



SIEMENS



Lern-/Lehrunterlagen

Siemens Automation Cooperates with Education
(SCE) | Ab Version V14 SP1

TIA Portal Modul 051-201
Hochsprachenprogrammierung
mit SCL und SIMATIC S7-1200

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

SIEMENS

Global Industry
Partner of
WorldSkills
International



Passende SCE Trainer Pakete zu dieser Lern-/Lehrunterlage

- **SIMATIC S7-1200 AC/DC/RELAIS 6er "TIA Portal"**
BestellNr.: 6ES7214-1BE30-4AB3
- **SIMATIC S7-1200 DC/DC/DC 6er "TIA Portal"**
BestellNr.: 6ES7214-1AE30-4AB3
- **Upgrade SIMATIC STEP 7 BASIC V14 SP1 (für S7-1200) 6er "TIA Portal"**
BestellNr.: 6ES7822-0AA04-4YE5

Bitte beachten Sie, dass diese Trainer Pakete ggf. durch Nachfolge-Pakete ersetzt werden. Eine Übersicht über die aktuell verfügbaren SCE Pakete finden Sie unter: [siemens.de/sce/tp](https://www.siemens.de/sce/tp)

Fortbildungen

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

Weitere Informationen rund um SCE

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

Verwendungshinweis

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

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

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

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

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

Wir danken der TU Dresden, besonders Prof. Dr.-Ing. Leon Urbas und der Fa. Michael Dziallas Engineering und allen weiteren Beteiligten für die Unterstützung bei der Erstellung dieser SCE Lern-/Lehrunterlage.

Inhaltsverzeichnis

1. Zielstellung.....	4
2. Voraussetzung.....	4
3. Benötigte Hardware und Software	5
4. Theorie	6
4.1 Zur Programmiersprache SCL.....	6
4.2 Zur Entwicklungsumgebung SCL.....	6
5. Aufgabenstellung	9
5.1 Beispielaufgabe Tankinhalt.....	9
5.2 Erweiterung der Beispielaufgabe.....	9
6. Planung	9
6.1 Globaler Datenbaustein „Daten_Tank“.....	9
6.2 Funktion „Berechnung_Tankinhalt“	10
6.3 Erweiterung der Funktion „Berechnung_Tankinhalt“	10
7. Strukturierte Schritt-für-Schritt-Anleitung.....	11
7.1 Deaktivieren eines vorhandenen Projekts.....	11
7.2 Speichern des Projektes unter einem neuen Namen.....	12
7.3 Anlegen des Datenbausteins „Daten_Tank“.....	12
7.4 Erstellen der Funktion „Berechne_Inhalt“	14
7.5 Schnittstelle der Funktion „Berechne_Inhalt“ festlegen.....	15
7.6 Programmierung der Funktion „Berechne_Inhalt“.....	16
7.7 Programmierung des Organisationsbausteins „Main [OB1]“	17
7.8 Programm übersetzen und laden.....	19
7.9 Organisationsbaustein beobachten und testen	20
7.10 Erweiterung der Funktion „Berechne_Inhalt“	22
7.11 Organisationsbaustein anpassen.....	27
7.12 Programm übersetzen, speichern und laden.....	28
7.13 Organisationsbaustein beobachten und testen	29
7.14 Funktion „Berechne_Inhalt“ beobachten und testen.....	31
7.15 Archivieren des Projektes.....	34
8. Checkliste	35
9. Übung.....	36
9.1 Aufgabenstellung – Übung	36
9.2 Planung	37
9.3 Checkliste – Übung	37
10. Weiterführende Information	38

Hochsprachenprogrammierung mit SCL und S7-1200

1 Zielstellung

In diesem Kapitel lernen Sie die grundlegenden Funktionen der Hochsprache SCL kennen. Des Weiteren werden Testfunktionen zur Beseitigung logischer Programmierfehler aufgezeigt.

Es können die unter Kapitel 3 aufgeführten SIMATIC S7-Steuerungen eingesetzt werden.

2 Voraussetzung

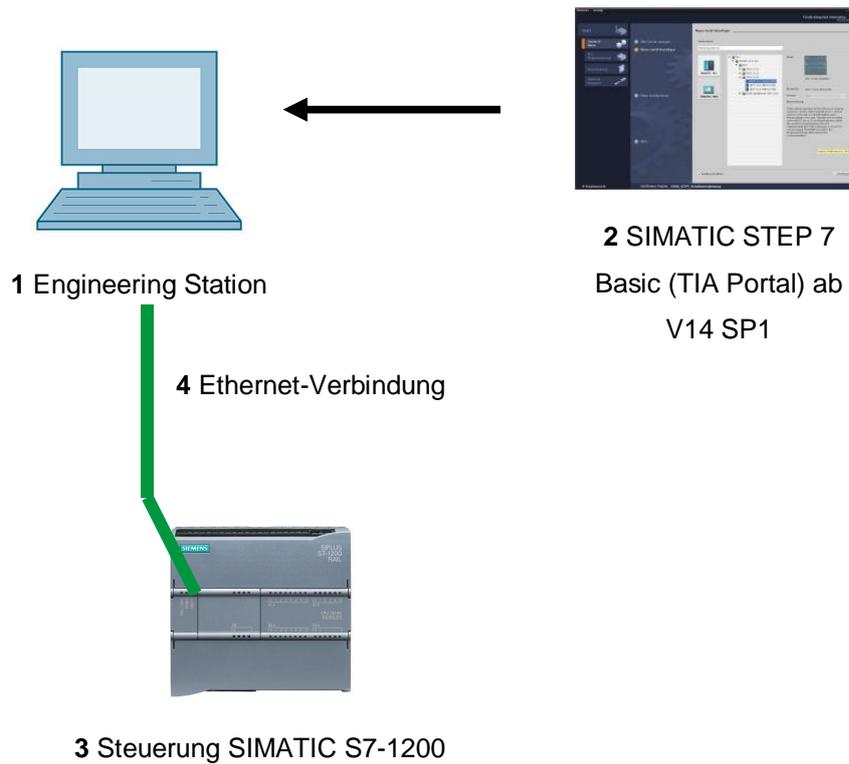
Dieses Kapitel baut auf der Hardwarekonfiguration einer SIMATIC S7-1200 auf. Es kann mit beliebigen Hardwarekonfigurationen, die digitale Eingangs- und Ausgangskarten besitzen, realisiert werden. Zur Durchführung dieses Kapitels können Sie z.B. auf das folgende Projekt zurückgreifen:

„SCE_DE_011_101_Hardwarekonfiguration_CPU1214C.....zap14“

Weiterhin sollten Grundlagenkenntnisse über Hochsprachenprogrammierung wie z.B. Pascal bekannt sein.

3 Benötigte Hardware und Software

- 1 Engineering Station: Voraussetzungen sind Hardware und Betriebssystem (weitere Informationen siehe Readme/Liesmich auf den TIA Portal Installations-DVDs)
- 2 Software SIMATIC STEP 7 Basic im TIA Portal – ab V14 SP1
- 3 Steuerung SIMATIC S7-1200, z.B. CPU 1214C DC/DC/DC – ab Firmware V4.2.1
- 4 Ethernet-Verbindung zwischen Engineering Station und Steuerung



4 Theorie

4.1 Zur Programmiersprache SCL

SCL (Structured Control Language) ist eine höhere Programmiersprache, die sich an Pascal orientiert und eine strukturierte Programmierung ermöglicht. Die Sprache entspricht der in der Norm DIN EN-61131-3 (IEC 61131-3) festgelegten Programmiersprache ST „Structured Text“. SCL enthält neben Hochsprachenelementen auch typische Elemente der SPS als Sprachelemente wie Eingänge, Ausgänge, Zeiten, Merker, Bausteinaufrufe usw. Sie unterstützt das Bausteinkonzept von STEP 7 und ermöglicht daher neben Kontaktplan (KOP) und Funktionsplan (FUP) die normkonforme Programmierung von Bausteinen. D. h. SCL ergänzt und erweitert die Programmiersoftware STEP 7 mit ihren Programmiersprachen KOP und FUP.

Sie müssen nicht jede Funktion selbst erstellen, sondern können auf vorgefertigte Bausteine wie Systemfunktionen oder Systemfunktionsbausteine zurückgreifen, die im Betriebssystem der Zentralbaugruppe vorhanden sind.

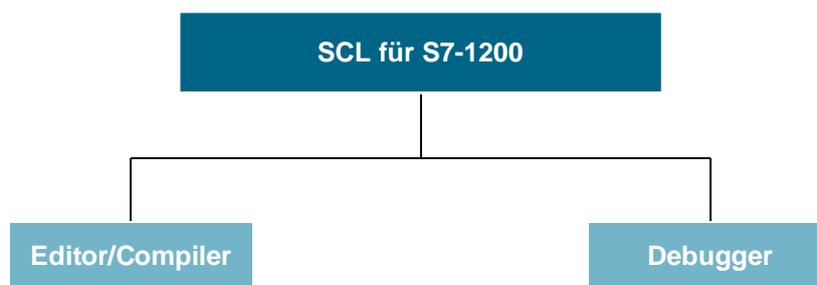
Sie können Bausteine, die mit SCL programmiert sind, mit KOP- und FUP-Bausteinen mischen. Das bedeutet, dass ein mit SCL programmierter Baustein einen anderen Baustein, der in KOP oder FUP programmiert ist, aufrufen kann. Entsprechend können SCL Bausteine auch in KOP- und FUP-Programmen aufgerufen werden.

Es können ebenfalls SCL-Netzwerke in KOP- und FUP-Bausteinen eingefügt werden.

Die Testfunktionen von SCL ermöglichen die Suche nach logischen Programmierfehlern in einer fehlerfreien Übersetzung.

4.2 Zur Entwicklungsumgebung SCL

Zur Verwendung und zum Einsatz von SCL gibt es eine Entwicklungsumgebung, die sowohl auf spezifische Eigenschaften von SCL, als auch auf STEP 7 abgestimmt ist. Diese Entwicklungsumgebung besteht aus einem Editor/Compiler und einem Debugger.



Editor/Compiler

Der SCL-Editor ist ein Texteditor, mit dem beliebige Texte bearbeitet werden können. Seine zentrale Aufgabe ist das Erzeugen und Bearbeiten von Bausteinen für STEP 7-Programme. Während der Eingabe erfolgt eine grundlegende Syntaxprüfung, welche das fehlerfreie Programmieren vereinfacht. Syntaxfehler werden in unterschiedlichen Farben dargestellt.

Der Editor bietet folgende Möglichkeiten:

- Programmierung eines S7-Bausteines in der Sprache SCL
- Komfortables Einfügen von Sprachelementen und Bausteinaufrufen mittels Drag&Drop
- Direkte Syntaxprüfung während der Programmierung
- Einstellung des Editors nach Ihren Anforderungen, z.B. durch syntaxgerechtes Einfärben der verschiedenen Sprachelemente
- Überprüfung des fertiggestellten Bausteines mittels übersetzen
- Anzeigen aller Fehler und Warnungen, die beim Übersetzen auftreten
- Lokalisieren der fehlerhaften Stellen im Baustein, optional mit Fehlerbeschreibung und Angaben zur Fehlerbeseitigung

Debugger

Der SCL-Debugger kann ein Programm in seinem Ablauf im Automatisierungssystem (AS) kontrollieren und somit mögliche logische Fehler finden.

SCL bietet dazu zwei verschiedene Testmodi an:

- Kontinuierliches Beobachten
- Schrittweises Beobachten

Mit dem „Kontinuierlichen Beobachten“ können Sie eine Gruppe von Anweisungen innerhalb eines Bausteins testen. Während des Testlaufs werden die Werte der Variablen und Parameter in chronologischer Abfolge angezeigt und – sofern möglich – zyklisch aktualisiert.

Beim „Schrittweisen Beobachten“ wird der logische Programmablauf nachvollzogen. Sie können den Programm-Algorithmus Anweisung für Anweisung ausführen und in einem Ergebnisfenster beobachten, wie sich die dabei bearbeiteten Variableninhalte verändern.

Ob das „Schrittweise Beobachten“ möglich ist, hängt von der eingesetzten CPU ab. Diese muss den Einsatz von Haltepunkten unterstützen. Die in diesem Dokument eingesetzte CPU unterstützt keine Haltepunkte.

5 Aufgabenstellung

5.1 Beispielaufgabe Tankinhalt

Im ersten Teil soll die Berechnung eines Tankinhaltes programmiert werden.

5.2 Erweiterung der Beispielaufgabe

Im zweiten Teil soll die Aufgabe erweitert und eine Fehlerauswertung programmiert werden.

6 Planung

Der Tank hat die Form eines stehenden Zylinders. Der Füllstand des Inhaltes wird mit einem Analogsensor gemessen. Für den ersten Test soll der Wert des Füllstandes schon normiert, in der Einheit „Meter“, vorliegen.

Globale Parameter wie z.B. der Durchmesser und die Höhe des Tanks sollen in einem globalen Datenbaustein „Daten_Tank“ strukturiert abgelegt werden.

Das Programm zur Berechnung des Inhaltes soll in einer Funktion „Berechnung_Tankinhalt“ geschrieben werden und die Parameter die Einheit „Meter“ bzw. „Liter“ verwenden.

6.1 Globaler Datenbaustein „Daten_Tank“

Die globalen Parameter werden in einem globalen Datenbaustein in mehreren Strukturen abgelegt.

Name	Datentyp	Startwert	Kommentar
Abmessungen	STRUCT		
Hoehe	REAL	12.0	in Meter
Durchmesser	REAL	3.5	in Meter
Messwerte	STRUCT		
Fuellstand_per	INT	0	Wert zwischen 0...27648
Fuellstand_skal	REAL	0.0	Wert zwischen 0...12.0
Inhalt	REAL	0.0	Inhalt des Tanks in Liter
Fehlerflags	STRUCT		
berechne_inhalt	BOOL		im Fehlerfall = TRUE
skalieren	BOOL		im Fehlerfall = TRUE

Tabelle 1: Parameter im Datenbaustein "Daten_Tank"

6.2 Funktion „Berechnung_Tankinhalt“

Dieser Baustein berechnet den Inhalt des Tanks in Litern.

Im ersten Schritt soll keine Überprüfung auf Sinnhaftigkeit der übergebenen Parameter erfolgen.

Für diesen Schritt sind folgende Parameter erforderlich:

Input	Datentyp	Kommentar
Durchmesser	REAL	Durchmesser des Zylindertanks in Meter
Fuellstand	REAL	Füllstand des Tankinhaltes in Meter
Output		
Inhalt	REAL	Inhalt des Zylindertanks in Liter

Tabelle 2: Parameter für FC "Berechnung_Tankinhalt" im ersten Schritt

Zur Lösung der Aufgabe wird die Formel zur Volumenberechnung eines stehenden Zylinders angewendet. Der Umrechnungsfaktor 1000 wird verwendet, um das Ergebnis in Litern zu berechnen.

$$V = \frac{d^2}{4} \cdot \rho \cdot h \Rightarrow \quad \#Inhalt = \frac{\#Durchmesser^2}{4} \cdot 3.14159 \cdot \#Fuellstand \cdot 1000$$

6.3 Erweiterung der Funktion „Berechnung_Tankinhalt“

Der zweite Schritt prüft, ob der Durchmesser größer als Null ist. Des Weiteren soll getestet werden, ob der Füllstand größer oder gleich Null und kleiner oder gleich der Höhe des Tanks ist.

Im Fehlerfall wird der neue Parameter „er“ auf TRUE gesetzt und der Parameter „Inhalt“ erhält den Wert -1.

Erweitern Sie dazu die Schnittstelle um den Parameter „er“ und „Hoehe“.

Input	Datentyp	Kommentar
Hoehe	REAL	Hoehe des Zylindertanks in Meter
Durchmesser	REAL	Durchmesser des Zylindertanks in Meter
Fuellstand	REAL	Füllstand des Tankinhaltes in Meter
Output		
er	BOOL	Fehlerflag; bei Fehler = TRUE
Inhalt	REAL	Inhalt des Zylindertanks in Liter

Tabelle 3: Parameter für FC "Berechnung_Tankinhalt" im zweiten Schritt

7 Strukturierte Schritt-für-Schritt-Anleitung

Im Folgenden finden Sie eine Anleitung zur Umsetzung der Planung. Sollten Sie gut zurechtkommen, so reichen Ihnen die nummerierten Schritte zur Bearbeitung aus. Ansonsten folgen Sie einfach den folgenden detaillierten Schritten der Anleitung.

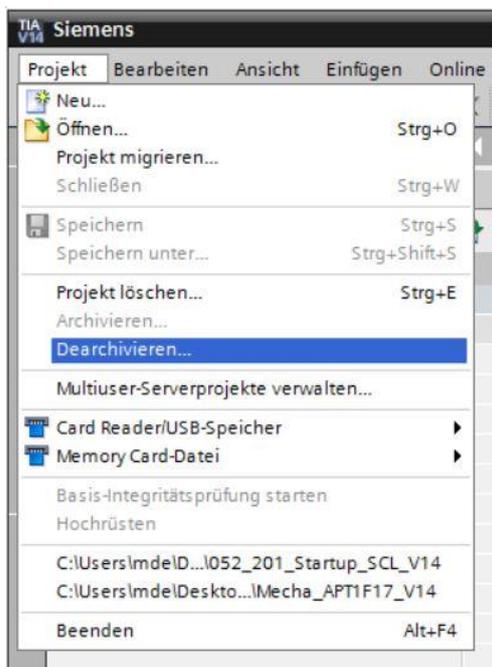
7.1 Dearchivieren eines vorhandenen Projekts

- ① Bevor Sie mit der Programmierung beginnen können, benötigen Sie ein Projekt mit einer Hardwarekonfiguration.

(z.B. SCE_DE_011-101_Hardwarekonfiguration_CPU1214C_....zap14)

Zum Dearchivieren eines vorhandenen Projekts müssen Sie aus der Projektansicht heraus unter ① Projekt ① Dearchivieren das jeweilige Archiv aussuchen. Bestätigen Sie Ihre Auswahl anschließend mit „öffnen“.

(① Projekt ① Dearchivieren ① Auswahl eines .zap-Archivs ① öffnen)

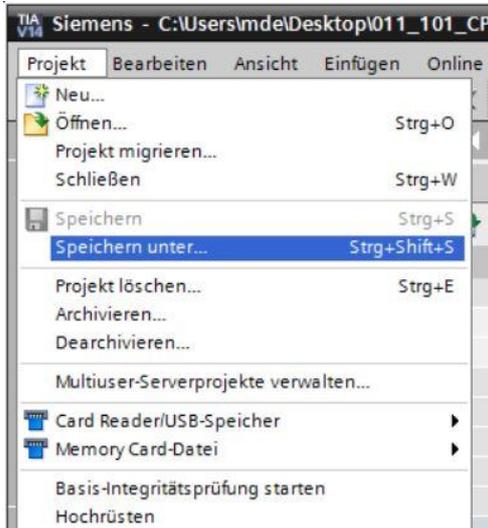


- ② Als nächstes kann das Zielverzeichnis ausgewählt werden, in welchem das dearchivierte Projekt gespeichert werden soll. Bestätigen Sie Ihre Auswahl mit „OK“.

(② Projekt ② Speichern unter ② OK)

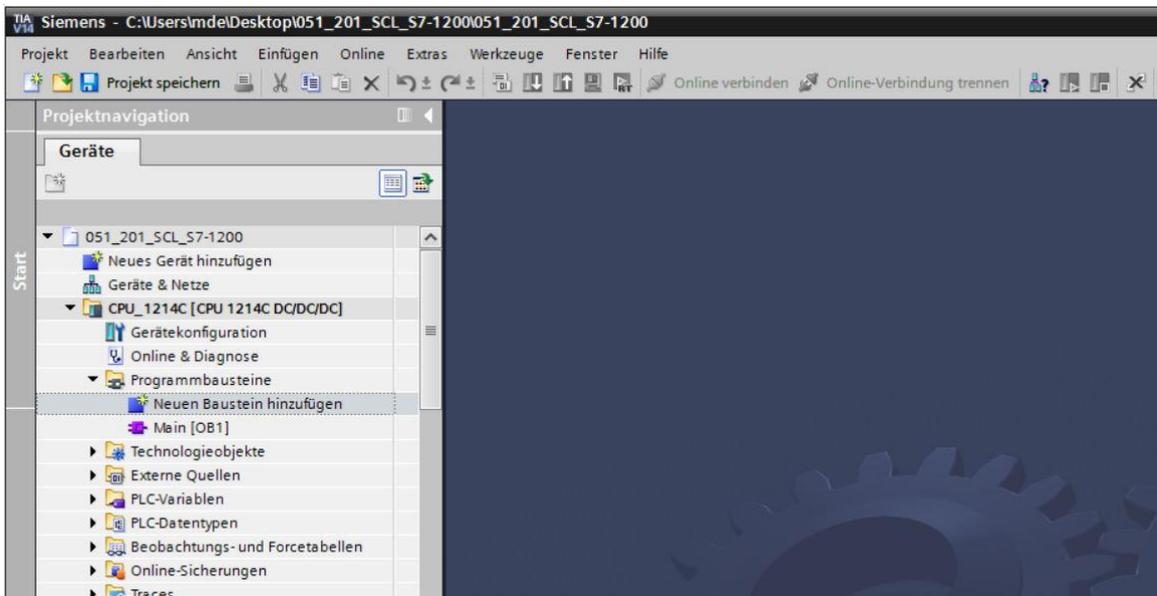
7.2 Speichern des Projektes unter einem neuen Namen

- Ⓡ Das geöffnete Projekt speichern Sie unter dem Namen 051-201_SCL_S7-1200.
 (Ⓡ Projekt Ⓡ Speichern unter ... Ⓡ 051-201_SCL_S7-1200 Ⓡ Speichern)

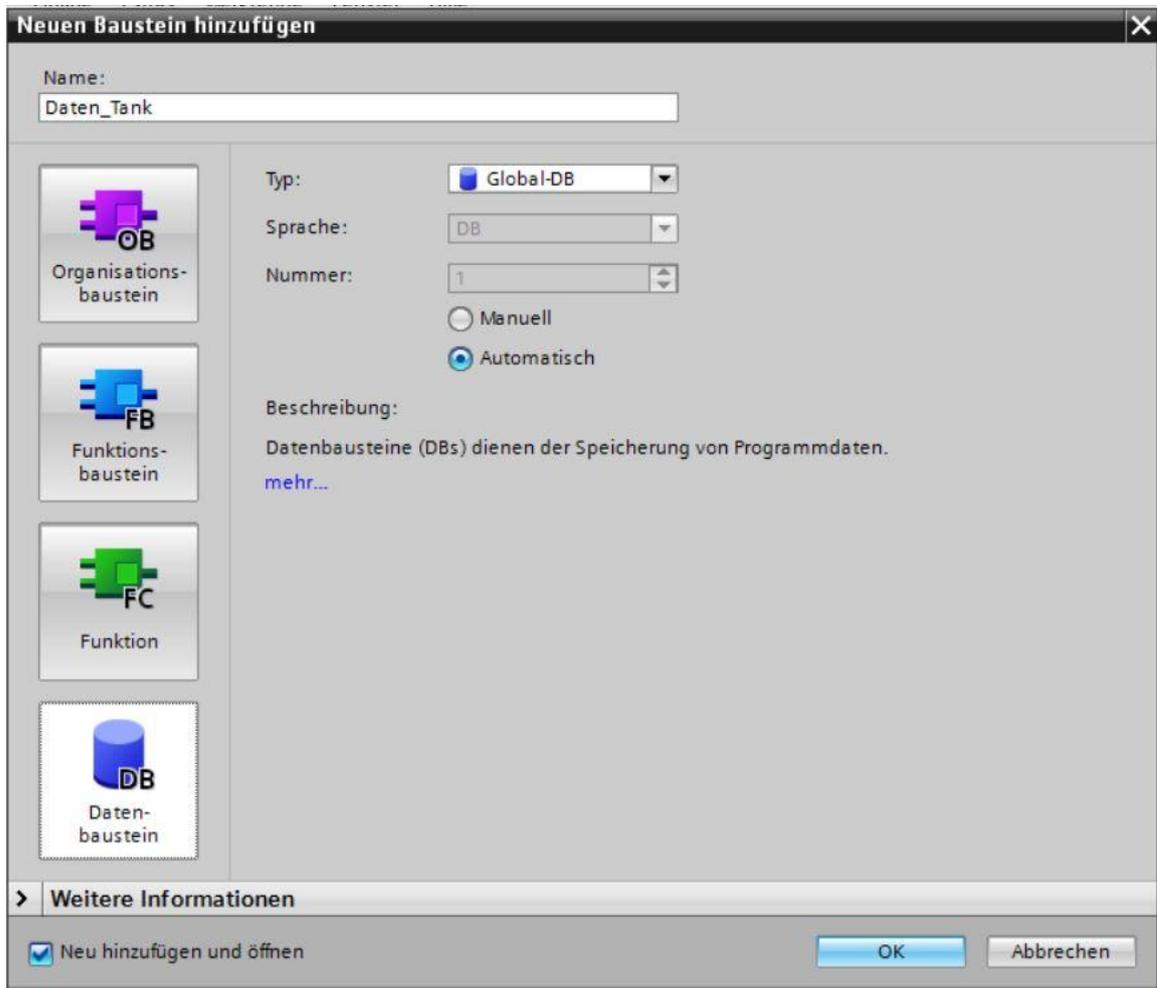


7.3 Anlegen des Datenbausteins „Daten_Tank“

- Ⓡ Navigieren Sie in der Projektansicht zu den Ⓡ Programmbausteinen und erstellen Sie einen neuen Baustein, indem Sie doppelklicken auf Ⓡ Neuen Baustein hinzufügen.



- ③ Wählen Sie nun einen Datenbaustein an und geben den Namen ein.



- Ⓜ Geben Sie anschließend die unten angegebenen Namen der Variablen mit Datentyp, Startwert und Kommentar ein.

	Name	Datentyp	Startwert	Remanenz	Erreichbar a..	Schrei...	Sichtbar i...	Einstellwert	Kommentar
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Abmessungen	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	Hoehe	Real	12.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	in Meter
4	Durchmesser	Real	3.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	in Meter
5	Messwerte	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	Fuellstand_per	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Wert zwischen 0..27648
7	Fuellstand_skal	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Wert zwischen 0..12.0
8	Inhalt	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	in Liter
9	Fehlerflags	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	berechne_inhalt	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	im Fehlerfall = TRUE
11	skalieren	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	<Hinzufügen>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

7.4 Erstellen der Funktion „Berechne_Inhalt“

- Ⓜ Jetzt fügen Sie eine Funktion hinzu, geben den Namen ein und wählen die Sprache aus.



(Ⓜ Neuen Baustein hinzufügen Ⓜ Funktion Ⓜ „Berechne_Inhalt“ Ⓜ SCL Ⓜ OK)

Name: Berechne_Inhalt

Sprache: SCL

Nummer: 1

Manuell
 Automatisch

Beschreibung:
Funktionen sind Codebausteine ohne Gedächtnis.

[mehr...](#)

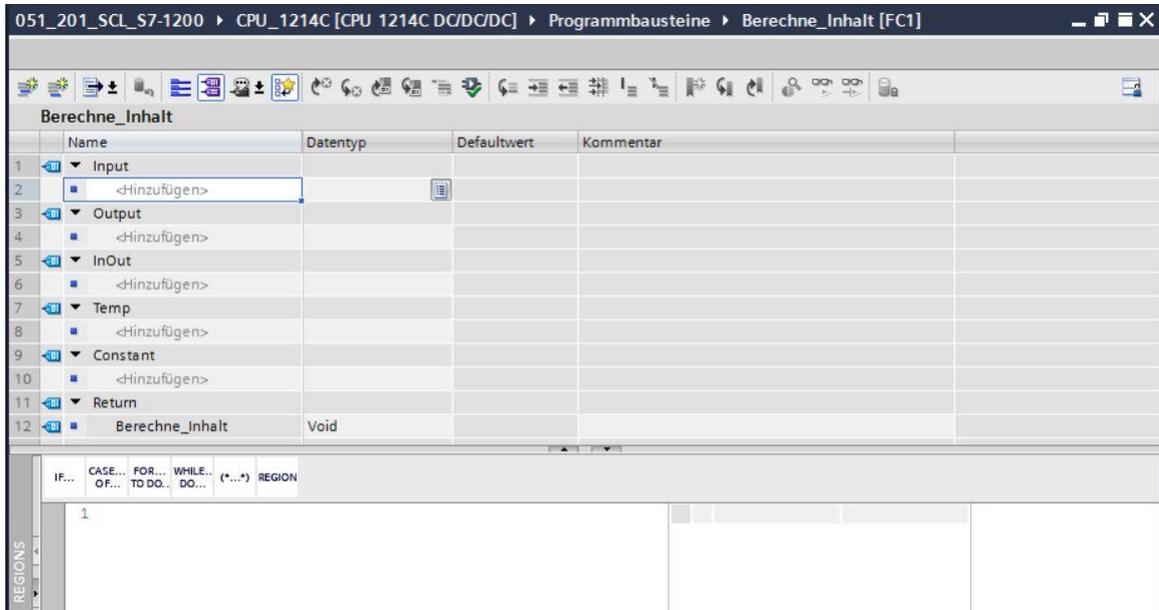
Weitere Informationen

Neu hinzufügen und öffnen

OK Abbrechen

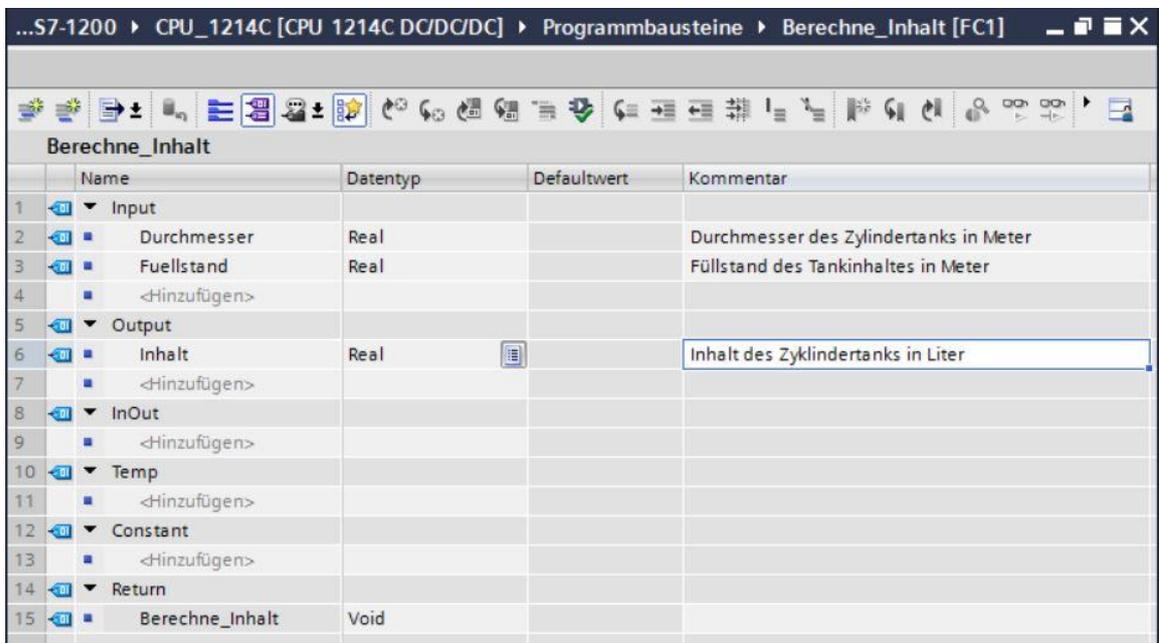
7.5 Schnittstelle der Funktion „Berechne_Inhalt“ festlegen

- ® Im oberen Abschnitt Ihrer Programmiersicht finden Sie die Schnittstellenbeschreibung Ihrer Funktion.



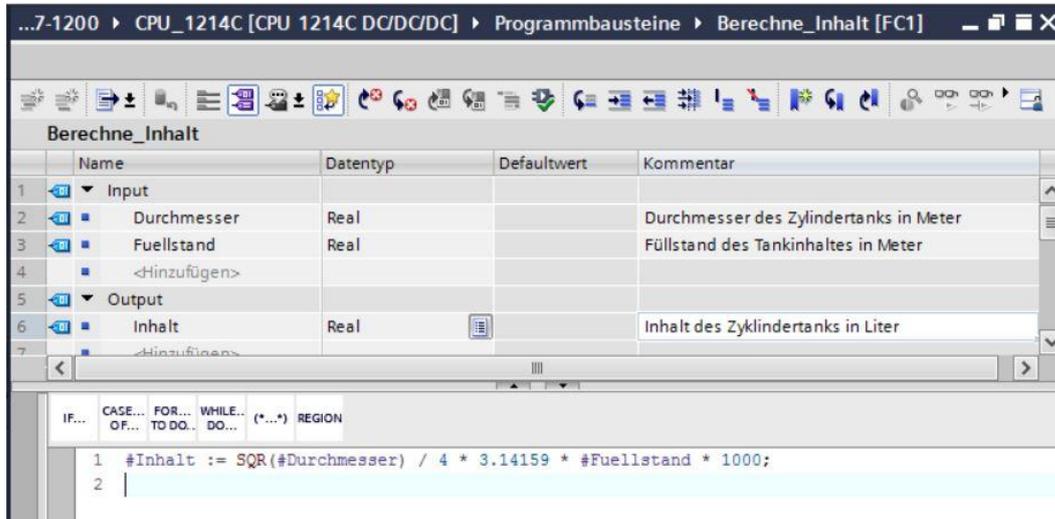
- ® Legen Sie die folgenden Input- und Outputparameter an.

(® Name ® Datentyp ® Kommentar)



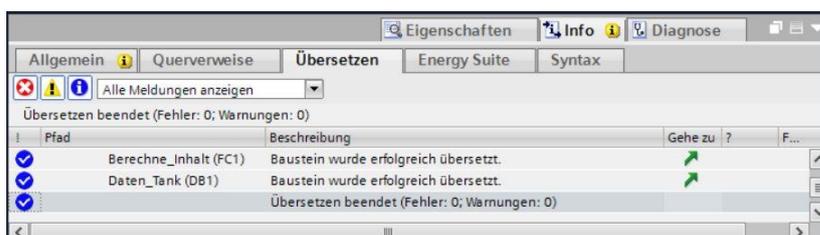
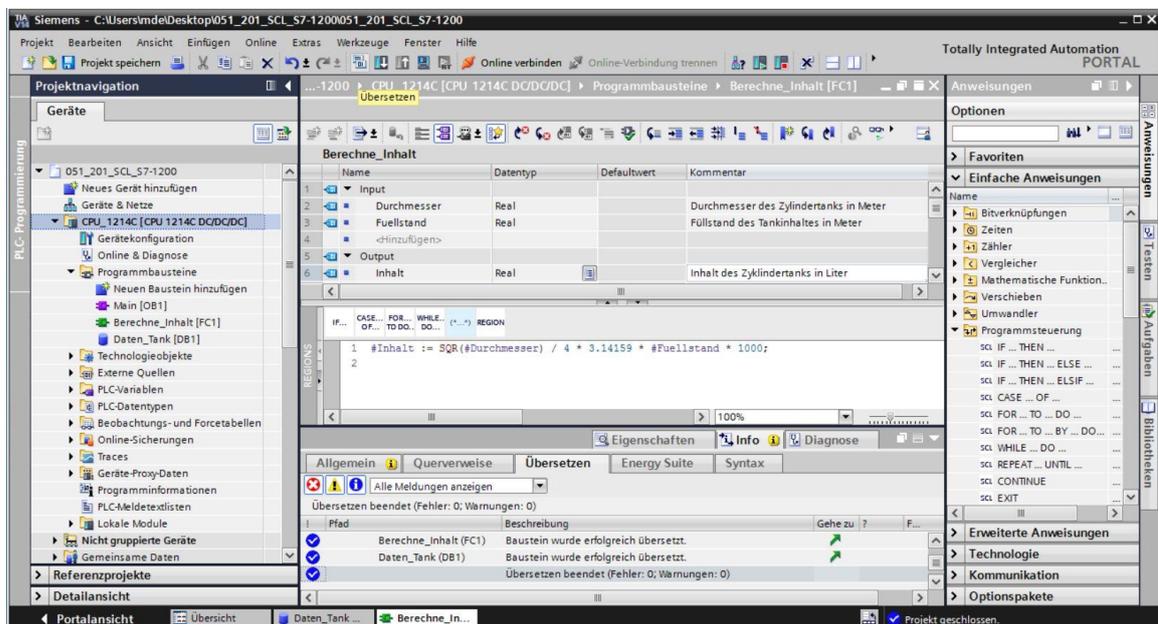
7.6 Programmierung der Funktion „Berechne_Inhalt“

Ⓜ Geben Sie unten stehendes Programm ein. (Ⓜ Programm eingeben)



Ⓜ Übersetzen Sie nun Ihr Programm und überprüfen es auf syntaktische Fehler. Diese werden im Inspektorfenster unterhalb der Programmierung angezeigt. Beheben Sie gegebenenfalls die Fehler und übersetzen Sie anschließend erneut.

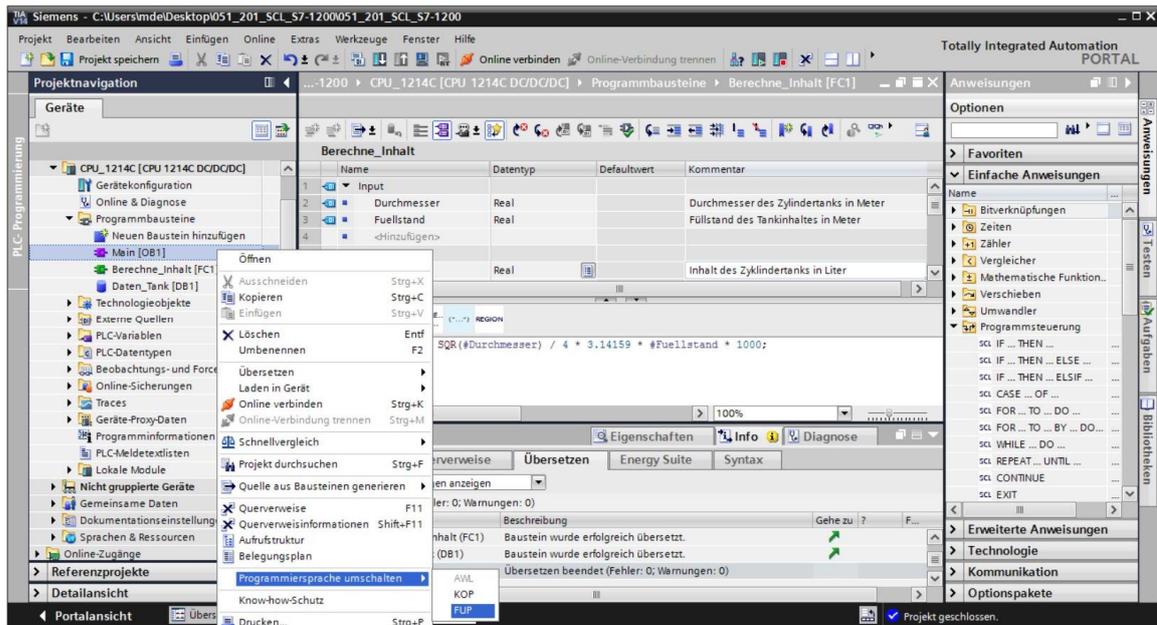
Speichern Sie danach Ihr Programm. (Ⓜ Fehler beheben Ⓜ **Projekt speichern**)



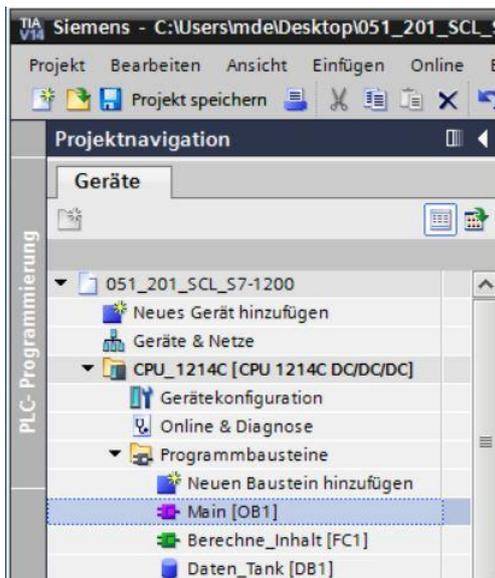
7.7 Programmierung des Organisationsbausteins „Main [OB1]“

Ⓡ Vor der Programmierung des Organisationsbausteins „Main [OB1]“ stellen Sie die Programmiersprache auf FUP um. Klicken Sie hierzu vorher mit der linken Maustaste im Ordner „Programmbausteine“ auf „Main [OB1]“.

(Ⓡ CPU_1214C[CPU 1214C DC/DC/DC] Ⓡ Programmbausteine Ⓡ Main [OB1] Ⓡ Programmiersprache umschalten Ⓡ FUP)

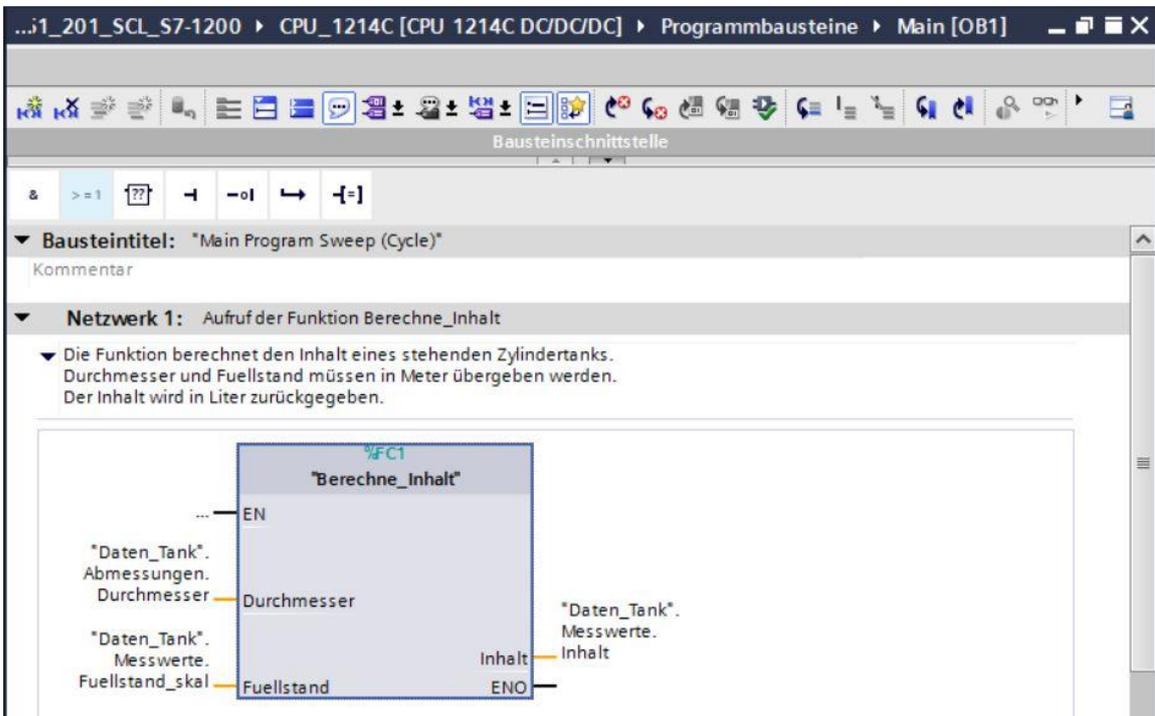
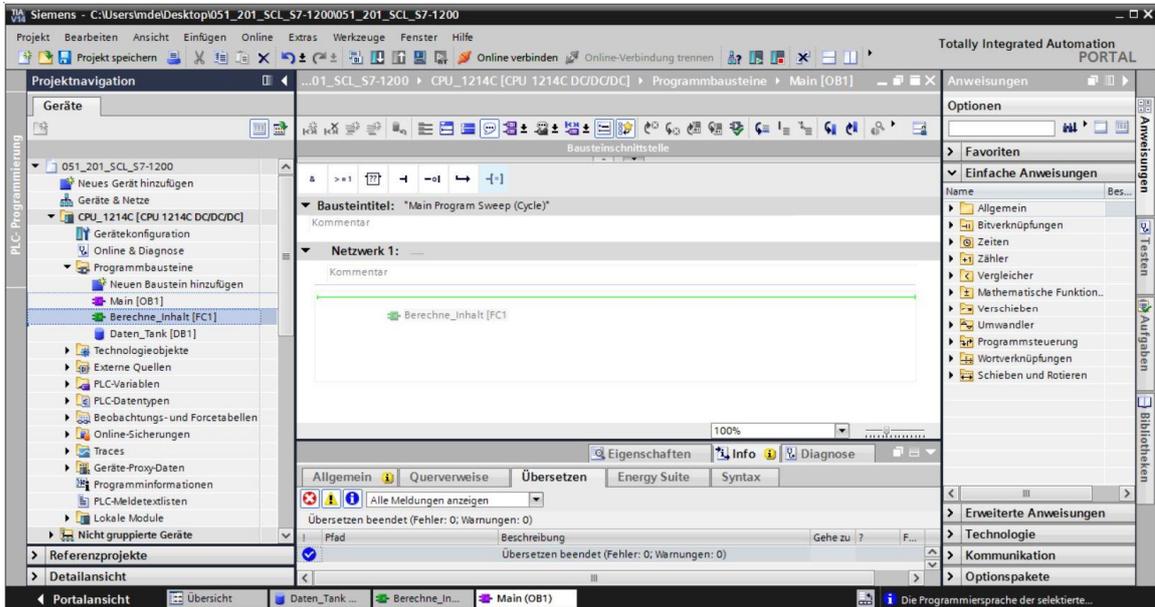


Ⓡ Öffnen Sie nun den Organisationsbaustein „Main [OB1]“ mit einem Doppelklick.



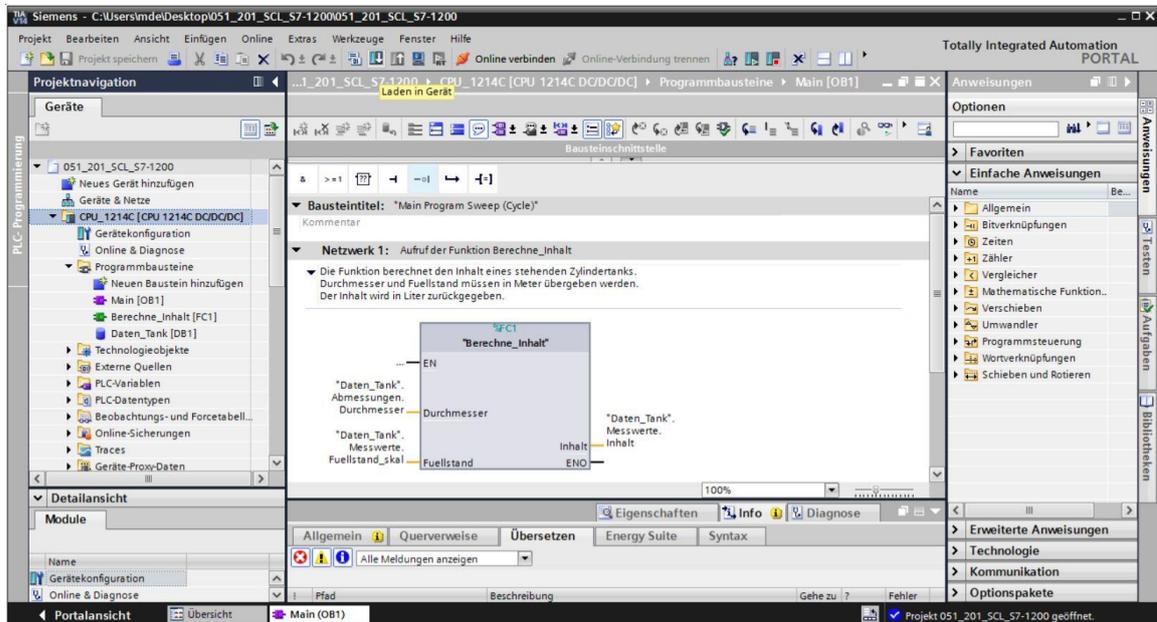
Ⓜ Rufen Sie die Funktion „Berechne_Inhalt“ im ersten Netzwerk auf. Vergeben Sie Netzwerktitel, Kommentar und beschriften Sie die Parameter.

(Ⓜ Aufruf „Berechne_Inhalt“ Ⓜ Netzwerktitel vergeben Ⓜ Netzwerkkommentar schreiben Ⓜ Parameter beschriften)

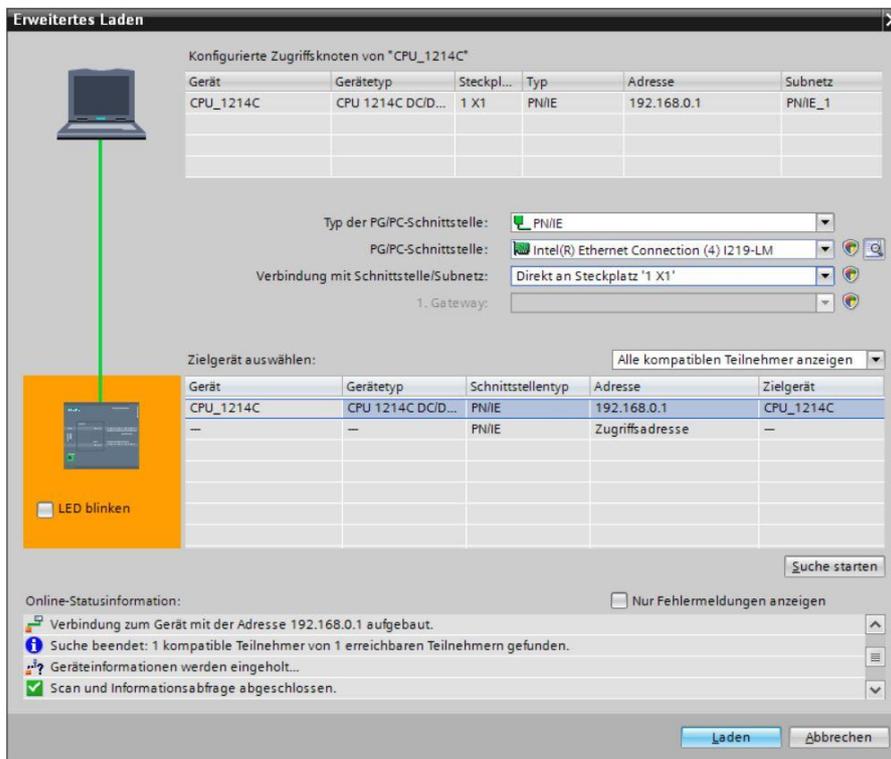


7.8 Programm übersetzen und laden

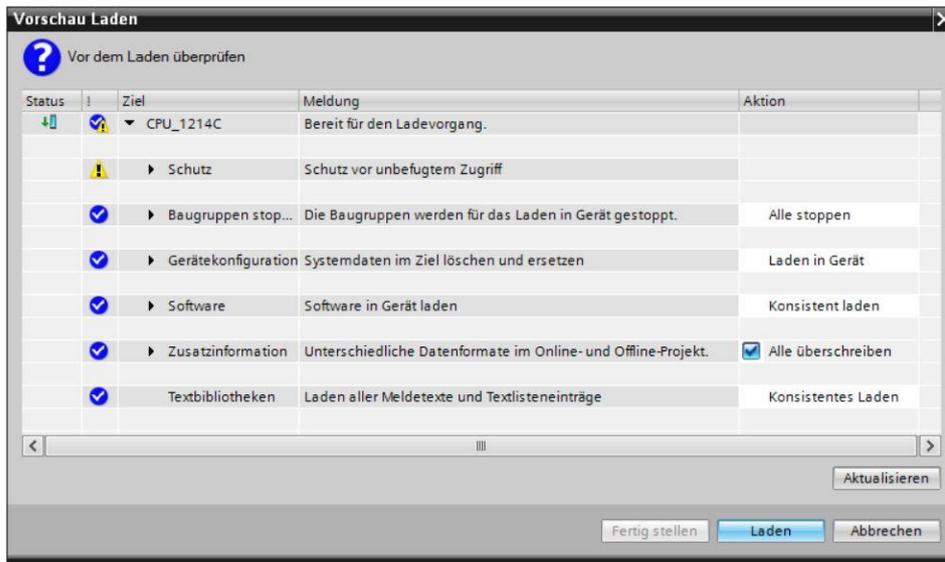
- Ⓜ Klicken Sie auf den Ordner „Programmbausteine“ und übersetzen Sie das gesamte Programm. Nach erfolgreichem Übersetzen speichern Sie Ihr Projekt und laden es in die Steuerung. (Ⓜ  Ⓜ  Ⓜ )



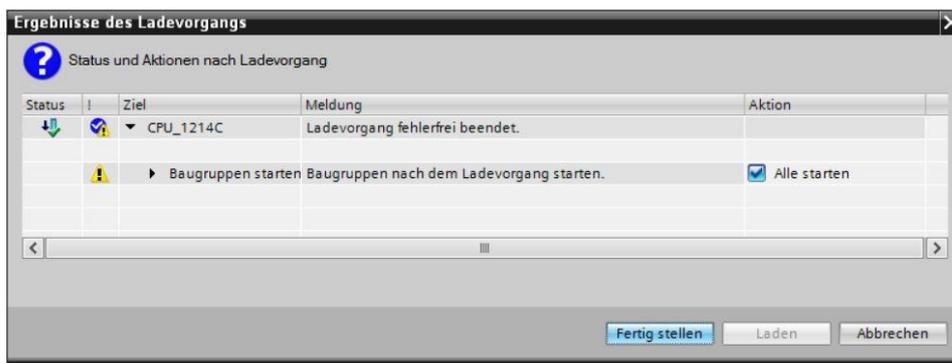
- Ⓜ PG/PC-Schnittstelle auswählen Ⓜ Subnetz auswählen Ⓜ Suche starten Ⓜ Laden



Ⓜ Eventuell Auswahl treffen Ⓜ Laden

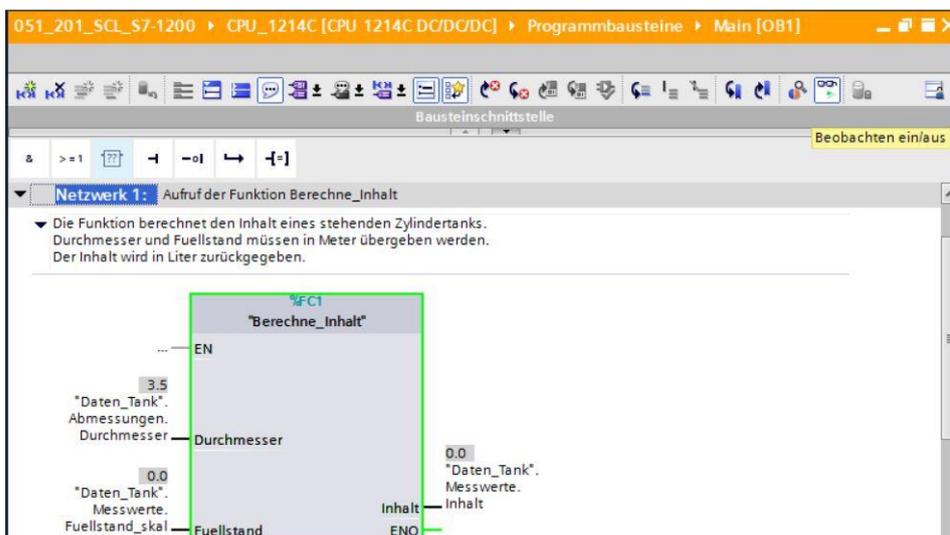


Ⓜ Fertig stellen



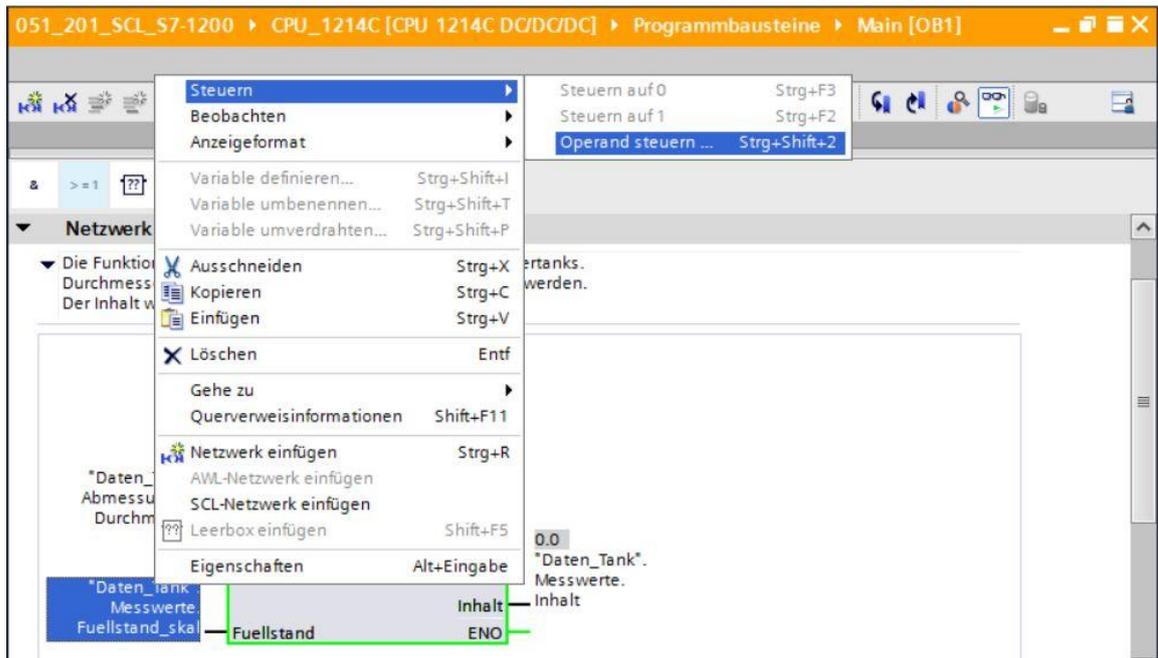
7.9 Organisationsbaustein beobachten und testen

Ⓜ Klicken Sie im geöffneten OB1 auf das Symbol , um den Baustein zu beobachten.

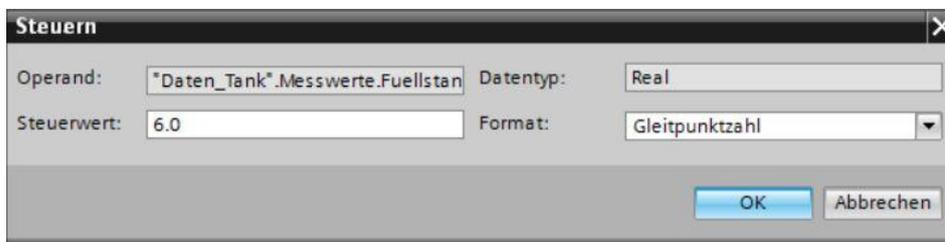


® Testen Sie Ihr Programm, indem Sie einen Wert in die Variable „Fuellstand_skal“ im Datenbaustein schreiben.

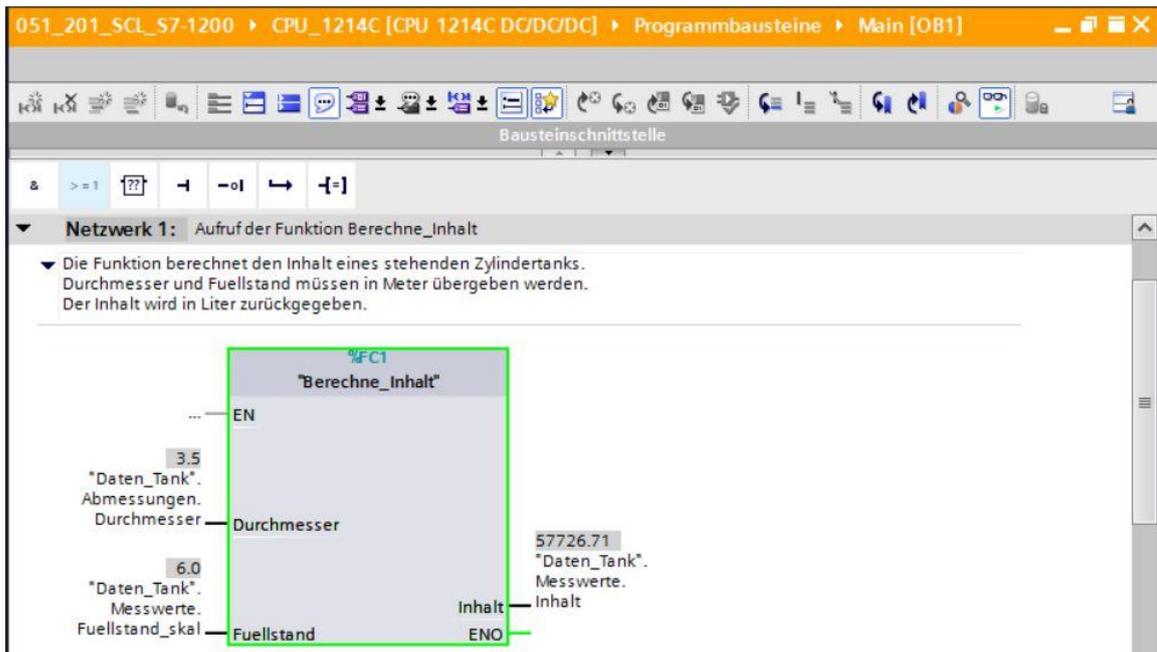
(® Rechtsklick auf „Fuellstand_skal“ ® Menü „Steuern“ ® Operand steuern)



® Wert 6.0 eintragen ® OK



- ® Überprüfen Sie das Ergebnis auf Richtigkeit.



7.10 Erweiterung der Funktion „Berechne_Inhalt“

- ® Öffnen Sie die Funktion „Berechne_Inhalt“ und fügen Sie durch Rechtsklick, auf die Zeile in der Schnittstelle, eine weitere Zeile bei den Outputparametern ein.

(® „Berechne_Inhalt“ öffnen ® Rechtsklick auf Zeile 5 ® Zeile einfügen)

The screenshot shows the 'Berechne_Inhalt' function block with a context menu open over the 'Inhalt' output parameter. The menu options are:

- Zeile einfügen
- Zeile hinzufügen
- Ausschneiden (Strg+X)
- Kopieren (Strg+C)
- Einfügen (Strg+V)
- Löschen (Entf)
- Umbenennen (F2)
- Schnittstelle aktualisieren
- Zur nächsten Verwendungsstelle gehen (Strg+Shift+G)
- Gehe zu Definition (Strg+Shift+D)
- Querverweise (F11)
- Querverweisinformationen (Shift+F11)

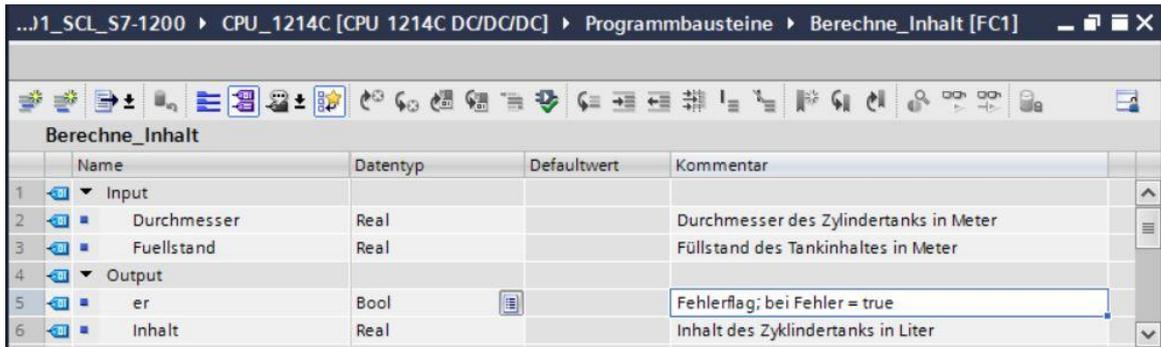
The function block table is as follows:

Name	Datentyp	Defaultwert	Kommentar
Input			
Durchmesser	Real		Durchmesser des Zylindertanks in Meter
Fuellstand	Real		Füllstand des Tankinhaltes in Meter
Output			
Inhalt	Real		Inhalt des Zylindertanks in Liter

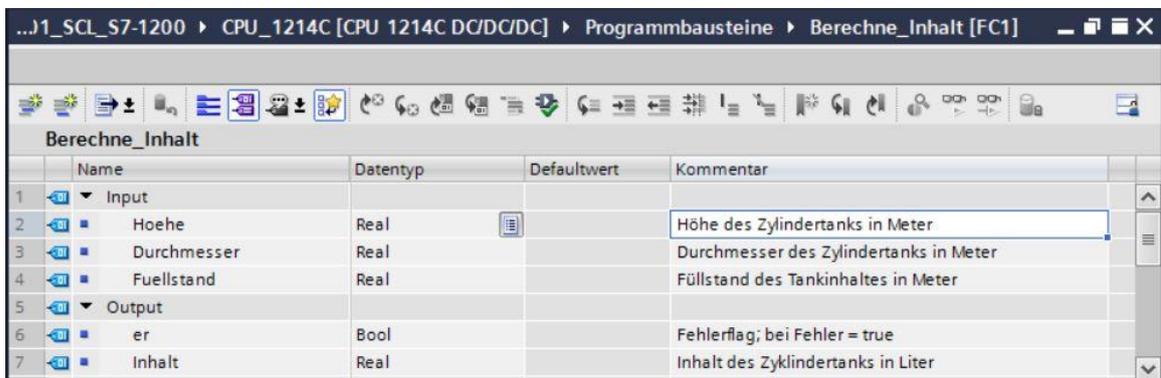
The code editor shows the following code:

```
9 * #Fuellstand * 1000;
```

Ⓜ Tragen Sie den Parameter „er“ mit Datentyp BOOL und Kommentar ein.



Ⓜ Nachfolgend fügen Sie auf die gleiche Weise die Input- Variable „Hoehe“ mit Datentyp Real und Kommentar ein.



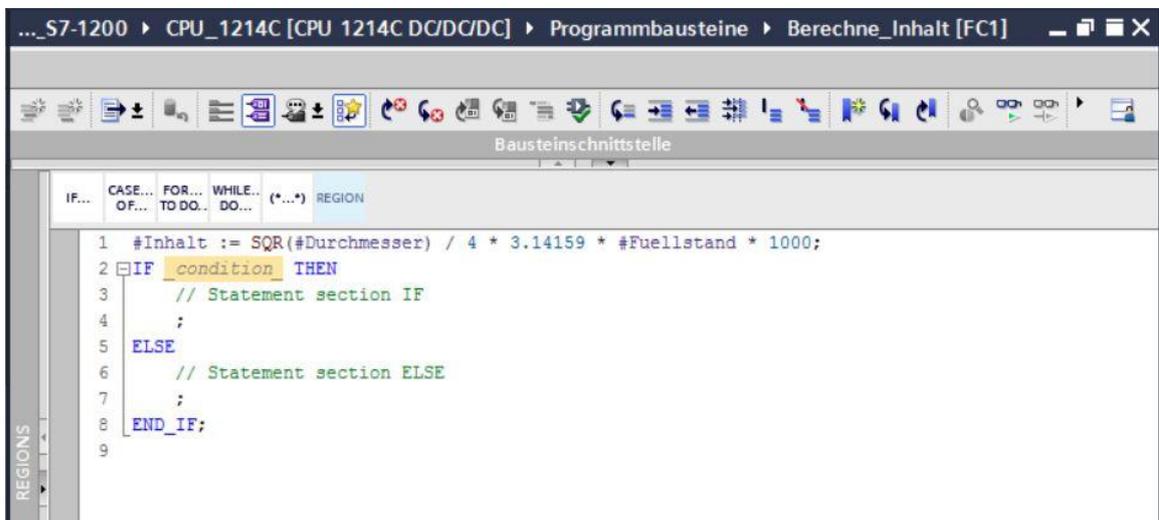
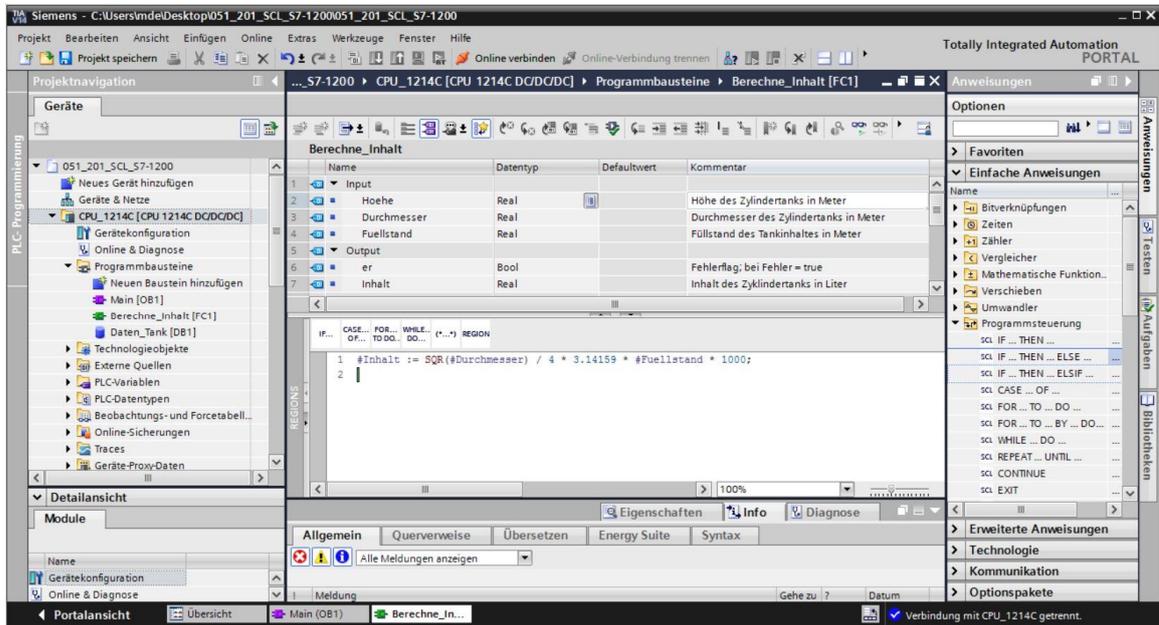
Ⓜ Navigieren Sie danach zur Kontrollstruktur „IF...THEN...ELSE“ aus dem Ordner „Programmsteuerung“ der einfachen Anweisungen.

(Ⓜ Anweisungen Ⓜ Einfache Anweisungen Ⓜ Programmsteuerung Ⓜ „IF...THEN...ELSE“)



Ⓜ Ziehen Sie anschließend die Kontrollstruktur „IF...THEN...ELSE“ per Drag&Drop in die zweite Zeile des Programms.

(Ⓜ „IF...THEN...ELSE“ Ⓜ Drag&Drop)



- Ⓜ Markieren Sie die mathematische Formel und ziehen Sie diese per Drag & Drop auf das Semikolon vor dem ELSE. (Ⓜ Markieren Ⓜ Drag & Drop)

```

1 #Inhalt := SQR(#Durchmesser) / 4 * 3.14159 * #Fuellstand * 1000;
2 IF condition THEN
3   // Statement section IF
4   ;
5 ELSE
6   // Statement section ELSE
7   ;
8 END_IF;
9

```

```

1
2 IF condition THEN
3   // Statement section IF
4   #Inhalt := SQR(#Durchmesser) / 4 * 3.14159 * #Fuellstand * 1000;
5 ELSE
6   // Statement section ELSE
7   ;
8 END_IF;
9

```

- Ⓜ Vervollständigen Sie die Funktion und überprüfen Ihr Programm durch Übersetzen.
(Ⓜ Programm ergänzen Ⓜ )

```

1 IF #Durchmesser > 0 AND #Fuellstand >= 0 AND #Fuellstand <= #Hoehe THEN
2     // Statement section IF
3     #er := FALSE;
4     #Inhalt := SQR(#Durchmesser) / 4 * 3.14159 * #Fuellstand * 1000;
5 ELSE
6     // Statement section ELSE
7     #er := TRUE;
8     #Inhalt := -1;
9 END_IF;
10

```

ⓐ Kommentare können mit „(**)“ als Blockkommentar und mit „//“ als Zeilenkommentar eingefügt werden. Jetzt können Sie Ihr Programm durch Kommentare ergänzen.

(ⓐ Blockkommentar ab Zeile 1 einfügen ⓐ Zeilenkommentar in Zeile 12/16 einfügen)

Name	Datentyp	Defaultwert	Kommentar
Input			
Hoehe	Real		Höhe des Zylindertanks in Meter
Durchmesser	Real		Durchmesser des Zylindertanks in Meter
Fuellstand	Real		Füllstand des Tankinhaltes in Meter
Output			
er	Bool		Fehlerflag; bei Fehler = true
Inhalt	Real		Inhalt des Zylindertanks in Liter

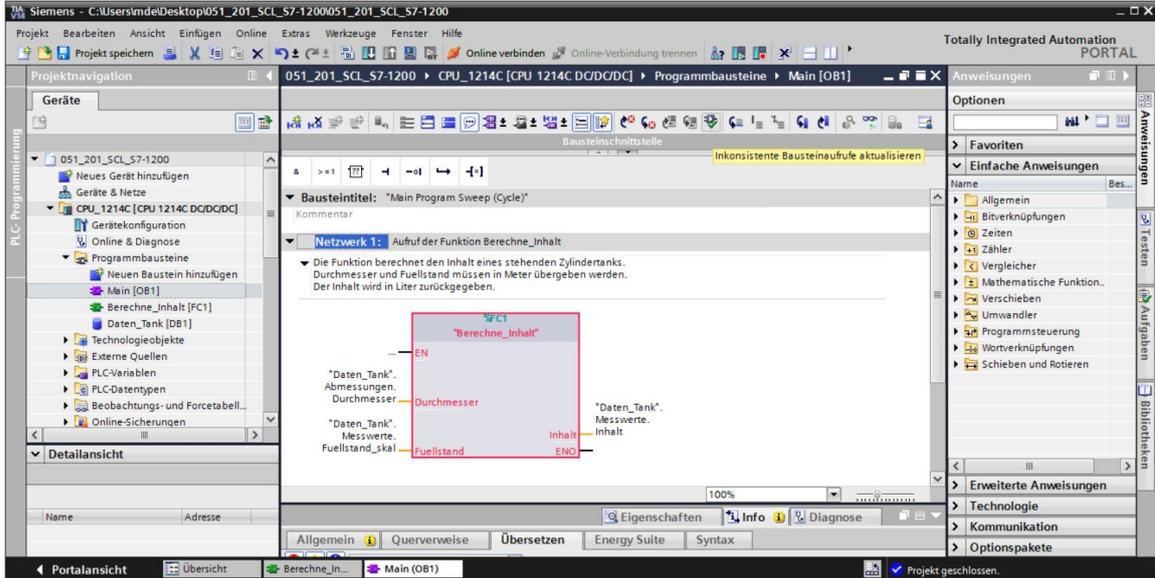
```

1 (**
2 Die Funktion berechnet den Inhalt eines stehenden Zylindertanks.
3 Hoehe, Durchmesser und Fuellstand müssen in Meter übergeben werden.
4 Der Inhalt wird in Liter zurückgegeben.
5 Im Fehlerfall wird das Errorflag auf TRUE gesetzt und der
6 Wert der Variable Inhalt mit -1 beschrieben.
7 Ein Fehler ist gegeben wenn der Durchmesser kleiner oder gleich Null
8 oder der Fuellstand kleiner als Null oder größer als die Höhe des
9 Tanks ist.
10 **)
11 IF #Durchmesser > 0 AND #Fuellstand >= 0 AND #Fuellstand <= #Hoehe THEN
12     // kein Fehler
13     #er := FALSE;
14     #Inhalt := SQR(#Durchmesser) / 4 * 3.14159 * #Fuellstand * 1000;
15 ELSE
16     // Fehlerfall
17     #er := TRUE;
18     #Inhalt := -1;
19 END_IF;

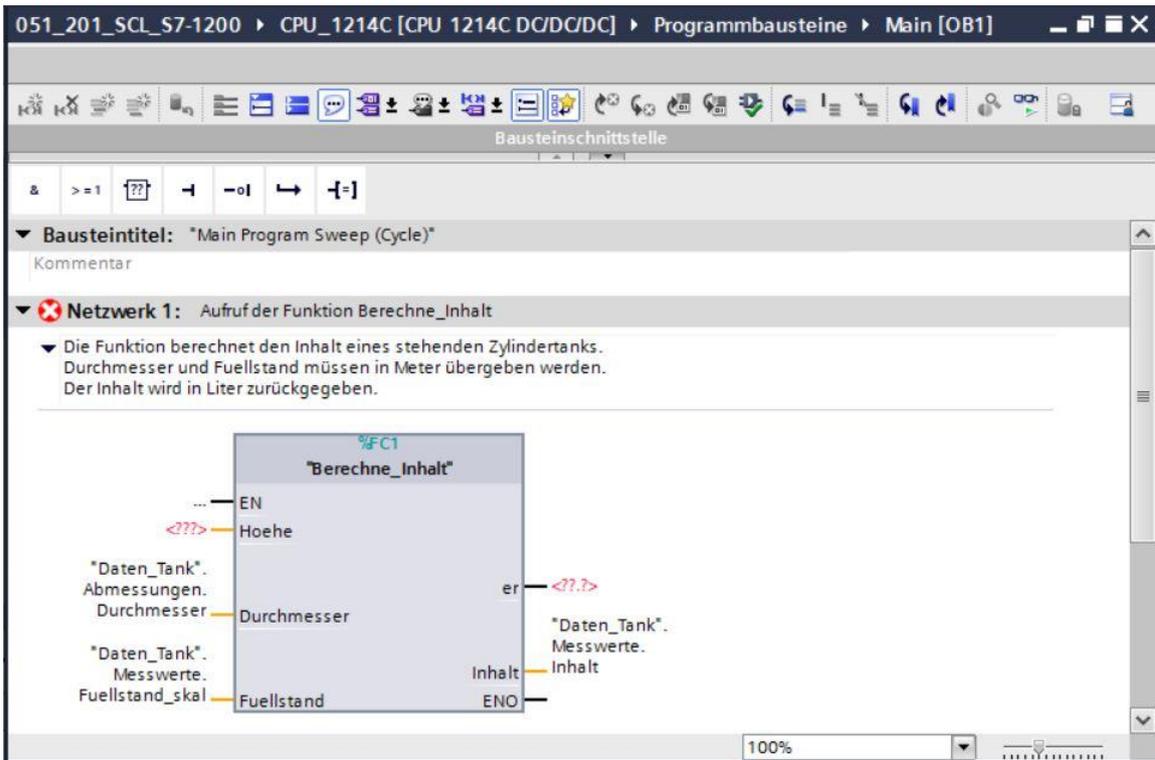
```

7.11 Organisationsbaustein anpassen

- Ⓜ Öffnen Sie den OB1 und aktualisieren Sie die inkonsistenten Bausteinaufrufe durch Klicken auf . (Ⓜ OB1 öffnen Ⓜ )

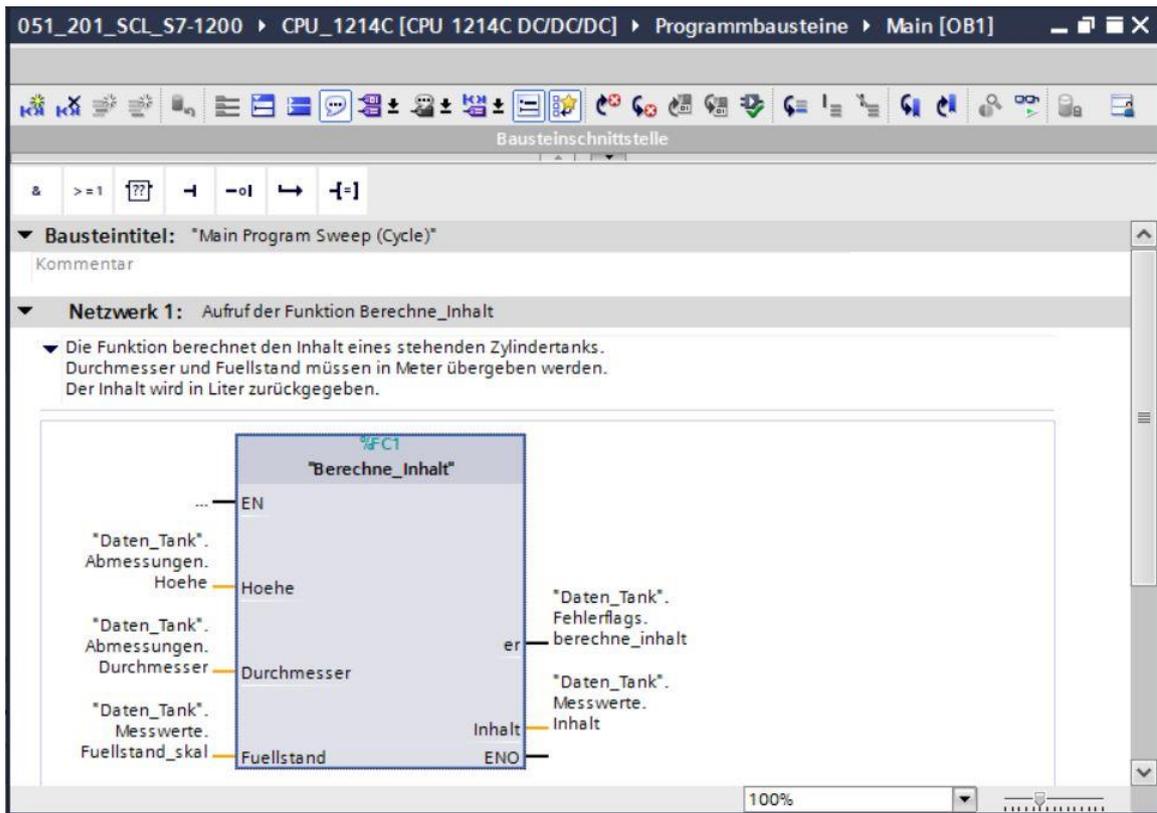


The screenshot shows the Siemens TIA Portal interface. The main workspace displays a function block diagram for 'Berechne_Inhalt' (FC1). The diagram shows inputs: 'Hoehe' (red), 'Durchmesser' (orange), and 'Fuellstand' (yellow). The output is 'Inhalt' (orange). The block is highlighted in red, indicating an inconsistency. The right sidebar shows 'Einfache Anweisungen' (Simple Instructions) and 'Erweiterte Anweisungen' (Advanced Instructions).



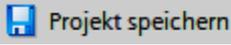
This is a close-up view of the 'Berechne_Inhalt' function block. The block is highlighted in red. The inputs are 'Hoehe' (red, with a red question mark), 'Durchmesser' (orange), and 'Fuellstand' (yellow). The output is 'Inhalt' (orange). The block is labeled '%FC1' and 'Berechne_Inhalt'. The diagram shows the block's name, inputs, and outputs.

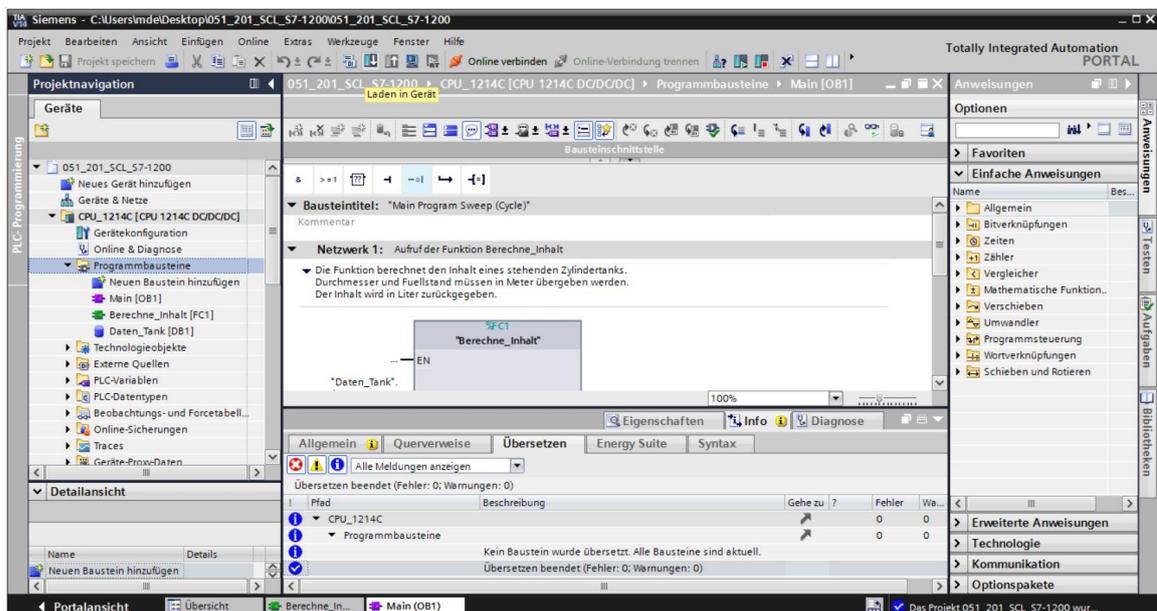
Ⓡ Ergänzen Sie die Beschaltung der Parameter „er“ und „Hoehe“.



7.12 Programm übersetzen, speichern und laden

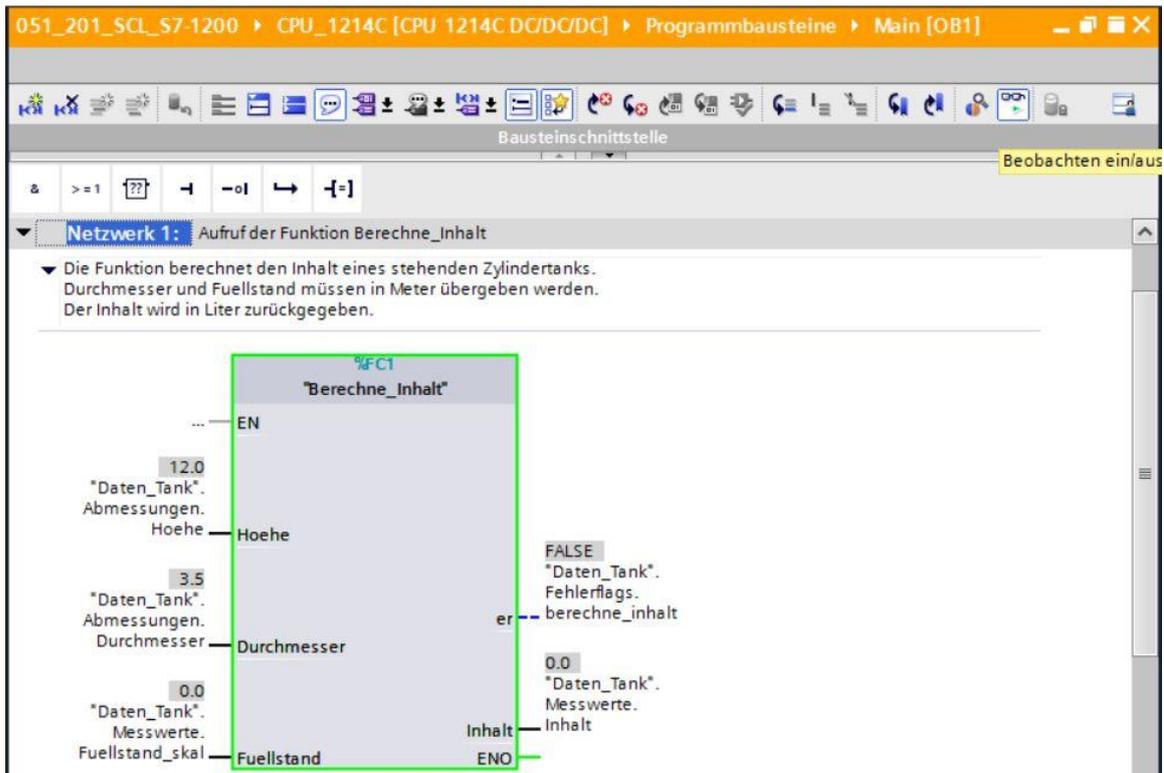
Ⓡ Klicken Sie auf den Ordner „Programmbausteine“, übersetzen Sie das gesamte Programm und speichern Sie es abschließend. Nach erfolgreichem Übersetzen und Speichern laden Sie das Projekt in die Steuerung.

(Ⓡ Programmbausteine Ⓡ  Ⓡ  Ⓡ )



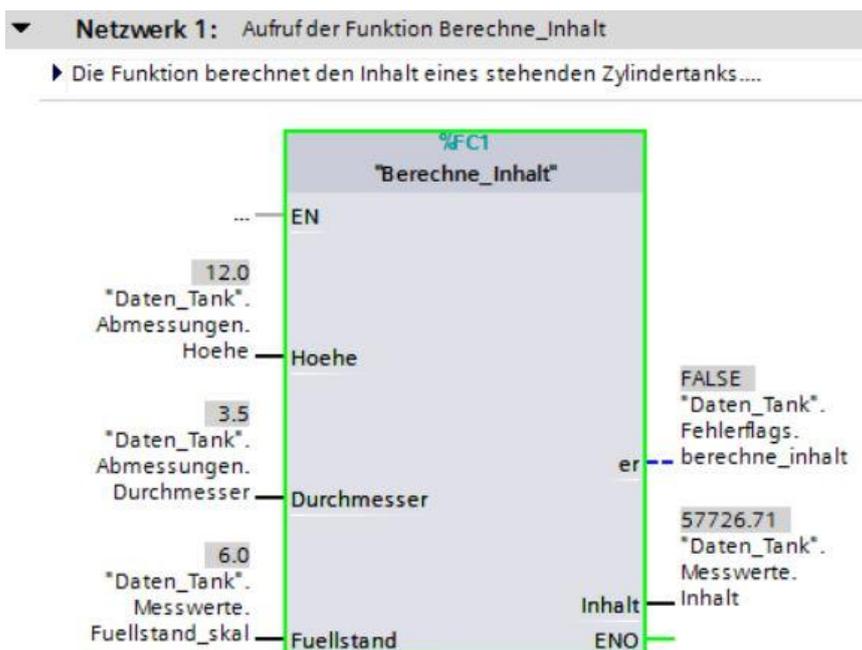
7.13 Organisationsbaustein beobachten und testen

- Ⓡ Klicken Sie im geöffneten OB1 auf das Symbol , um den Baustein zu beobachten.



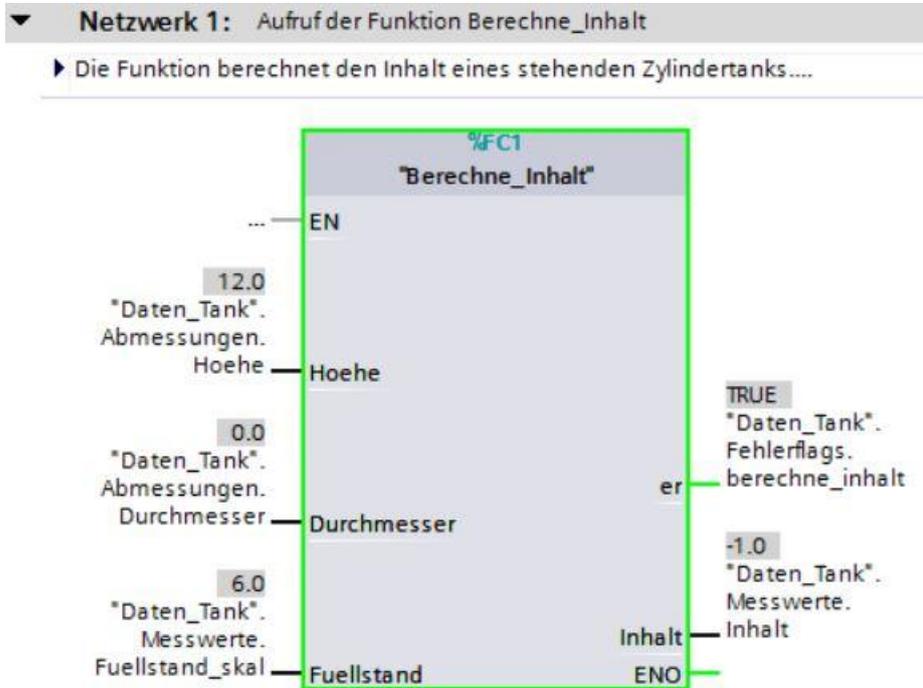
- Ⓡ Testen Sie Ihr Programm, indem Sie einen Wert in die Variable „Fuellstand_skal“ im Datenbaustein schreiben.

(Ⓡ Rechtsklick auf „Fuellstand_skal“ Ⓡ Menü „Steuern“ Ⓡ Operand steuern Ⓡ Wert 6.0 eingeben Ⓡ OK Ⓡ Überprüfen)



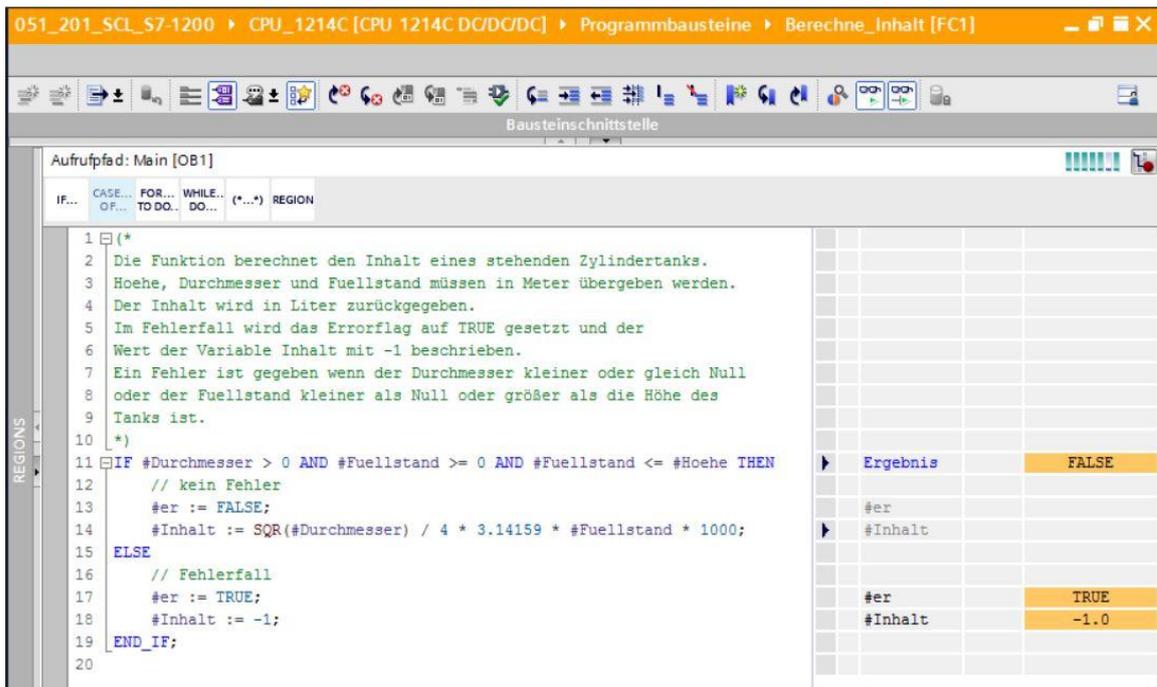
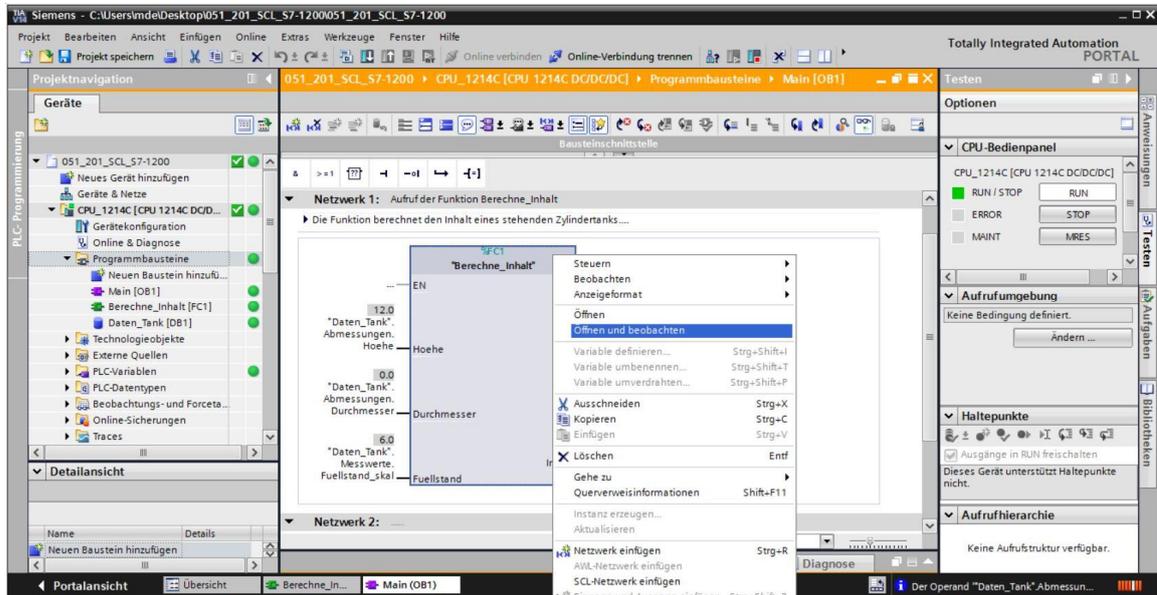
® Testen Sie nachfolgend, ob ein Fehler ausgegeben wird, indem Sie den Durchmesser auf null setzen.

(® Rechtsklick auf „Durchmesser“ ® Menü „Steuern“ ® Operand steuern ® Wert 0.0 eingeben ® OK ® Überprüfen)



7.14 Funktion „Berechne_Inhalt“ beobachten und testen

- Ⓜ Öffnen und beobachten Sie schließlich die Funktion „Berechne_Inhalt“, indem Sie mit Rechtsklick auf die Funktion gehen und den Menüpunkt „Öffnen und beobachten“ auswählen. (Ⓜ Rechtsklick auf Funktion Ⓜ Öffnen und beobachten)

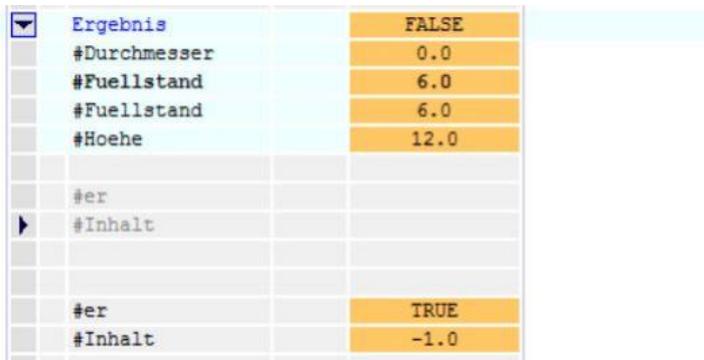
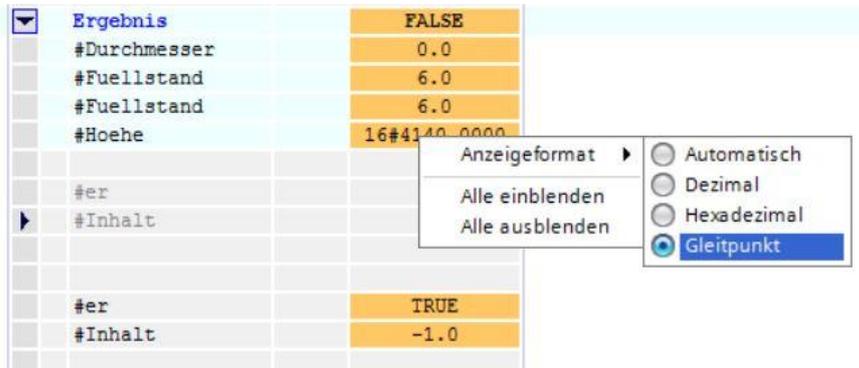


- Ⓜ Sie können die Werte der einzelnen Variablen der IF-Abfrage per Klick auf den schwarzen Pfeil ▼ einblenden. (Ⓜ ▼)

▼	Ergebnis	FALSE
	#Durchmesser	0.0
	#Fuellstand	6.0
	#Fuellstand	6.0
	#Hoehe	12.0
	#er	
▶	#Inhalt	
	#er	TRUE
	#Inhalt	-1.0

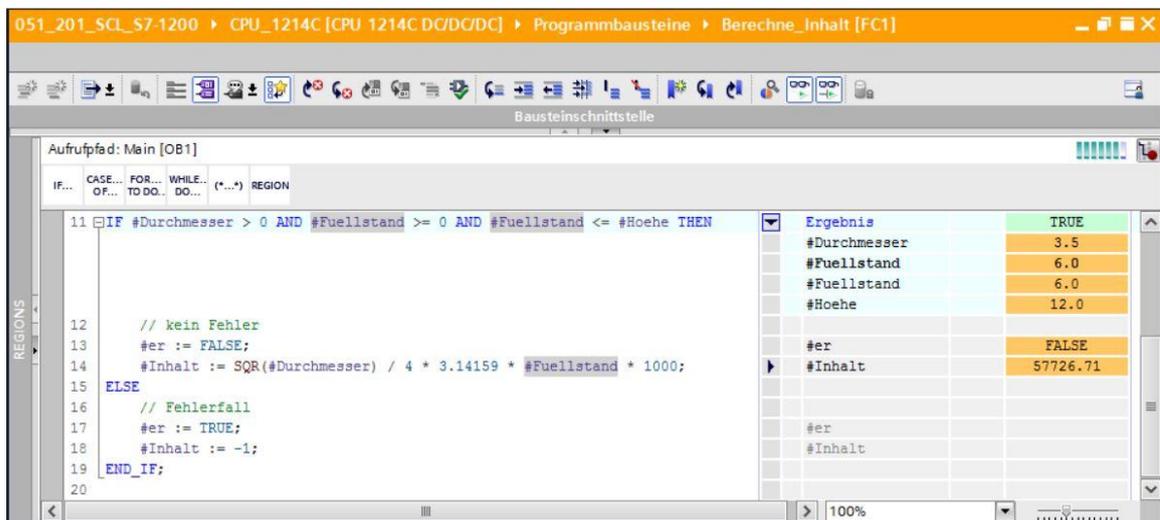
The screenshot shows the Siemens TIA Portal interface. The main window displays a ladder logic program for a function block 'Berechne_Inhalt [FC1]'. The program includes comments in German and a conditional statement (IF) that checks for errors based on diameter, fuel level, and tank height. Below the program, a data table shows the current values of variables: Ergebnis (FALSE), Durchmesser (0.0), Fuellstand (6.0), Hoehe (12.0), er (TRUE), and Inhalt (-1.0). The right-hand side of the interface shows the 'Totally Integrated Automation PORTAL' control panel with buttons for RUN, STOP, ERROR, and MAINT, along with other system settings.

- Ⓜ Das Anzeigeformat kann durch Rechtsklick auf die Variable angepasst werden.
(Ⓜ Rechtsklick auf Variable Ⓜ Anzeigeformat Ⓜ Gleitpunkt)



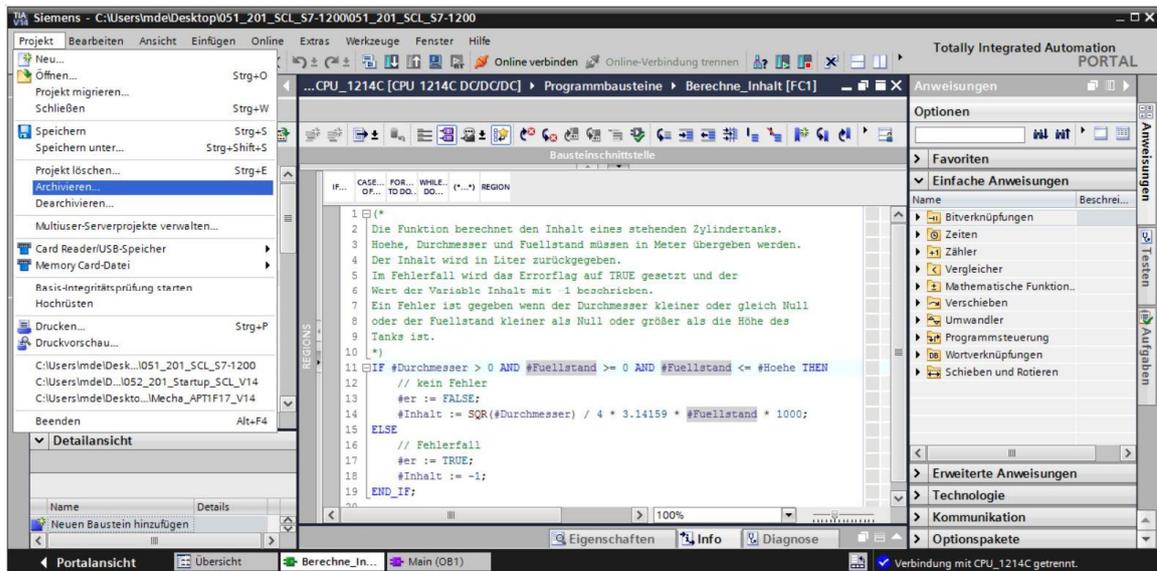
- Ⓜ Testen Sie nun den anderen Zweig der IF-Verzweigung, indem Sie den Durchmesser im OB1 wieder auf 3.5 Meter steuern.

(Ⓜ OB1 öffnen Ⓜ Durchmesser auf 3.5 steuern Ⓜ Funktion öffnen und beobachten)



7.15 Archivieren des Projektes

- Ⓜ Zum Abschluss soll das komplette Projekt noch archiviert werden. Wählen Sie bitte im Menüpunkt Ⓜ „Projekt“ Ⓜ „Archivieren ...“ aus. Öffnen Sie den Ordner, in welchem Sie Ihr Projekt archivieren wollen und speichern Sie es als Dateityp „TIA Portal-Projektarchiv“ ab. (Ⓜ Projekt Ⓜ Archivieren Ⓜ TIA Portal-Projektarchiv Ⓜ Dateiname: SCE_DE_051-201 SCL_S7-1200... Ⓜ Archivieren)



8 Checkliste

Nr.	Beschreibung	Geprüft
1	Übersetzen erfolgreich und ohne Fehlermeldung	
2	Laden erfolgreich und ohne Fehlermeldung	
3	Operand steuern (Durchmesser = 0.0) Ergebnis Variable Inhalt = -1 Ergebnis Variable „er“ = TRUE	
4	Operand steuern (Durchmesser = 3.5 und Fuellstand_skal = 0) Ergebnis Inhalt = 0 Ergebnis Variable „er“ = FALSE	
5	Operand steuern (Fuellstand_ska l= 6.0) Ergebnis Inhalt = 57726.72 Ergebnis Variable „er“ = FALSE	
6	Operand steuern (Fuellstand_skal = 12.0) Ergebnis Inhalt = 115453.4 Ergebnis Variable „er“ = FALSE	
7	Operand steuern (Fuellstand_skal = 14.0) Ergebnis Inhalt = -1 Ergebnis Variable „er“ = TRUE	
8	Projekt erfolgreich archiviert	

9 Übung

9.1 Aufgabenstellung – Übung

In dieser Übung wird eine Funktion „Skalieren“ programmiert. Das Programm soll allgemeingültig für jegliche positiven Analogwerte anwendbar sein. In unserer Beispielaufgabe „Tank“ wird der Füllstand über einen Anlogsensor eingelesen und mittels dieser Funktion skaliert im Datenbaustein abgelegt.

Im Fehlerfall soll der Baustein das Errorflag „er“ auf TRUE und als Ergebnis den Parameter „Analog_skal“ auf null setzen. Ein Fehlerfall besteht, wenn der Parameter „mx“ kleiner oder gleich „mn“ ist.

Die Funktion muss folgende Parameter beinhalten.

Input	Datentyp	Kommentar
Analog_per	INT	Analogwert von der Peripherie zwischen 0..27648
mx	REAL	Maximum des neuen Maßstabs
mn	REAL	Minimum des neuen Maßstabs
Output		
er	BOOL	Fehlerflag, kein Fehler = 0, Fehler = 1
Analog_skal	REAL	Analogwert skaliert zwischen mn..mx Im Fehlerfall = 0

Zur Lösung der Aufgabe wird folgende Formel verwendet:

$$\# \text{Analog_skal} = \frac{\# \text{Analog_per}}{27648} \cdot (\# \text{mx} - \# \text{mn}) + \# \text{mn}$$

Für diese Übungsaufgabe ist ein Anlogsignal notwendig. Der hierzu verwendete Operand muss in die PLC-Variablentabelle eingetragen werden.

Name	Datentyp	Adresse	Kommentar
B1	INT	%EW64	Füllstand zwischen 0..27648

9.2 Planung

Planen Sie nun selbstständig die Umsetzung der Aufgabenstellung!

9.3 Checkliste – Übung

Nr.	Beschreibung	Geprüft
1	Operand in PLC-Variablen-Tabelle eingefügt	
2	Funktion FC: „Skalieren“ erstellt	
3	Schnittstelle definiert	
4	Funktion programmiert	
5	Funktion „Skalieren“ ins Netzwerk 1 des OB1 eingefügt	
6	Eingangsvariablen verschaltet	
7	Ausgangsvariablen verschaltet	
8	Übersetzen erfolgreich und ohne Fehlermeldung	
9	Laden erfolgreich und ohne Fehlermeldung	
10	Analogwert für Füllstand auf null gesetzt Ergebnis Fuellstand_skal = 0 Ergebnis er = FALSE	
11	Analogwert für Füllstand auf 27648 gesetzt Ergebnis Fuellstand_skal = 12.0 Ergebnis er = FALSE	
12	Analogwert für Füllstand auf 13824 Ergebnis Fuellstand_skal = 6.0 Ergebnis er = FALSE	
13	Operand steuern (mx = 0.0) Ergebnis Fuellstand_skal = 0 Ergebnis Variable er = TRUE	
14	Projekt erfolgreich archiviert	

10 Weiterführende Information

Zur Einarbeitung bzw. Vertiefung finden Sie als Orientierungshilfe weiterführende Informationen, wie z.B. Getting Started, Videos, Tutorials, Apps, Handbücher, Programmierleitfaden und Trial Software/Firmware, unter nachfolgendem Link:

[siemens.com/sce/s7-1200](https://www.siemens.com/sce/s7-1200)

Voransicht „Weiterführende Informationen“

☐ Getting Started, Videos, Tutorials, Apps, Handbücher, Trial-SW/Firmware

- TIA Portal Videos
- TIA Portal Tutorial Center
- Getting Started
- Programmierleitfaden
- Leichter Einstieg in SIMATIC S7-1200
- Download Trial Software/Firmware
- Technische Dokumentation SIMATIC Controller
- Industry Online Support App
- TIA Portal, SIMATIC S7-1200/1500 Überblick
- TIA Portal Website
- SIMATIC S7-1200 Website
- SIMATIC S7-1500 Website

Weitere Informationen

Siemens Automation Cooperates with Education
[siemens.de/sce](https://www.siemens.de/sce)

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

SCE Trainer Pakete
[siemens.de/sce/tp](https://www.siemens.de/sce/tp)

SCE Kontakt Partner
[siemens.de/sce/contact](https://www.siemens.de/sce/contact)

Digital Enterprise
[siemens.de/digital-enterprise](https://www.siemens.de/digital-enterprise)

Industrie 4.0
[siemens.de/zukunft-der-industrie](https://www.siemens.de/zukunft-der-industrie)

Totally Integrated Automation (TIA)
[siemens.de/tia](https://www.siemens.de/tia)

TIA Portal
[siemens.de/tia-portal](https://www.siemens.de/tia-portal)

SIMATIC Controller
[siemens.de/controller](https://www.siemens.de/controller)

SIMATIC Technische Dokumentation
[siemens.de/simatic-doku](https://www.siemens.de/simatic-doku)

Industry Online Support
support.industry.siemens.com

Katalog- und Bestellsystem Industry Mall
mall.industry.siemens.com

Siemens AG
Digital Factory
Postfach 4848
90026 Nürnberg
Deutschland

Änderungen und Irrtümer vorbehalten
© Siemens AG 2018

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