Quellen, Trace	199
14.1	Quellen mit einem beliebigen ASCII-Editor schreiben.	199
14.2	Externe Quelle importieren und übersetzen	199
14.3	Vorhandene SCL-Bausteine in eine Quelldatei umwandeln.	199
	Aufgabe 14.1: PT1-Glied, Verzögerungsglied	199
14.4	Trace	201
	Aufgabe 14.2: Sinus-Generator.	203
	Übung 14.1: Filter, Tiefpass	204

15	Digitale Regelungen	207
15.1	Abtastregelung	207
15.2	Regelkreisglieder	208
	<i>Aufgabe 15.1: Streckensimulationen einer Füllstandsregelstrecke.</i>	210
	<i>Aufgabe 15.2: Streckensimulation einer Temperaturregelstrecke</i>	212
	<i>Aufgabe 15.3: Temperaturregelung mit einem P-, PI- und PID-Regler</i>	214
	<i>Aufgabe 15.4: Temperaturregelung mit einem PID-Siemens-Regler</i>	219
	<i>Aufgabe 15.5: Füllstandsregelung mit einem PI-Regler, Peripheriewerte</i>	222
15.3	Zweipunktregelung	224
	<i>Aufgabe 15.6: Temperaturregelung mit einem Zweipunktregler.</i>	224
	<i>Übung 15.1: Füllstandsregelung mit einem Zweipunktregler</i>	225
15.4	Fuzzy, Fuzzyregler	226
	<i>Aufgabe 15.7: Füllstandsregelung mit einem Fuzzy-Regler.</i>	226
	<i>Übung 15.2: Temperaturregelung mit einem Fuzzy-Regler.</i>	233
15.5	Programmwurf von Bausteinen aus der Regelungstechnik mit Hilfe der KI.	233
	<i>Aufgabe 15.8 Temperaturregelung, Programmwurf mit der KI.</i>	233
16	Kommunikation, Netzwerke	241
16.1	PROFINET	241
16.2	Projekt in die CPU laden	242
16.3	Vernetzung und die Verbindung zu einem HMI	243
16.4	Dezentrale Peripherie	243
16.5	IO-Link	245
16.6	Intelligentes IO-Device	245
16.7	PROFIBUS	246
16.8	AS-Interface (AS-i)	247
	16.8.1 AS-i mit den CPUs 1200 und 1500	247
	16.8.2 AS-i mit den CPUs300	249
16.9	Open User Communication	251
16.10	Punkt-zu-Punkt-Kommunikation (PtP-Kommunikation)	251
Anhang		253
A.1	Lösungen der Aufgaben und Übungen	253
A.2	Überblick der SCL-Anweisungen	339
A.3	Beschreibungskopf	342
A.4	Startseite InfoClick	343
A.5	Projekte	347
Glossar		351
Stichwortverzeichnis		355

1 Einführung

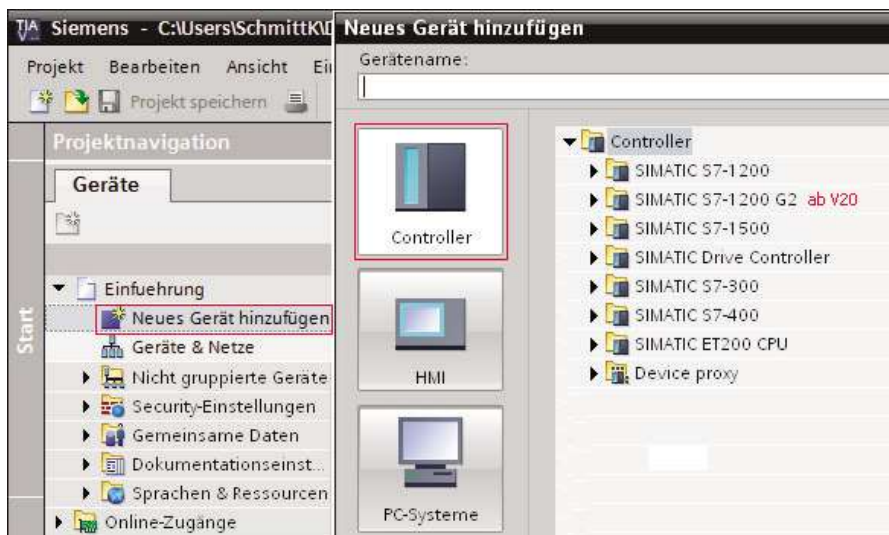
1.1 Das TIA-Portal

Das TIA-Portal ist ein Engineering-Framework von der Firma Siemens, das STEP7 enthält. Mit STEP 7 können Sie SIMATIC-Steuerungen konfigurieren, programmieren und eine Diagnose des Systems durchführen. Sie können weitere Module im TIA-Portal installieren, z.B. WinCC Advanced, um Bedien- und Beobachtungsgeräte zu projektieren, sowie Startdrive, um Antriebe zu projektieren.

1.2 Speicherprogrammierbare Steuerung (SPS), Programmable Logic Controller (PLC)

Es stehen unterschiedliche Systeme zur Verfügung:

- ❑ **Controller**, Hardware-PLCs: Diese arbeiten mit einem eigenen echtzeitfähigen Betriebssystem und sie stellen den Speicher für die Daten zur Verfügung, z.B. CPU 3xx, CPU 4xx, CPU 12xx, CPU12xx G2, CPU 15xx oder ET 200 CPUs. Sie können Ein- und Ausgangsbaugruppen mit digitalen und analogen Ein- bzw. Ausgängen (I/Os), Kommunikations- und Technologiemodule mit der CPU verbinden.
- ❑ **HMIs**: (*Human Machine Interface*) Bedien- und Beobachtungsgeräte. *HMI-Steuerungen*: Die Steuerung und das Bedien- und Beobachtungsgerät sind in einem Gehäuse eingebaut.
- ❑ **PC-basierte Steuerung**, eine Software-PLC: Eine Software, die auf einen Industrie-PC (IPC) oder auf ein Embedded-PC läuft, z.B. SIMATIC Box PC. Diese Art der Steuerung ist besonders geeignet, wenn verschiedene Aufgaben wie Datenverarbeitung, Kommunikation, Visualisierung und Steuerung auf einem PC integriert werden sollen.



In der Gerätekonfiguration stellen Sie das Automatisierungssystem zusammen.

Anmerkungen: ☐ Die Bilder im Buch können abhängig von der TIA-Portalversion geringfügig von der verwendeten Version abweichen.
☐ Projekte mit einer CPU12xx können Sie ab Firmware Version 4.1 mit PLCSIM ab V13 SP1 testen. Projekte mit der CPU12XX G2 können Sie ab TIA-Portal V20 projektieren.

1.3 Sprachen

Sie können bei den CPUs 15xx, CPUs 4xx und CPUs 3xx in den Sprachen FUP, KOP, SCL, AWL und GRAPH- die Ablaufsprache programmieren. Die CPU 12xx und die CPU 12xx G2 sind nicht in AWL und nicht in GRAPH programmierbar. In den folgenden Kapiteln lernen Sie die Hochsprache SCL und Programmentwurfsmethoden kennen.

Beachten Sie: Wenn Sie ein neues Projekt erstellen, entfernen Sie zunächst den Baustein Main[OB1], da dieser standardmäßig nur in den Sprachen FUP und KOP zu programmieren ist. Fügen Sie einen neuen Organisationsbaustein (OB) *Program cycle* in der Sprache SCL ein.

1.4 Programmorganisationseinheiten (POEs)

Nach IEC 61 131-3 gibt es drei Programmorganisationseinheiten: das Programm, die Funktion und den Funktionsblock, um umfangreiche Programme gliedern und strukturieren zu können. Das Programm wird dadurch leichter lesbar, es ist leichter zu testen und es ist leichter zu erweitern. Im TIA-Portal werden die POEs Programmbausteine genannt, wobei die Organisationsbausteine (OBs) die Programme im Sinne der IEC 61 131-3 sind.

Wie das Programm vom Echtzeit-Betriebssystem der Steuerung bearbeitet, d.h. aufgerufen wird, wird nach IEC 61 131-3 in der **Taskkonfiguration** (*task* = Arbeit, Aufgabe) festgelegt. Im TIA-Portal konfigurieren Sie die Task, indem Sie den richtigen Organisationsbaustein auswählen. Das Hauptprogramm ist standardmäßig der **Organisationsbaustein Main[OB1]**. Dieses wird vom Betriebssystem der PLC freilaufend **zyklisch** aufgerufen, und seine Anweisungen werden der Reihe nach (*sequenziell*) abgearbeitet.

In den **Funktionen** (FCs) und den **Funktionsbausteinen** (FBs) werden Teile der Anlage oder Maschine programmiert. Damit diese öfter im Programm verwendet werden können, werden Sie parametrisierbare FCs und FBs erstellen.

1.5 Variable, Datentypen



Definition

Variable sind Platzhalter für Daten; diese werden im Programm gelesen, beschrieben oder gelesen und beschrieben. Bei deren Deklaration wird ein Name, ein Datentyp und bei PLC-Variablen zusätzlich eine Ein-/Ausgangsadresse oder Merkeradresse zugeordnet.

Variablenamen sollten nur alphanumerische Zeichen und den Unterstrich, keine Umlaute (ä, ö, ü) und andere landesspezifische Sonderzeichen enthalten. In diesem Buch wurde als Präfix für den Variablenamen die Betriebsmittelkennzeichnung des Anlagenbauteils benutzt, z. B. P1_Uebertemp für eine Anzeige. Weiteres dazu auch unter **InfoClick**.

Variable deklarieren (= bekanntgeben) Sie in der PLC-Variablentabelle oder in Bausteinen.

Mit dem Datentyp werden die Codierung, die Darstellung, der Wertebereich, die Größe des notwendigen Speichers sowie gültige Operationen und Funktionen der Variablen festgelegt. Der Datentyp **Bool** kann den Wert TRUE oder FALSE annehmen. Weitere Datentypen lernen Sie in den folgenden Kapiteln kennen.

1.6 Prozessabbild, Merker, Datenbausteine, Remanenz

Damit den Anweisungen in den POEs auch Daten zur Verfügung stehen, ist ein Speicher für deren Werte notwendig.

Definition



In der CPU wird der Speicherbereich, in den die Daten der Eingangsbaugruppen am Anfang jedes Zyklus kopiert werden, **Prozessabbild der Eingänge (PAE)** genannt. Der Speicherbereich, von dem die Werte am Ende eines Zyklus in die Ausgangsbaugruppe kopiert werden, wird **Prozessabbild der Ausgänge (PAA)** genannt.

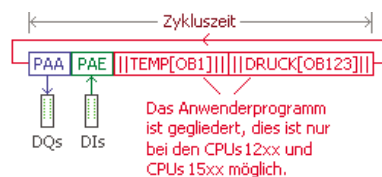
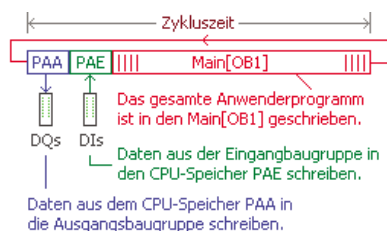
Das Prozessabbild ist notwendig, damit im Programm die Daten in einem Zyklus konsistent sind, d.h., am Anfang und am Ende der Programmabarbeitung ist der Wert einer Variablen aus dem PAE gleich.

Definition



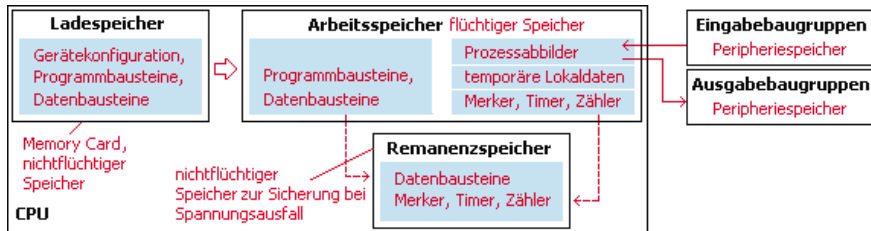
Die Zeit, die vergeht, bis das Prozessabbild und alle Anweisungen bearbeitet wurden, wird **Zykluszeit** genannt.

Ihr Wert ist von der Anzahl, dem Typ der Anweisungen und von der verwendeten CPU abhängig; er liegt im Bereich von einigen Millisekunden.



Für die Daten, die nicht im Prozessabbild liegen, stehen in der CPU der Merkerspeicherbereich, die Datenbausteine und der Speicher für die temporären Lokaldaten zur Verfügung.

Die Größe der Speicherbereiche ist vom CPU-Typ abhängig.



Die Variablen, die auf das Prozessabbild über die Adressen %I..., %Q... und den Merkerspeicherbereich über die Adressen %M... zugreifen sollen, werden in der PLC-Variablentabelle deklariert.

Einen Teil des Merkersbereichs sowie Datenbausteine und SIMATIC-Timer und -Zähler können Sie dem remanenten Speicherbereich zuordnen. Daten, die im **remanenten Speicher** liegen, haben nach einem Neustart, z.B. nach einem **Ausfall der Versorgungsspannung**, den gleichen Wert wie vor dem Spannungsausfall.

1.7 PLC-Simulation (PLCSIM)

Projekte mit einer CPU15xx können ab PLCSIM V12, Projekte mit einer CPU12xx ab PLCSIM V13 SP1 und Projekte mit einer CPU3xx mit PLCSIM V5 getestet werden. Sie benötigen keine echte CPU. Sie können auch eine ältere Version PLCSIM benutzen (z.B. V15), um ihr TIA-Portal-Projekt (z.B. V21) zu testen.

1.8 Visualisierung, HMI

Sie können eine Anlagenvisualisierung erstellen, über die Sie Ihre Anlage an einem HMI bedienen und beobachten können.

1.9 Anlagensimulationen

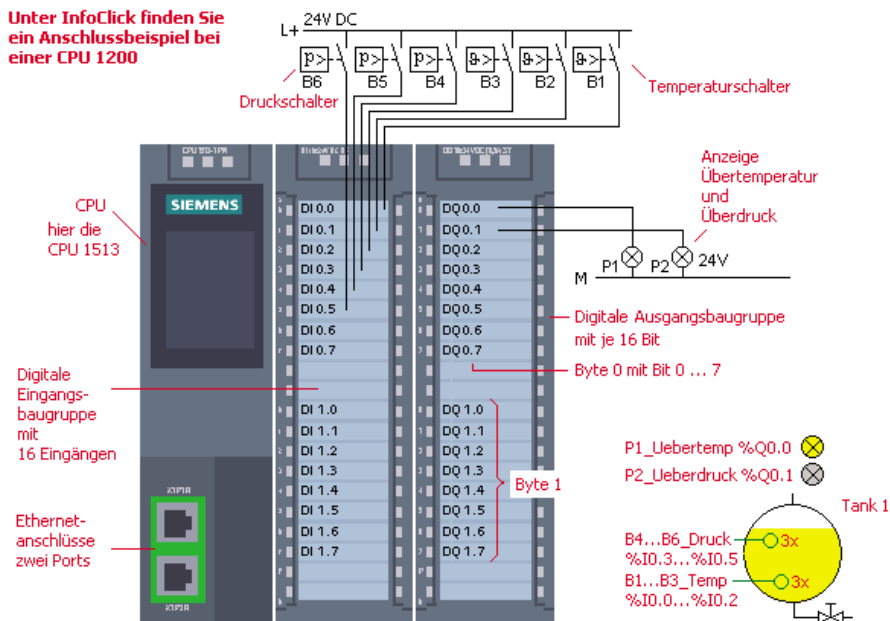
In den meisten Projektvorlagen für das TIA-Portal finden Sie ab Kapitel 6 Bausteine und Visualisierungen, mit deren Hilfe Sie die Anlage am PC simulieren, bedienen und beobachten können.

2 Verknüpfungssteuerung ohne Speicherverhalten

In diesem Kapitel werden Sie ein Projekt im TIA-Portal analysieren und erweitern. Sie lernen den prinzipiellen Aufbau einer PLC (SPS), die Gerätekonfiguration, die PLC-Variablendeklaration und den Programmierer in der Programmiersprache SCL kennen. Sie erweitern das Projekt.

In einem Behälter, in dem die Temperatur überwacht wird, soll zusätzlich der Druck überwacht werden. Es sind noch drei Druckschalter einzubauen. Sprechen zwei oder alle drei Temperatur- bzw. Druckschalter an, so erfolgt eine Meldung «Übertemperatur» bzw. «Überdruck».

Unter InfoClick finden Sie ein Anschlussbeispiel bei einer CPU 1200



2.1 TIA-Projekt analysieren und testen

Aufgabe 2.1: Funktion 2 aus 3

Analysieren und testen Sie zunächst das bestehende Projekt, um dann das Projekt in der nächsten Aufgabe zu erweitern.

InfoClick

Laden Sie die Vorlage A2.1 oder A2.1_1200 «2_2aus3_SCL.zap1x» der TIA-STEP7-Archivdatei von der InfoClick-Webseite unter <https://vogel-professional-education.de/infoclick/> herunter. Sie finden in der Spalte *Hilfen* unter *Gerätekonfiguration 1214C* Hinweise zum Anschluss.

2.1.1 TIA-Projekt analysieren

Vorgehensweise

1. Projekt im TIA-Portal dearchivieren und öffnen

Dearchivieren (extrahieren) Sie die STEP7-Archivdatei A2.1 im TIA-Portal über Menü *Projekt Dearchivieren*, ab Version 16 wählen Sie Menü *Projekt Öffnen*. Sie können jetzt im TIA-Portal das bestehende Projekt analysieren. Schalten Sie das TIA-Portal gegebenenfalls im TIA-Portal-Fenster unten links in die **Projektansicht**.

2. Gerätekonfiguration analysieren

Die tatsächliche Hardware muss mit der Gerätekonfiguration übereinstimmen. **Überprüfen Sie** im Projekt die Gerätekonfiguration. Welche CPU, welches Digitaleingabemodul und Digitalausgabemodul sind eingebaut?

Doppelklicken Sie die **Gerätekonfiguration**. Im sich öffnenden Fenster *Gerätesicht* finden Sie die CPU mit einer DI- und DQ-Signalbaugruppe. Bei den CPUs 1200 und auch 1200 G2 sind diese integriert, bei der CPU1214 14 DIs und 10 DOs. Sie können Informationen abfragen. Markieren Sie dazu die jeweilige Baugruppe und wählen Sie im Kontextmenü (rechte Maustaste) *Eigenschaften...*; Sie finden eine kurze Beschreibung.

Geräteübersicht

Baugruppe	E-Adres...	A-Adres...	Typ
PLC_1	0	1	CPU 1513-1 PN
PROFINET-Schnittstelle_1	0	1 X1	PROFINET-Sch...
DI 16x24VDC HF_1	0	2	DI 16x24VDC ...
DQ 16x24VDC/0.5A ST_1	0	3	DQ 16x24VDC...

Ein- und Ausgangs-adressbytes. Sie benötigen die Information in der PLC-Variablen-tabelle

Unter links Wechsel zwischen Portal/Projektansicht

In der Geräteübersicht werden die Speicher-Adressbytes angezeigt. *Beispiel:* E-Adresse 0...1 besagt, die Bytes 0 und 1 stehen zur Verfügung, das sind maximal 16 Inputs %I0.0 ... %I0.7 und %I1.0 ... %I1.7. Diese Informationen sind für die Programmierung wichtig. Ein Byte hat 8 Bits, z.B. %I0.0...0.7. Bei der Aus-

wahl der DI/DO-Baugruppen ist auf die Anzahl der Ein- und Ausgänge, die Spannung, die Stromart sowie die Belastbarkeit eines Ausganges zu achten. Liegt bei der vorhandenen DI-Baugruppe am Eingang eine Spannung von 11...30 V an, so wird diese Spannung als logisch 1, bei Spannungen von -30...+5 V als logisch 0 interpretiert. Der Eingangsstrom beträgt typ. 2,5 mA. Bei der DO-Baugruppe darf der Ausgangsstrom 0,5 A betragen. Der Summenstrom je Baugruppe darf bei 40 °C 4 A nicht überschreiten. Die Adressen der Ein- und Ausgänge werden abhängig vom Steckplatz automatisch vom System vergeben.

3. Variable in der PLC-Variablen-tabelle, Variablen sind Platzhalter

Einer PLC-Variablen wird ein Name, ein Datentyp, eine Ein-/Ausgangsadresse oder Merkeradresse zugeordnet. Als Präfix für den Namen wurde die Betriebsmittelkennzeichnung (hier B1 ...) benutzt. Variable mit dem Datentyp Bool können den Wert TRUE oder FALSE annehmen. TRUE steht für logisch 1, FALSE für logisch 0.

Standard-Variablen-tabelle

	Name	Datentyp	Adresse	Kommentar
1	B1_Temp	Bool	%I0.0	Temp.schalter
2	B2_Temp	Bool	%I0.1	Temp.schalter
3	B3_Temp	Bool	%I0.2	Temp.schalter
4	P1_Uebertemp	Bool	%Q0.0	Anzeige
5	<Hinzufügen>			
6				

4. Baustein, Programm in SCL analysieren

Programmbausteine sind Teile des Anwenderprogramms. Sie können in verschiedenen Programmiersprachen erstellt werden.

Baustein-schnittstelle

```

1 //Funktion 2 aus 3 — Zeilenkommentar
2 //Version 1.0, Autor: Schmitt
3 "Pl_Uebertemp":="B1_Temp" AND "B2_Temp"
4 OR "B1_Temp" AND "B3_Temp"
5 OR "B2_Temp" AND "B3_Temp";

```

Öffnen Sie den Baustein MAIN[OB1]. Welche Aufgabe hat der Programmbaustein? Betätigen Sie die Funktionstaste F1 (Hilfe), um das Informationssystem zu öffnen. Wählen Sie Index und geben Sie Main ein ... Das Main-Programm wird freilau- fend zyklisch vom Betriebssystem der PLC aufgerufen und die Anweisungen werden sequenziell abgearbeitet.

Im Menü *Ansicht Anzeigen mit...* können Sie die symbolische Darstellung aktivieren oder deaktivieren. Bei der Darstellung – symbolisch und absolut – werden die Namen und die Adressen der Variablen im Editor dargestellt.



Merksatz

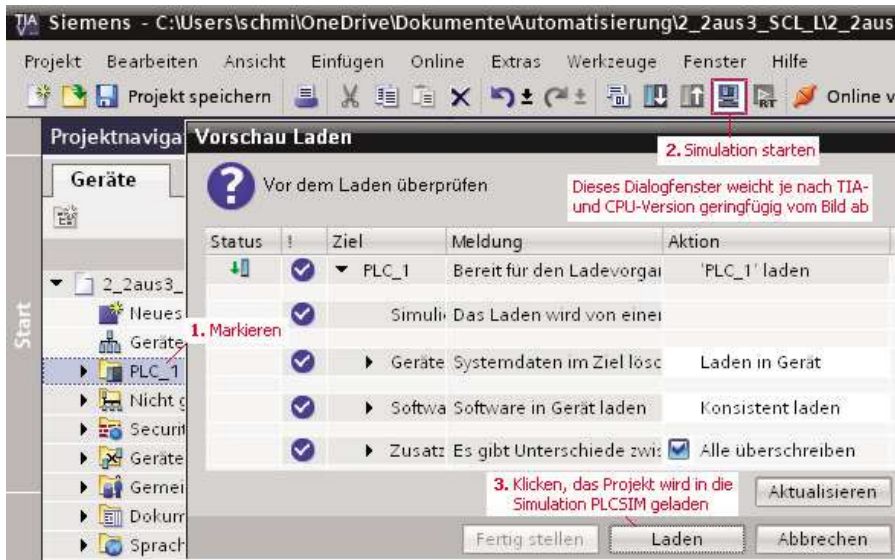
Eine Anweisung im SCL-Programm schließt immer mit einem Semikolon ab.

2.1.2 TIA-Projekt testen

Haben Sie PLCSIM (die PLC-Simulation) installiert, so können Sie Ihr Programm ohne Hardware-CPU testen. Sie können auch eine ältere Version PLCSIM benutzen (z.B. V15), um Ihr TIA-Portal-Projekt (z.B. V21) zu testen. Markieren Sie den Ordner PLC_1 im TIA-Projektnavigationsfenster und starten Sie die Simulation über Menü *Online – Simulation – Starten*. Klicken Sie in den neueren TIA-Portalversionen im Dialogfenster *Verbindung zum Gerät herstellen* auf *Verbindung herstellen*. Achten Sie darauf, dass die PLCSIM-CPU eingeschaltet ist. Aktivieren Sie, wenn nötig, die Simulationsunterstützung. Es öffnet sich das PLCSIM-Fenster.



Nach kurzer Zeit können Sie über das Dialogfenster *Verbindung herstellen* und dann *Vorschau Laden* bzw. *Erweitertes Laden* (eventuell *Suche starten* klicken) ihr TIA-Projekt in die PLC-Simulation laden. Aktivieren Sie eventuell die Kontrollkästchen «Alle überschreiben», klicken Sie «Laden», dann «Baugruppe starten» und klicken Sie «Fertigstellen». Achten Sie darauf, dass die CPU auf RUN steht.



Wenn Sie mit einer TIA-Version älter als V14 arbeiten, so wählen Sie im Dialogfenster *Erweitertes Laden* die Schnittstelle PN/IE und PLCSIM 1200/1500 aus, eventuell «Suche starten» und klicken Sie auf «Laden». Aktivieren Sie auch hier die Kontrollkästchen «Alle überschreiben», «Alle starten» und laden Sie das TIA-Projekt in die Simulation mit «Fertig stellen».

1. Testen mit PLCSIM und einer Beobachtungstabelle

Nachdem das Projekt in PLCSIM geladen wurde, können Sie in einer **Beobachtungstabelle** die Variablen steuern und die Funktion des Programms beurteilen. Fügen Sie im TIA-Portal über die Projektnavigation PLC_1 eine **Beobachtungstabelle** hinzu und wählen Sie in der Spalte Namen die Variablen aus, die Sie steuern und beobachten wollen. Das folgende Bild zeigt die Vorgehensweise.



2. Testen mit PLCSIM und im Programmbaustein

Sie können ab Version 13 Ihr Programm auch im Programmbaustein testen, wenn Sie das automatische Prozessabbild abschalten, wie das folgende Bild zeigt.

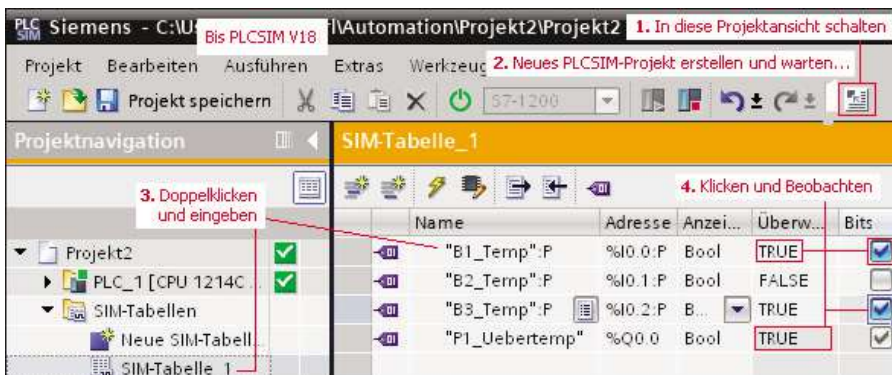


Laden Sie die Änderung in PLCSIM. Öffnen Sie den Programmbaustein Main[OB1] und klicken Sie im Bausteifenfenster auf das Icon «Beobachten ein», wie das folgende Bild zeigt. Markieren Sie die Variable, die Sie steuern wollen, und wählen Sie im Kontextmenü "Steuern ... auf 1 oder 0" und beurteilen Sie das Ergebnis von z.B. "P1_Uebertemp". Stellen Sie danach das Prozessabbild wieder auf «Automatisch».



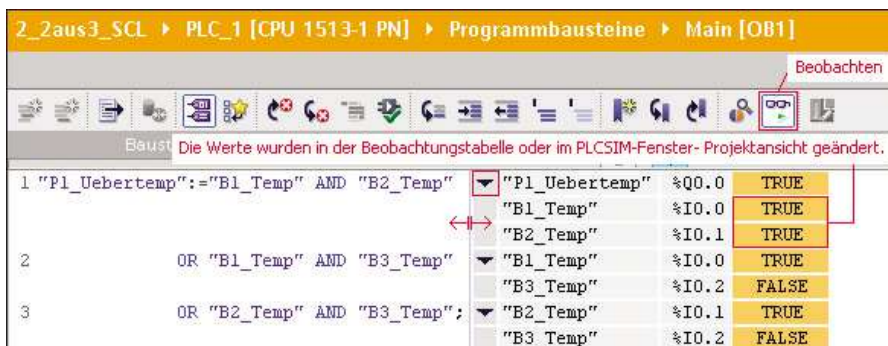
3. Testen über eine SIM-Tabelle in PLCSIM

Sie können auch im PLCSIM-Fenster die Funktion des Programms testen. Den folgenden beiden Bildern können Sie – abhängig von der PLCSIM-Version – die Vorgehensweise entnehmen. Starten Sie eventuell die CPU im PLCSIM-Fenster. Beobachten Sie den Überwachungswert der Ausgangsvariablen, hier "P1_Uebertemp".





Im Editor des TIA-Portal Fensters können Sie jetzt auch über Menü *Online Beobachten* den «Signalverlauf» beobachten, wenn Sie die Variablenwerte in PLCSIM ändern.



Sie können jetzt auch die PLC-Variablentabelle öffnen und Menü *Online Alle beobachten* wählen. Aktivieren Sie eventuell unter Menü *Extras Einstellungen* ... *SCL Erweiterte Statusinformationen erstellen*.

2.2 TIA-Projekt erweitern

Aufgabe 2.2: Funktion 2 aus 3, Erweiterung

Erweitern Sie das vorhandene Projekt mit drei Druckschaltern.

Sprechen zwei oder alle drei Druckschalter an, so erfolgt eine Meldung «Überdruck».

Vorgehensweise

1. Legen Sie die Eingänge und den Ausgang für die Drucküberwachung fest, deklarieren Sie die notwendigen Variablen.

Beachten Sie: **Variablen**namen dürfen nach IEC 61 131-3

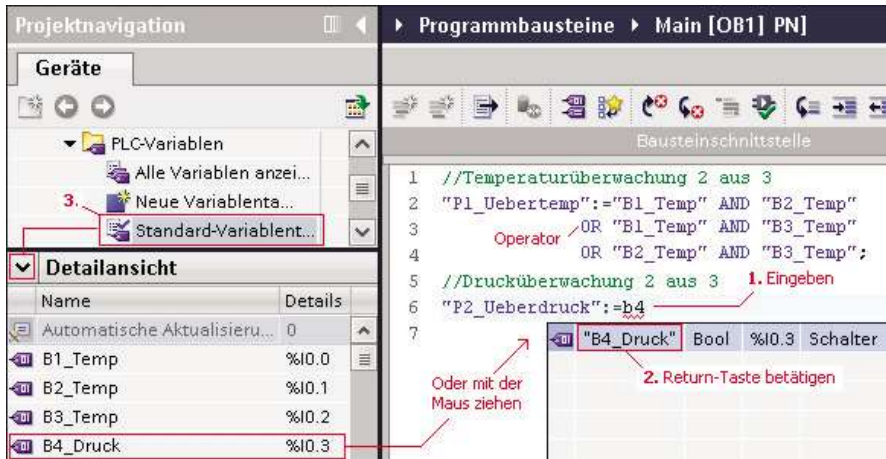
- ☐ nur alphanumerische Zeichen und den Unterstrich, keine Umlaute (ä, ö, ü) und andere landesspezifische Sonderzeichen enthalten;
- ☐ sie müssen mit einem Buchstaben oder einem Unterstrich beginnen.



2. Erweitern Sie im Editor den Programmcode. Wenn Sie die Variablen deklariert haben und im Editor die ersten Zeichen des Variablennamens eingeben, so können Sie aus der erscheinenden Tabelle die Variable auswählen. Vergessen Sie nicht am Ende der Anweisung das Semikolon; beachten Sie, dass das **Zuweisungszeichen** der Doppelpunkt und das Gleichheitszeichen ohne Leerzeichen ist. Die Variablen sind in diesem Beispiel mit den **Operatoren** AND und OR verknüpft. Im Ausdruck werden erst die AND- und dann die OR-Verknüpfung durchgeführt. Der Operator AND hat eine höhere Priorität gegenüber OR. Im Glossar finden Sie eine Übersicht der Operatorprioritäten.

Der Wert des Ausdrucks, der rechts vom Zuweisungszeichen steht, wird in die Variable links vom Zuweisungszeichen geschrieben.

Nach // können Sie einen Zeilenkommentar schreiben, Sie können Leerzeichen, Returns und Tabs zur Strukturierung Ihres Programmcodes verwenden.



Merksatz

Erst werden die AND- und dann die OR-Verknüpfung durchgeführt (siehe Glossar – Operatoren), Leerzeichen, Tabs und Returns können Sie zur Strukturierung Ihres Programmcodes benutzen. **Die Zuweisung des Wertes erfolgt rechts vom Zuweisungszeichen nach links.**

Hinweis: Im Editor hätten Sie auch die Zeilen 1 bis 4 kopieren und in Zeile 5 einfügen können. Sie hätten dann nur noch den Kommentar und die Variablennamen anpassen müssen. Im Kapitel 4 lernen Sie eine Programmierungsmethode dazu kennen.

Wenn Sie aus Sicherheitsgründen (Drahtbruch) Öffner für die Temperatur- und Druckschalter verwenden, so erweitern Sie das Programm mit dem Schlüsselwort NOT vor den Variablen B1_Temp...B6_Druck.

3. Testen Sie die Erweiterung.

Falls Sie PLCSIM nicht geschlossen haben, laden Sie Ihr erweitertes Projekt über *Menü Online-Laden in Gerät* in PLCSIM. Sie können über die Beobachtungstabelle Variablen steuern und das Ergebnis beobachten. Geben Sie die Namen der Variablen in die Beobachtungstabelle ein, aktivieren Sie «Steuerspalten einblenden» sowie «Erweiternden Modus», geben Sie den Steuerwert ein und klicken Sie auf «Beobachten ein» und «Steuern mit Triggerbedingung».



Wenn Sie mit einer älteren TIA-Portalversion als V14 arbeiten, so benutzen Sie die PLCSIM-Tabelle.

Stichpunkte

Nach der Bearbeitung des Kapitels sollten die folgenden Stichpunkte für Sie verständlich sein:

- | | | |
|--|--|--------------------------------------|
| ☐ Gerätekonfiguration | ☐ MAIN[OB1] | ☐ Operator |
| ☐ Bool | ☐ Zeilenkommentar | ☐ PLCSIM |
| ☐ Byte | ☐ Anweisung | ☐ SIM-Tabelle |
| ☐ PLC-Variable | ☐ Zuweisung | |

Selbsttest

Wählen Sie die richtigen Ergänzungen bzw. Aussagen aus.

1. PAE ist
 - ☐ ein Speicherbereich in der DI-Baugruppe.
 - ☐ ein Speicherbereich in der DO-Baugruppe.
 - ☐ ein Speicherbereich im Ladespeicher der CPU.
 - ☐ ein Speicherbereich im Arbeitsspeicher der CPU.
2. Zwischen den Variablennamen und dem Operanden muss im Programmcode

☐ ein Semikolon	☐ ein Leerzeichen
☐ zwei Schrägstriche	☐ ein Doppelpunkt

 eingefügt werden.
3. Eine SCL-Anweisung schließt immer mit einem

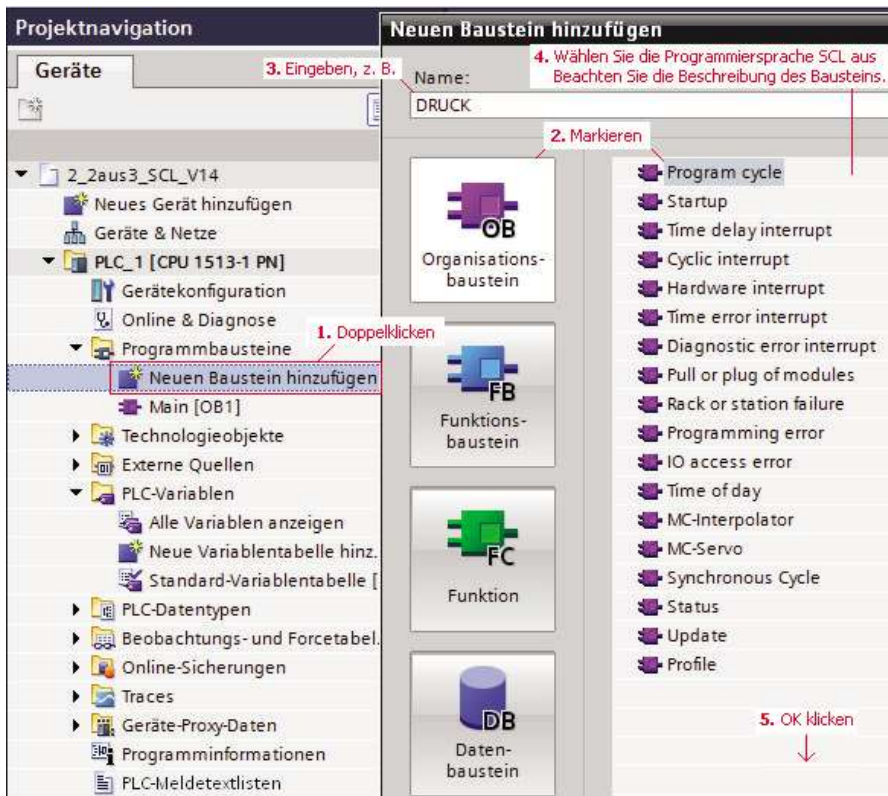
☐ Komma ab.	☐ Doppelpunkt ab.
☐ Semikolon ab.	☐ Return (Zeilenumbruch) ab.
4. Es werden
 - ☐ erst die OR- und dann die AND-Verknüpfungen durchgeführt.
 - ☐ erst die AND- und dann die OR-Verknüpfungen durchgeführt.
 - ☐ die Verknüpfungen in einer Anweisung immer von links nach rechts durchgeführt.
 - ☐ die Verknüpfungen in einer Anweisung immer von rechts nach links durchgeführt.

3 Programmorganisationseinheiten, Gliederung

Sie gliedern ein Programm.

3.1 Gliederung mit OBs – Program cycle

Wollen Sie ein umfangreiches Programm gliedern und verwenden Sie die CPU 15xx, die CPU 12xx oder auch die CPU12xx G2, so können Sie weitere OBs mit dem Namen Program cycle, die freilaufend zyklisch bearbeitet werden, in den Programmordner hinzufügen. Ihr Programm kann dann sinnvoll aufgeteilt werden, wie die folgende Aufgabe zeigt.



Aufgabe 3.1: Überwachung mit Gliederung

Teilen Sie Ihr Programm aus dem vorhergehenden Kapitel auf, um die Gliederungsmöglichkeit kennen zu lernen. Gliedern Sie in die Einheiten «Temperaturüberwachung» und «Drucküberwachung».

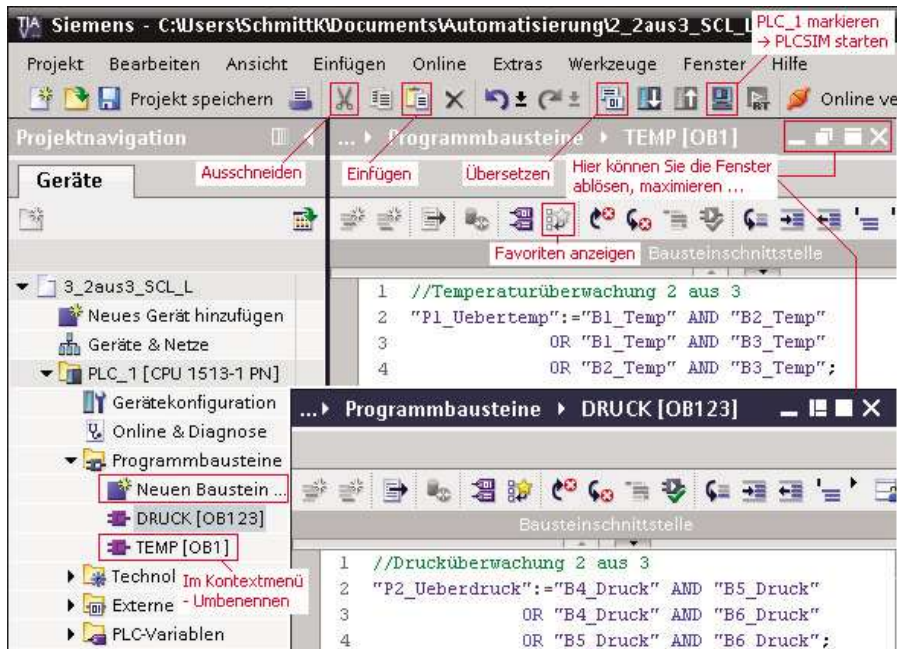
WWW

InfoClick

Sie können die STEP7-Archivdatei A3.1 (CPU 1500) «2_2aus3_SCL_L.zap1x» oder A3.1-1200 (CPU 1200) von der [InfoClick-Webseite](#) als Vorlage benutzen.

Vorgehensweise

1. Fügen Sie einen Organisationsbaustein **Program cycle** im Programmbausteinordner hinzu. Vergessen Sie nicht, die Sprache SCL zu wählen. Markieren Sie im Baustein Main[OB1] die Zeilen 5...8, schneiden Sie diese aus und fügen Sie diese im neuen OB-Baustein DRUCK[OB123] ein. Sie können den Baustein Main umbenennen zu TEMP. Markieren Sie dazu diesen Baustein, öffnen Sie das Kontextmenü *Eigenschaften* und geben Sie z.B. den Namen TEMP ein.



2. Übersetzen Sie die Bausteine. Beim Übersetzen wird der Maschinencode für die CPU erzeugt. Markieren Sie dazu den Programmbausteinordner und wählen Sie im Menü *Bearbeiten Übersetzen*. Bei größeren Programmen ist es sinnvoll, den Programmcode beim Erstellen zwischendurch zu übersetzen, um **Syntax-Fehler** (Abweichungen von der Sprachvereinbarung) leichter zu finden und zu korrigieren.
3. **Testen** Sie die Funktion des Projekts mit PLCSIM. Sie können nach der Beschreibung im Kapitel 2.1.2 vorgehen: Markieren Sie PLC_1 in der Projektnavigation und laden Sie Ihr Projekt über *Menü Online-Simulation Start* in PLCSIM. Sie können z.B. mit einer Beobachtungstabelle arbeiten, um Variablenwerte zu steuern. Sie können dann im TIA-Portal über *Menü Online Laden in Gerät...* Ihr erweitertes Projekt in PLCSIM laden.



Anmerkung: Wenn Sie ein neues TIA-Projekt erstellen, entfernen Sie zunächst den Baustein Main[OB1], da dieser standardmäßig nur in den Sprachen FUP und KOP zu programmieren ist. Fügen Sie einen neuen Organisationsbaustein (OB) Program cycle in der Sprache SCL ein.

Im SCL-Editor des TIA-Portals ab Version 14 können Sie Ihren Programmcode mit Hilfe des Schlüsselworts REGION xxx ... END_REGION in Bereiche strukturieren. Dies verbessert ebenfalls die Übersichtlichkeit Ihres Programms. Im Anhang unter Lösung der Aufgabe 6.2 finden Sie diese Strukturmöglichkeit erläutert.

3.2 Gliederung mit einem Hauptprogramm und Unterprogramme

Wenn Sie eine CPU3xx verwenden, so steht Ihnen nur der OB1 als Programm zur Verfügung, das freilaufend zyklisch bearbeitet wird. Sie können hier gliedern, indem Sie Funktionen als «Container» für Ihre Unterprogramme hinzufügen, das Programm in diesen aufteilen und diese Unterprogramme im Hauptprogramm Main[OB1] aufrufen. Diese Art der Gliederung lernen Sie in Kapitel 5 kennen.

Merksatz

Gliedern Sie Ihr Programm, so können Sie dieses leichter erweitern, leichter Fehler finden und korrigieren, die Unterprogramme nacheinander testen und die Anlage schrittweise in Betrieb nehmen.

Stichpunkte

Nach der Bearbeitung des Kapitels sollten die folgenden Stichpunkte für Sie verständlich sein:

- ☐ mehrere Program-cycle-Bausteine
- ☐ Syntax-Fehler
- ☐ neuer Baustein
- ☐ PLCSIM
- ☐ übersetzen

4 Bibliotheksfähige, parametrisierbare Funktionen erstellen

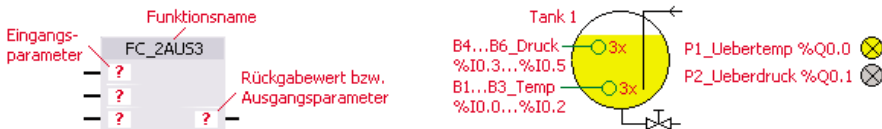
Sie lernen Programmentwurfsmethoden für Verknüpfungssteuerungen ohne Speicherverhalten kennen. Sie legen ein neues Projekt an, erstellen eine parametrisierbare Funktion in der Sprache SCL, rufen diese auf und testen das Programm. Sie unterscheiden globale und lokale Variable. Sie legen die erstellte Funktion in eine globale Bibliothek ab. Sie versuchen mit Hilfe der KI – über einen Chatbot – lauffähige Programmbausteine zu generieren.

Wird ein Programmteil öfter benötigt, so ist es vorteilhaft, diesen in einer parametrisierbaren Funktion (FC) zu schreiben. Sie können diese FC beliebig oft in Ihrem Programm, auch in anderen Bausteinen, erneut aufrufen. Dadurch wird weniger Speicher für das Anwenderprogramm belegt und eine Änderung des Programmcodes der Funktion ist leichter möglich.

Bibliotheksfähige Funktionen werden in FC-Bausteinen erstellt. In ihnen werden ausschließlich lokale Variable verwendet, um die FC-Bausteine mehrfach verwendbar zu halten. Lokale Variablen werden im Deklarationsteil des Bausteineditors deklariert.

Aufgabe 4.1: Funktion 2 aus 3

Erstellen Sie eine parametrisierbare Funktion «2 aus 3», die für die Temperatur- oder auch für die Drucküberwachung benutzt werden kann, und testen Sie diese.



4.1 Programmentwurf

Vorüberlegungen

Überlegen Sie, wie viele Ein- und Ausgangsparameter notwendig sind, wie Sie diese benennen wollen und wie diese logisch verknüpft werden.

Programmentwurf mit der Funktionstabelle

Das Programm einer Verknüpfungssteuerung ohne Speicherverhalten können Sie mit Hilfe einer Funktionstabelle und einer daraus erstellten **Funktionsgleichung** entwerfen. Das folgende Bild zeigt die Tabelle. Suchen Sie die Zeilen in der Tabelle, bei denen OUT den Wert TRUE (1) hat, und verknüpfen Sie die dazugehörigen INs mit AND. Ist der Wert eines IN FALSE (0), so schreiben Sie NOT vor dem IN. Sie finden auf **InfoClick** dazu eine Animation. Hat OUT in mehreren Zeilen den Wert TRUE, so werden die einzelnen Ausdrücke (Minterme) mit OR verknüpft.