**이러한 학습-/교육 문서와 일치하는 SCE 교육 담당자 패키지**



교육-/학습 문서  
Siemens Automation Cooperates with Education (SCE) | 버전 V14 SP1부터

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

**siemens.com/sce**

TIA Portal Module 051-201

SCL 및 SIMATIC S7-1200을 사용한

고급 언어 프로그래밍

* **SIMATIC S7-1200 AC/DC/RELAY(6개 세트) "TIA Portal"**  
  주문 번호: 6ES7214-1BE30-4AB3
* **SIMATIC S7-1200 DC/DC/DC(6개 세트) "TIA Portal"**  
  주문 번호: 6ES7214-1AE30-4AB3
* **업그레이드 SIMATIC STEP 7 BASIC V14 SP1(S7-1200용)(6개 세트) "TIA Portal"**  
  주문 번호: 6ES7822-0AA04-4YE5

이러한 교육 담당자 패키지는 필요 시 후임자 패키지로 대체됩니다. 다음 웹 사이트에서 현재 사용 가능한 SCE 패키지를 대략적으로 확인할 수 있습니다. [siemens.com/sce/tp](http://www.siemens.com/sce/tp)  
  
**교육 연장**

지역별 Siemens SCE 교육 연장은 지역 SCE 담당자에게 문의하십시오. [siemens.com/sce/contact](http://www.siemens.com/contact)

**SCE 관련 추가 정보**

[siemens.com/sce](http://www.siemens.com/sce)  
  
  
**사용 관련 정보**

통합 자동화 솔루션 TIA(Totally Integrated Automation)에 대한 본 SCE 학습-/교육 문서는 특히 공공 교육 기관 및 R&D 기관의 교육 목적으로 "SCE(Siemens Automation Cooperates with Education)" 프로그램을 위해 준비되었습니다. Siemens AG는 내용을 보장하지 않습니다.

이 문서는 Siemens 제품/시스템에 대한 최초 교육용으로만 사용해야 합니다. 이 문서의 전체 또는 일부를 복사해 교육을 받는 사람들에게 제공해 교육 범위 내에서 사용할 수 있습니다. 이 학습-/교육 문서 배포 또는 복사와 내용 공유는 교육 목적의 공개 교육 및 고등 교육 기관에서만 허용됩니다.

그 외의 경우에는 다음 Siemens AG 담당자의 서면 동의가 필요합니다. Roland Scheuerer roland.scheuerer@siemens.com.

이를 위반하면 법적 책임을 지게 됩니다. 특히 특허가 부여되거나 실용신안 또는 디자인이 등록되어 있는 경우 번역을 포함해 모든 권리가 보장됩니다.

산업 고객을 위한 과정에서의 사용은 명시적으로 허용되지 않습니다. 본 학습-/교육 문서를 상업적으로 사용하는 데 동의하지 않습니다.

TU Dresden, 특히 Prof. Dr.-Ing. Leon Urbas 및 Michael Dziallas Engineering Corporation을 비롯한 모든 관계자들께 이 학습-/교육 문서를 준비하는 동안 보내주신 성원에 대해 감사를 표하고자 합니다.

목차

[1 목적 4](#_Toc504388219)

[2 요구사항 4](#_Toc504388220)

[3 하드웨어 및 소프트웨어 전제조건 5](#_Toc504388221)

[4 이론 6](#_Toc504388222)

[4.1 S7-SCL 프로그래밍 언어 6](#_Toc504388223)

[4.2 S7-SCL 개발 환경 6](#_Toc504388224)

[5 과제 9](#_Toc504388225)

[5.1 태스크 탱크 볼륨 예 9](#_Toc504388226)

[5.2 샘플 태스크 확장 9](#_Toc504388227)

[6 계획 수립 9](#_Toc504388228)

[6.1 글로벌 데이터 블록 "Data\_Tank" 9](#_Toc504388229)

[6.2 “Calculate\_Volume” 펑션 10](#_Toc504388230)

[6.3 “Calculate\_Volume” 펑션 확장 10](#_Toc504388231)

[7 단계별 따라 해보기 11](#_Toc504388232)

[7.1 기존 프로젝트 압축 풀기 11](#_Toc504388233)

[7.2 새 이름에서 프로젝트 저장 12](#_Toc504388234)

[7.3 “Data\_Tank” 데이터 블록 생성 12](#_Toc504388235)

[7.4 “Calculate\_Volume” 펑션 생성 14](#_Toc504388236)

[7.5 “Calculate\_Volume” 펑션의 인터페이스 지정 15](#_Toc504388237)

[7.6 “Calculate\_Volume” 펑션 프로그래밍 16](#_Toc504388238)

[7.7 “Main [OB1]” 오거나이제이션 블록 프로그래밍 17](#_Toc504388239)

[7.8 프로그램 컴파일 및 다운로드 19](#_Toc504388240)

[7.9 오거나이제이션 블록 모니터링 및 테스트 20](#_Toc504388241)

[7.10 “Calculate\_Volume” 펑션 확장 22](#_Toc504388242)

[7.11 오거나이제이션 블록 사용자 정의 27](#_Toc504388243)

[7.12 프로그램 컴파일 및 다운로드 28](#_Toc504388244)

[7.13 오거나이제이션 블록 모니터링 및 테스트 29](#_Toc504388245)

[7.14 “Calculate\_Volume” 펑션 모니터링 및 테스트 31](#_Toc504388246)

[7.15 프로젝트 아카이브 34](#_Toc504388247)

[8 체크리스트 35](#_Toc504388248)

[9 연습 36](#_Toc504388249)

[9.1 과제 설명 – 연습 36](#_Toc504388250)

[9.2 계획 수립 37](#_Toc504388251)

[9.3 체크리스트 – 연습 37](#_Toc504388252)

[10 추가 정보 38](#_Toc504388253)

SCL 및 S7-1200을 사용한   
고급 언어 프로그래밍

# 목적

이 섹션에서는 SCL 고급 언어의 기본 기능을 학습합니다. 논리 프로그래밍 오류를 제거하기 위한 테스트 함수도 설명합니다.

섹션 3에 나열된 SIMATIC S7 컨트롤러를 사용할 수 있습니다.

# 요구사항

이 섹션은 SIMATIC S7-1200의 하드웨어 컨피규레이션을 기반으로 합니다. 디지털 입력 및 출력 카드가 포함된 하드웨어 컨피규레이션으로 실행할 수 있습니다. 이 섹션을 실행하려면 다음 프로젝트를 사용할 수 있습니다. 예:

"SCE\_EN\_011\_101\_Hardware\_Configuration\_CPU1214C…..zap14"

Pascal과 같은 고급 언어 프로그래밍도 학습해야 합니다.

# 하드웨어 및 소프트웨어 전제조건

1. 엔지니어링 스테이션: 전제조건은 하드웨어 및 운영 체제입니다.

(추가 정보는 TIA Portal 설치 DVD에 추가 정보 참조).

**2** TIA Portal의 SIMATIC STEP 7 Basic 소프트웨어 – V14 SP1 기준

**3** SIMATIC S7-1200 컨트롤러, 예: CPU 1214C DC/DC/DC – 펌웨어 V4.2.1 이상

**4** 엔지니어링 스테이션과 컨트롤러 간의 이서네트 연결



**2** SIMATIC STEP 7 Basic (TIA Portal),  
 V14 SP1 기준



**1** 엔지니어링 스테이션

**4** 이서네트 연결



**3** SIMATIC S7-1200 컨트롤러

# 이론

## S7-SCL 프로그래밍 언어

S7-SCL(Structured Control Language)은 Pascal 기반 고급 프로그래밍 언어이며 구조화된 프로그래밍이 가능합니다. 언어는 표준 DIN EN-61131-3(IEC 61131-3)에서 지정된 “SFC(Sequential Function Chart)” 언어에 해당합니다. 고급 언어 요소에 추가하여 S7-SCL은 입력, 출력, 타이머, 비트 메모리, 블록 호출 등과 같은 언어 요소로 일반 PLC 요소도 포함합니다. S7-SCL은 STL(Statement List), LAD(Ladder Logic) 및 FBD(Function Block Diagram)와 함께 특히 STEP 7 블록 개념을 지원하므로 표준을 충족시키는 프로그램 블록을 만들 수 있습니다. 즉, S7-SCL은 프로그래밍 언어 LAD, FBD 및 STL을 사용하여 STEP 7 프로그래밍 소프트웨어를 보완하고 확장합니다.

모든 펑션을 직접 생성할 필요는 없으며 CPU의 운영 체제에 있는 시스템 펑션 및 시스템 펑션 블록과 같이 사전에 컴파일된 블록을 사용할 수 있습니다.

S7-SCL로 프로그래밍한 블록을 STL, LAD 및 FBD 블록과 조합시킬 수 있습니다. 즉, S7-SCL로 프로그래밍된 블록은 STL, LAD 또는 FBD로 프로그래밍된 다른 블록을 호출할 수 있습니다. 이에 따라 S7-SCL 블록도 STL, LAD 및 FBD 프로그램으로 호출될 수 있습니다.

S7-SCL 테스트 펑션을 사용하면 오류 방지 컴파일에서 논리적 프로그래밍 오류를 검색할 수 있습니다.

## S7-SCL 개발 환경

S7-SCL 사용과 관련하여 S7-SCL 및 STEP 7의 특정 속성에 맞춘 개발 환경이 있습니다. 이 개발 환경은 편집기/컴파일러 및 디버거로 구성됩니다.

**S7-SCL S7-1500**

디버거

편집기/컴파일러

편집기/컴파일러

S7-SCL 편집기는 모든 종류의 텍스트를 편집하는 데 사용할 수 있는 텍스트 편집기입니다. S7-SCL 편집기의 주된 태스크은 STEP 7 프로그램용 블록을 생성하고 편집하는 것입니다. 입력 도중 기본 구문 검사가 수행되므로 프로그래밍에서 발생하는 오류를 피할 수 있습니다. 구문 오류는 다른 색으로 표시됩니다.

편집기는 다음과 같은 옵션을 제공합니다.

* + S7-SCL 언어에서 S7 블록 프로그래밍.
  + 끌어다 놓기로 언어 요소 및 블록 호출을 손쉽게 추가 가능.
  + 프로그래밍 도중 직접 구문 검사.
  + 필요에 맞게 편집기 사용자 정의 가능(예: 구문에 따라 다른 언어 요소에 대한 색 지정).
  + 컴파일을 통해 완료된 블록 검사.
  + 컴파일 도중 발생하는 모든 오류 및 경고 표시.
  + 선택 사항으로 오류 설명 및 문제 해결 정보가 포함된 블록의 오류 위치 현지화.

디버거

S7-SCL 디버거를 사용하면 자동화 시스템(AS)에서 실행되는 동안 프로그램을 검사할 수 있으므로 잠재적인 논리 오류를 찾을 수 있습니다.

S7-SCL는 두 가지 다른 테스트 모드를 제공합니다.

* + 연속 모니터링
  + 단계별 모니터링

“연속 모니터링”으로 블록 내에서 명령 그룹을 테스트할 수 있습니다. 테스트 도중 태그 및 파라미터 값이 시간순으로 표시되고 가능하면 주기적으로 업데이트됩니다.

“단계별 모니터링”으로 논리적 프로그램 순서를 따를 수 있습니다. 프로그램 알고리즘을 명령별로 실행하고 결과 창에서 처리된 태그 내용이 어떻게 변경되었는지 조사할 수 있습니다.

사용 중인 CPU 유형이 “단계별 모니터링” 사용 여부를 결정합니다. CPU가 중단점 사용을 지원해야 합니다. 이 문서에서 사용되는 CPU는 중단점을 지원하지 않습니다.

# 과제

## 태스크 탱크 볼륨 예

첫 번째 부분에서 탱크 볼륨 계산을 프로그래밍합니다.

## 샘플 태스크 확장

두 번째 부분에서 태스크가 확장되어 오류 평가를 프로그래밍합니다.

# 계획 수립

탱크는 수직 실린더 형태입니다. 볼륨의 채움 수준이 아날로그 센서로 측정됩니다. 첫 번째 테스트에서 채움 수준 값은 표준화된 형태(미터 단위)로 사용할 수 있어야 합니다.

탱크의 직경 및 높이와 같은 글로벌 파라미터는 “Data\_Tank” 데이터 블록에 저장해야 합니다.

내용 계산을 위한 프로그램은 “Calculate\_Volume” 펑션으로 작성되어야 하고 파라미터는 ‘미터’ 또는 ‘리터’ 단위를 사용해야 합니다.

## 글로벌 데이터 블록 "Data\_Tank"

글로벌 파라미터는 글로벌 데이터 블록의 여러 구조에 저장됩니다.

|  |  |  |  |
| --- | --- | --- | --- |
| 이름 | 데이터 유형 | 초기값 | 코멘트 |
| 치수 | STRUCT |  |  |
| 높이 | REAL | 12.0 | 미터 |
| 직경 | REAL | 3.5 | 미터 |
| measured\_data | STRUCT |  |  |
| filling\_level\_per | INT | 0 | 범위 0...27648 |
| filling\_level\_scal | REAL | 0.0 | 범위 0...12.0 |
| volume\_liquid | REAL | 0.0 | 리터 |
| fault\_flags | STRUCT |  |  |
| calculate\_volume | BOOL |  | fault == true |

표 1: “Data\_Tank” 데이터 블록의 파라미터

## “Calculate\_Volume” 펑션

이 블록은 탱크 내용물을 리터 단위로 계산합니다.

첫 번째 단계에서 전송된 파라미터를 합리적으로 검사하지 않습니다.

이 단계에 다음과 같은 파라미터가 필요합니다.

|  |  |  |
| --- | --- | --- |
| 입력 | 데이터 유형 | 코멘트 |
| 직경 | REAL | 직경이 원통형인 탱크(미터) |
| Filling\_level | REAL | 액체의 채움 수준(미터) |
| 출력 |  |  |
| 볼륨 | REAL | 탱크의 액체 볼륨(리터) |

표 2: 첫 번째 단계에서 “Calculate\_Volume” 펑션 파라미터

수직 실린더 볼륨을 계산하는 공식이 태스크를 해결하는 데 사용됩니다. 변환 계수 1000이 결과(리터)를 계산하는 데 사용됩니다.

2

 => 

## “Calculate\_Volume” 펑션 확장

두 번째 단계는 직경이 0보다 큰지 확인하는 것입니다. 또한 채움 수준이 0보다 크거나 같은지 또는 탱크 높이보다 작거나 큰지를 확인할 수도 있습니다. 오류가 발생하는 경우 새 파라미터 “er”이 TRUE로 설정되고 파라미터 “Volume”이 값 -1로 설정됩니다.

이렇게 하려면 파라미터 “er” 및 “Height”로 인터페이스를 확장합니다.

|  |  |  |
| --- | --- | --- |
| 입력 | 데이터 유형 | 코멘트 |
| 높이 | REAL | 높이가 원통형인 탱크(미터) |
| 직경 | REAL | 직경이 원통형인 탱크(미터) |
| Filling\_level | REAL | 액체의 채움 수준(미터) |
| 출력 |  |  |
| er | BOOL | 결함 플래그, fault == true |
| 볼륨 | REAL | 탱크의 액체 볼륨(리터) |

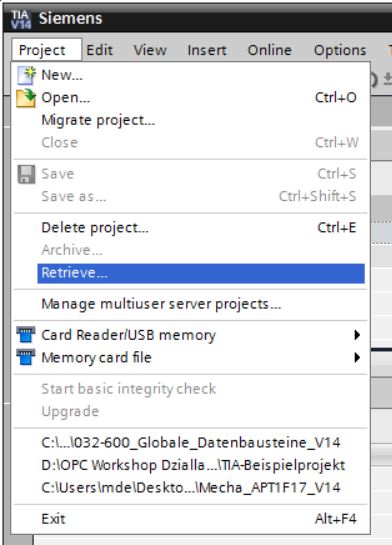
표 3: 두 번째 단계에서 “Calculate\_Volume” 펑션 파라미터

# 단계별 따라 해보기

아래에는 계획을 수립하는 방법에 대한 지침이 나와 있습니다. 모든 내용을 이미 충분히 숙지했다면 숫자가 표시된 단계에만 집중하셔도 좋습니다. 그렇지 않다면, 아래 설명된 지침의 단계를 따라가면 됩니다.

## 기존 프로젝트 압축 풀기

* 프로그래밍을 시작하기 전에 하드웨어 구성이 포함된 프로젝트가 필요합니다. (예: SCE\_EN\_012-101\_Hardware\_configuration\_S7-1516F\_....zap)   
  기존 프로젝트를 압축 풀기하려면 → Project → Retrieve의 프로젝트 뷰에서 해당하는 아카이브를 선택해야 합니다. “Open”로 선택을 확정합니다. (→ Project → Retrieve → .zap 보관 위치 선택 → Open)

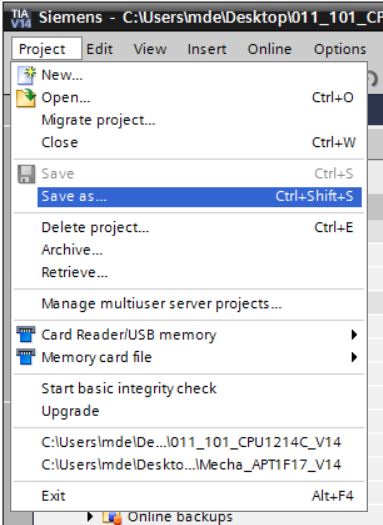


* 다음으로 검색할 프로젝트를 저장할 대상 디렉터리를 선택할 수 있습니다. “OK”을 클릭해 선택을 확정합니다. (→ Project → Save as… → OK)

## 새 이름에서 프로젝트 저장

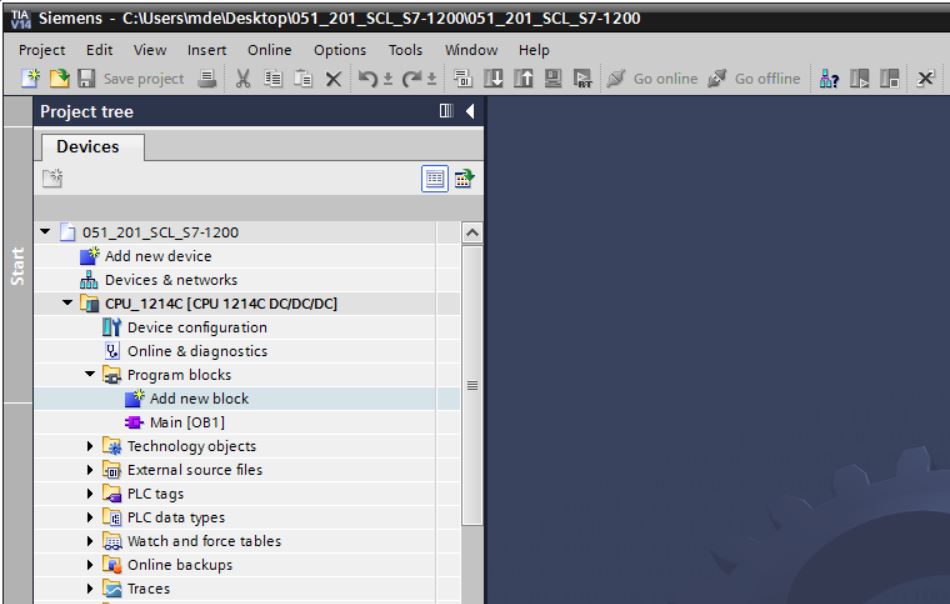
* 열려 있는 프로젝트를 052-201\_Startup\_SCL 이름으로 저장합니다.

(→ Project → Save as… → 052-201\_Startup\_SCL → OK)

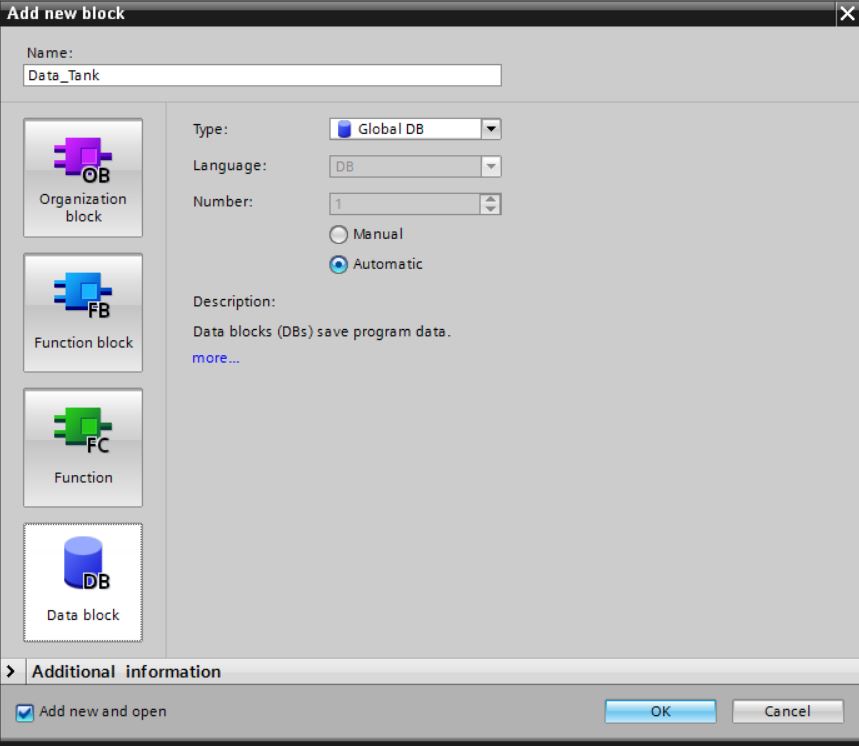


## “Data\_Tank” 데이터 블록 생성

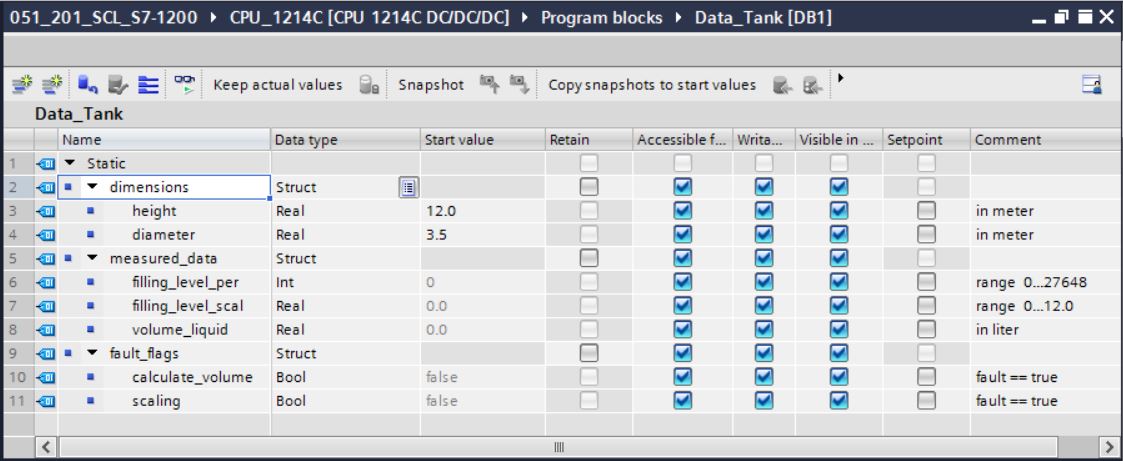
* 프로젝트 뷰에서 → Program blocks으로 이동하고 → Add new block 에서 더블 클릭하여 새 블록을 생성합니다.



* 이제 데이터 블록을 선택하고 이름을 입력합니다. ( →→ “Data\_Tank” → OK)



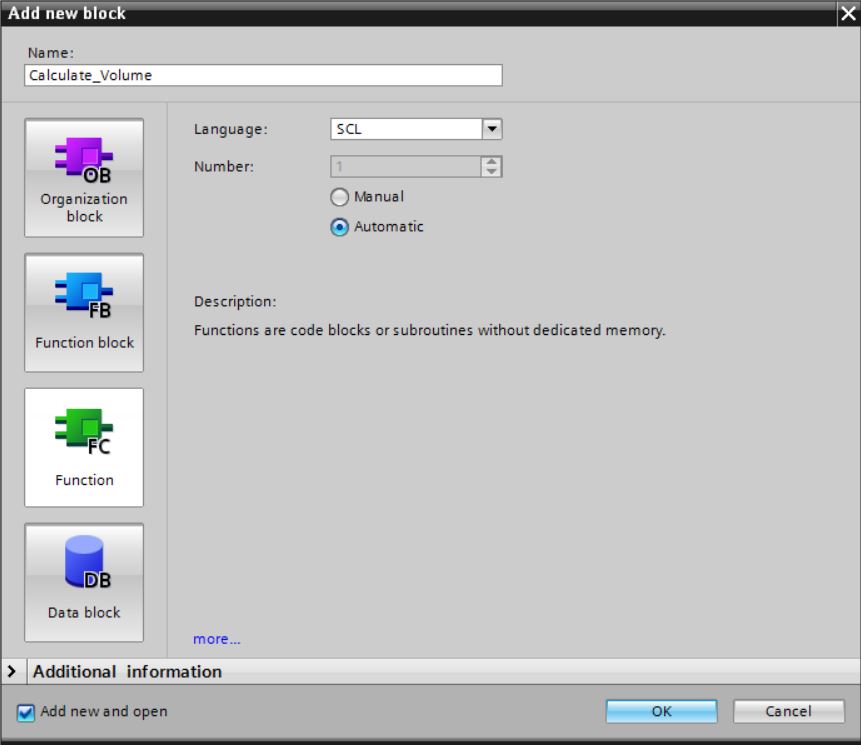
* 이제 데이터 유형, 초기값 및 설명과 함께 아래에 나열된 태그 이름을 입력합니다.



## “Calculate\_Volume” 펑션 생성

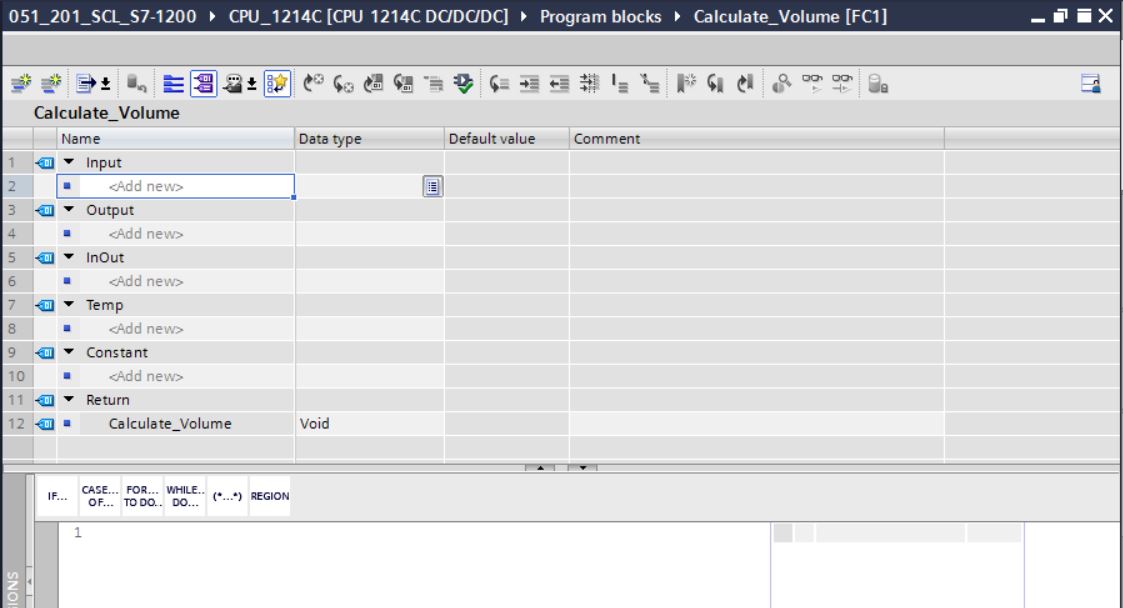
* 이제 펑션을 추가하고 이름을 입력하며 언어를 선택합니다.

( → Add new block → → “Calculate\_Volume” → SCL → OK)

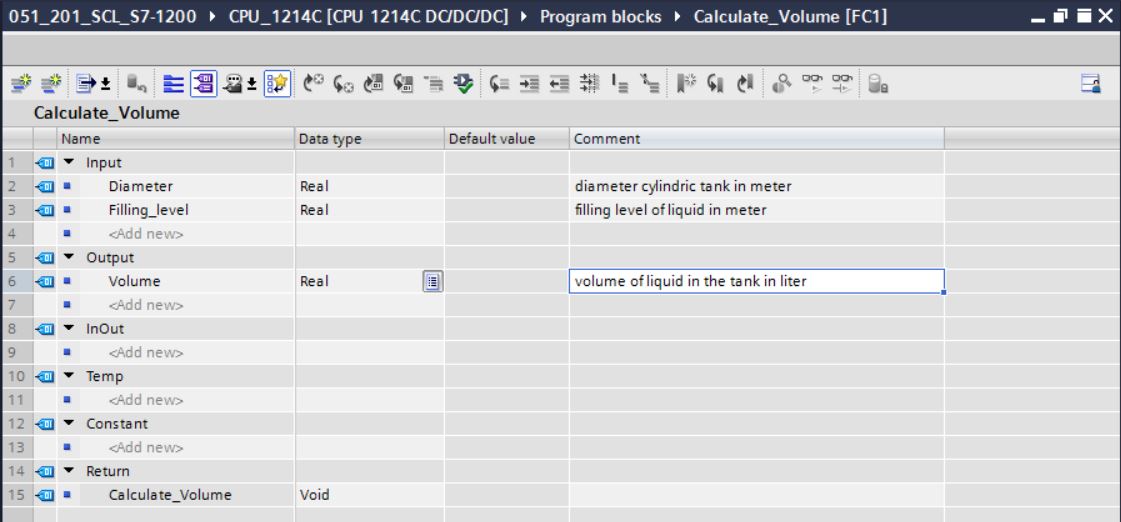


## “Calculate\_Volume” 펑션의 인터페이스 지정

* 프로그래밍 뷰의 상단 섹션은 펑션의 인터페이스 설명을 표시합니다.

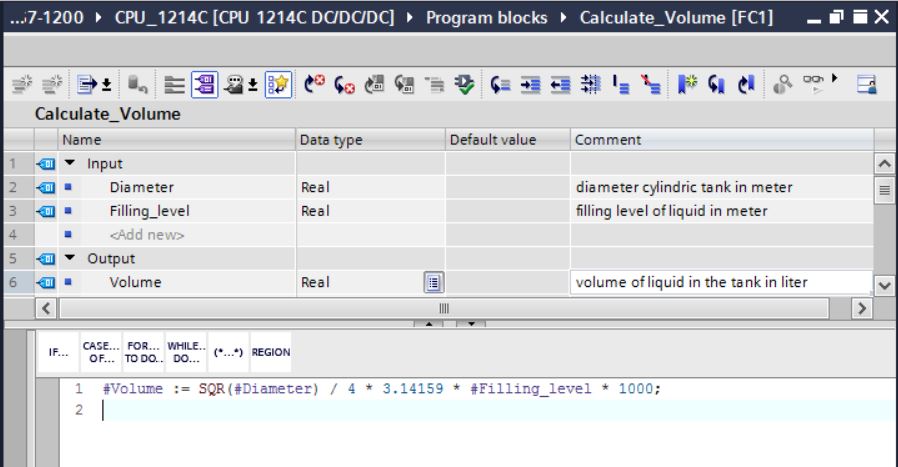


* 다음 입출력 파라미터를 생성합니다. ( → Name → Data type → Comment)

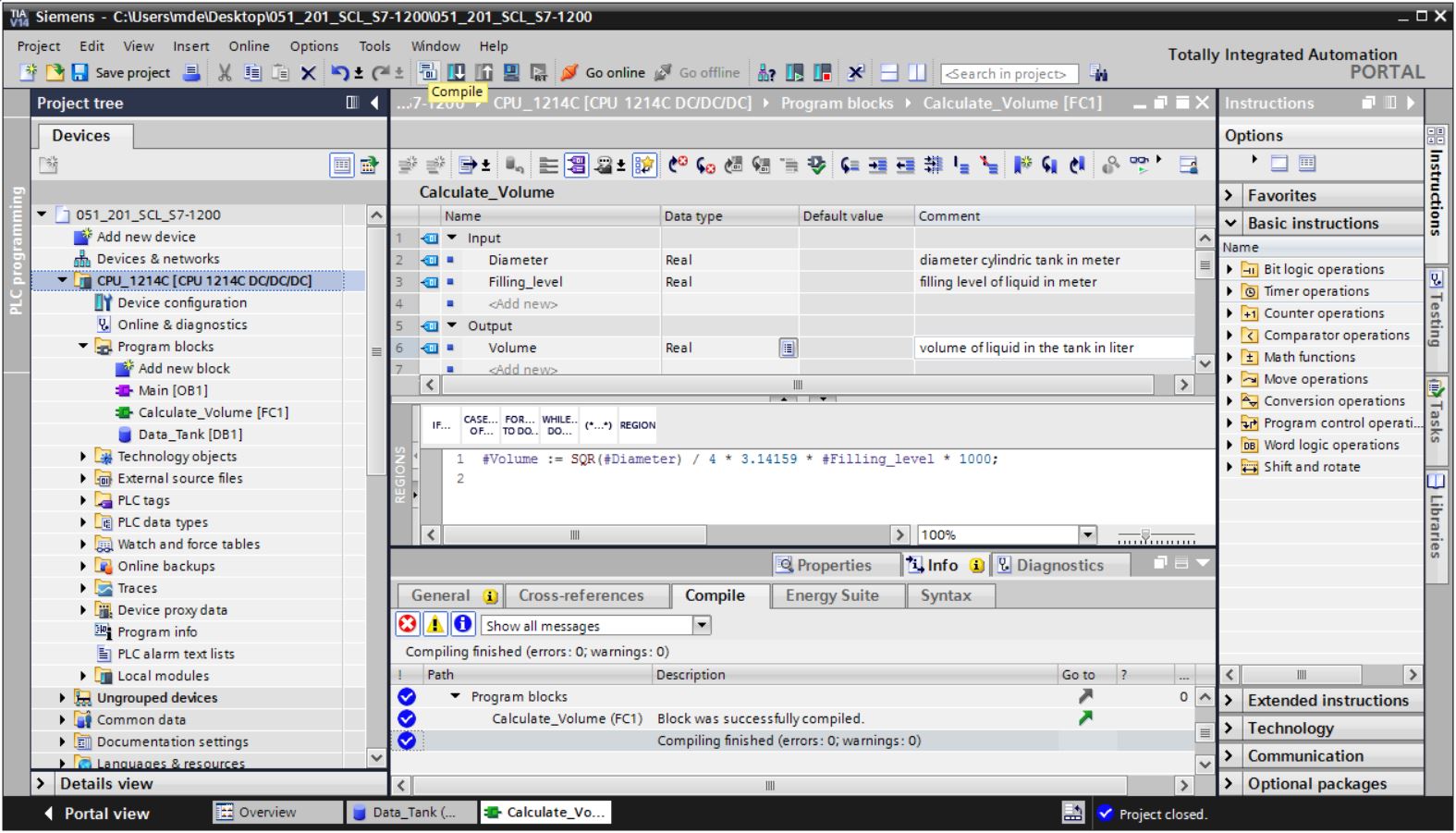


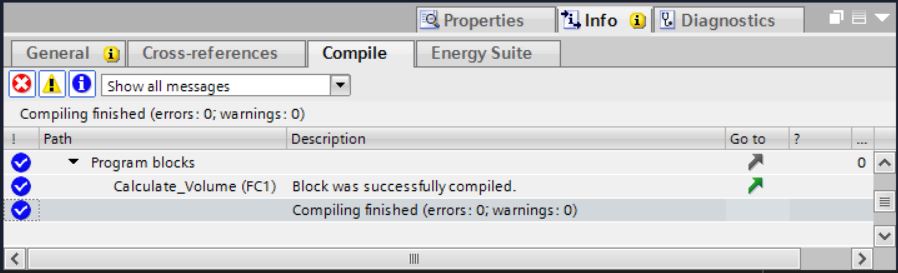
## “Calculate\_Volume” 펑션 프로그래밍

* 아래 표시된 프로그램을 입력합니다. ( → 프로그램 입력)



* 이제 프로그램을 컴파일하고 구문 오류를 검사합니다. 이는 프로그래밍 아래 조사 창에 표시됩니다. 오류를 수정하고 프로그램을 다시 컴파일을 합니다. 그런 다음 프로그램을 저장합니다. ( →  → 오류 제거 →  )

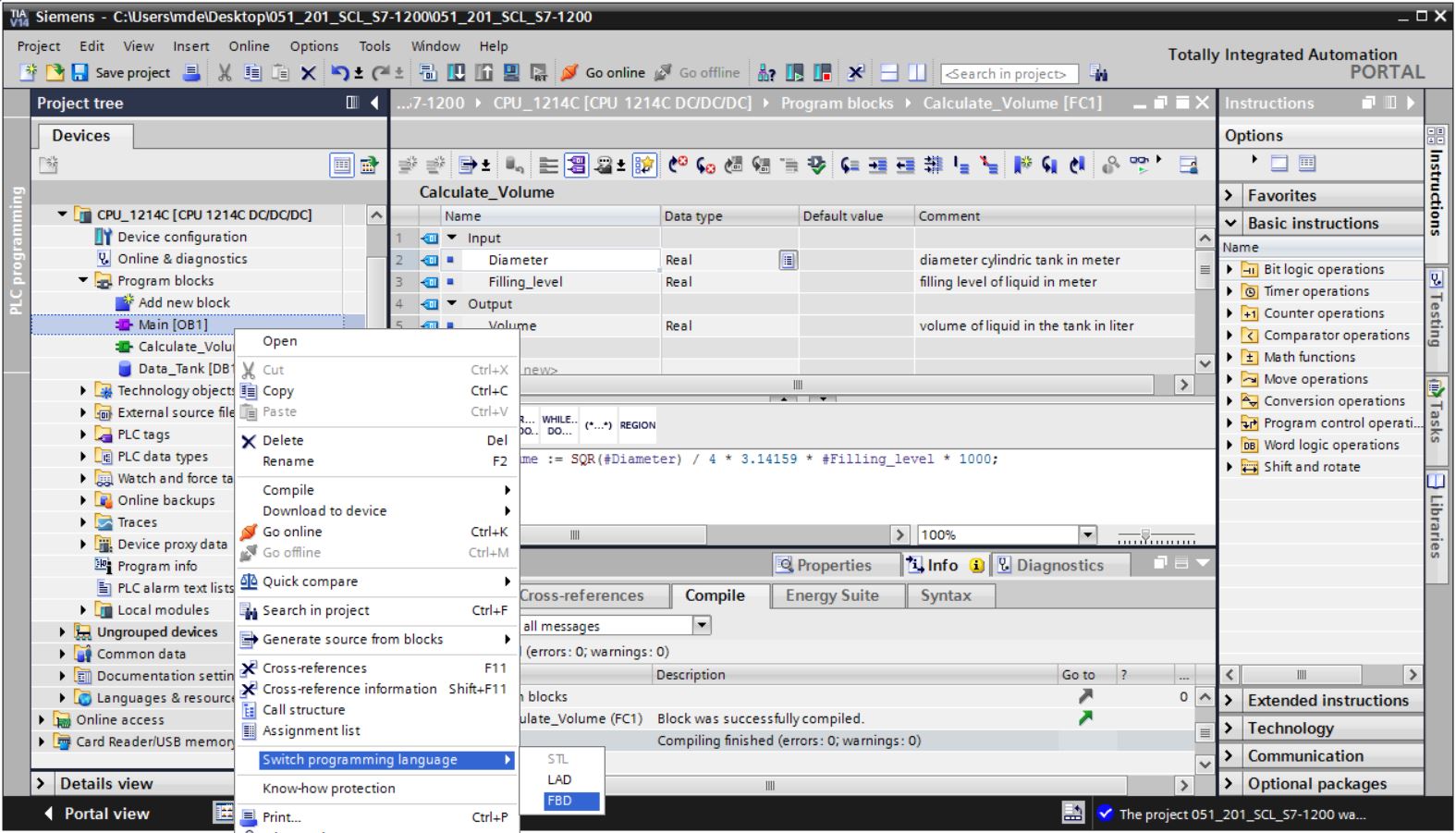




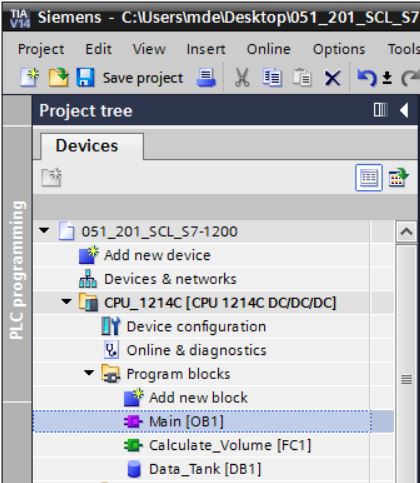
## “Main [OB1]” 오거나이제이션 블록 프로그래밍

* 오거나이제이션 블록 “Main [OB1]”을 프로그래밍하기 전에 펑션 블록 다이어그램(FBD)으로 프로그래밍 언어를 전환해야 합니다. 이를 위해 먼저, “프로그램 블록” 폴더에서 “Main [OB1]”를 클릭합니다.

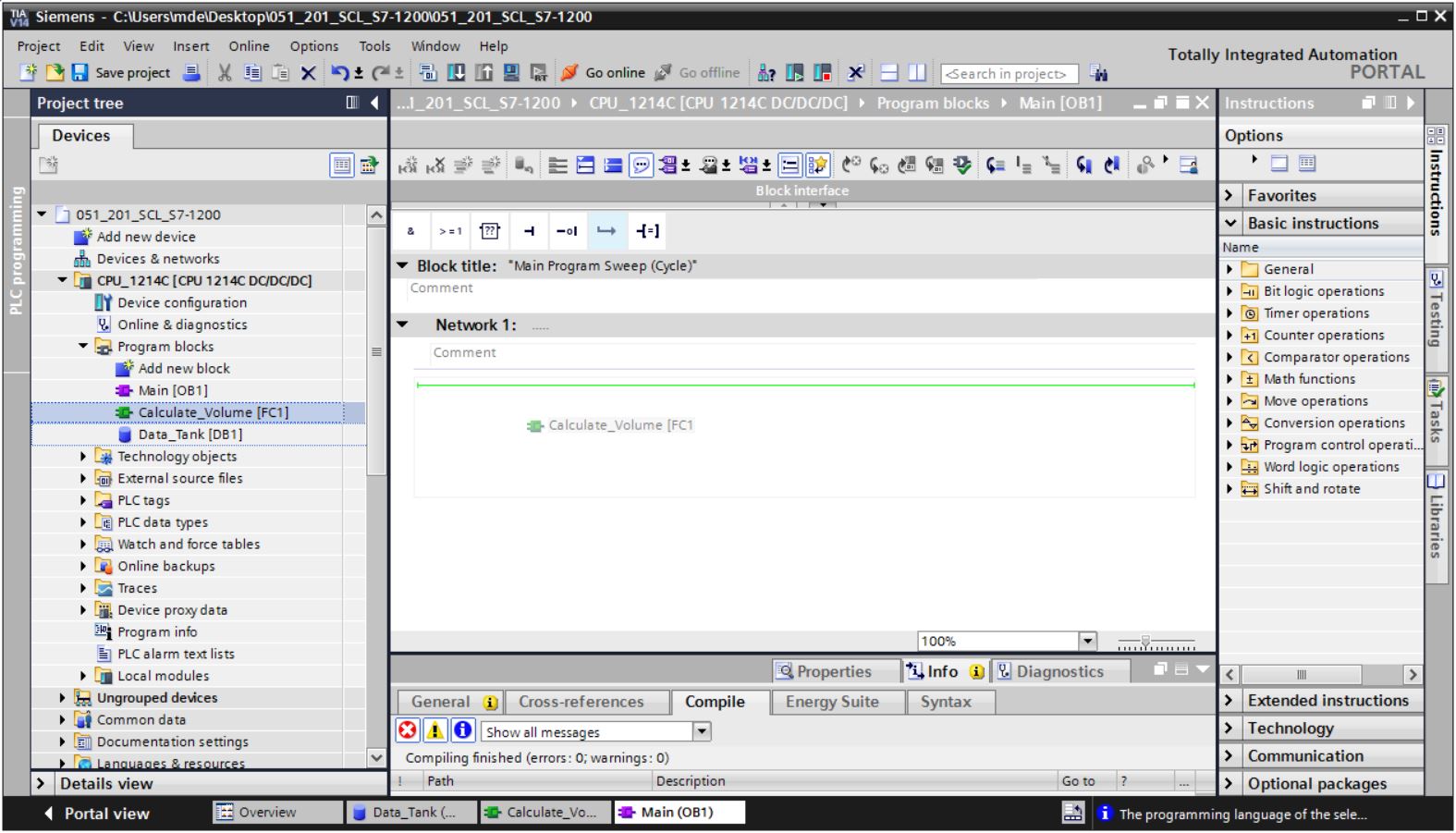
(→ CPU\_1516F[CPU 1516F-3 PN/DP] → Program blocks → Main [OB1] → Switch programming language → FBD)

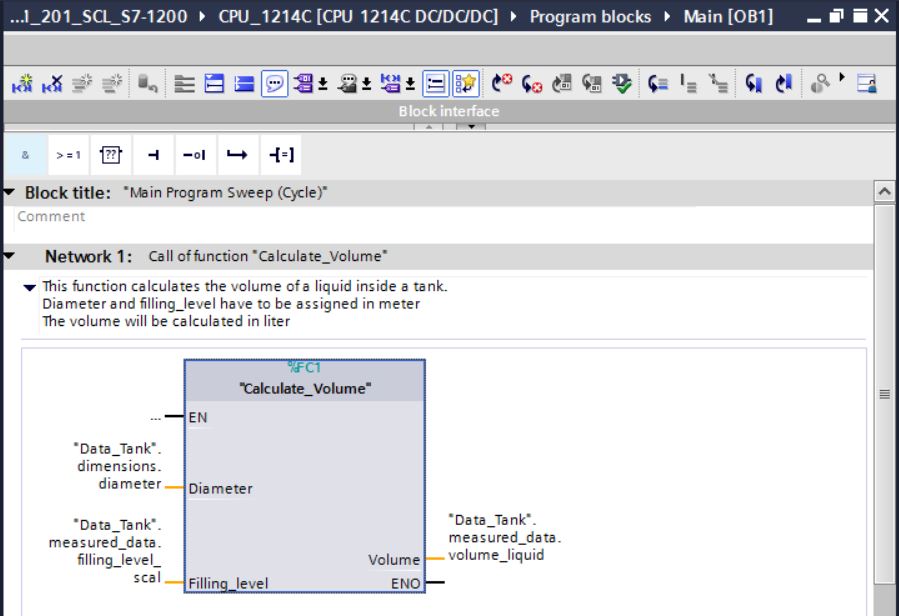


* 이제 “Main [OB1]” 오거나이제이션 블록을 두 번 클릭하여 엽니다.



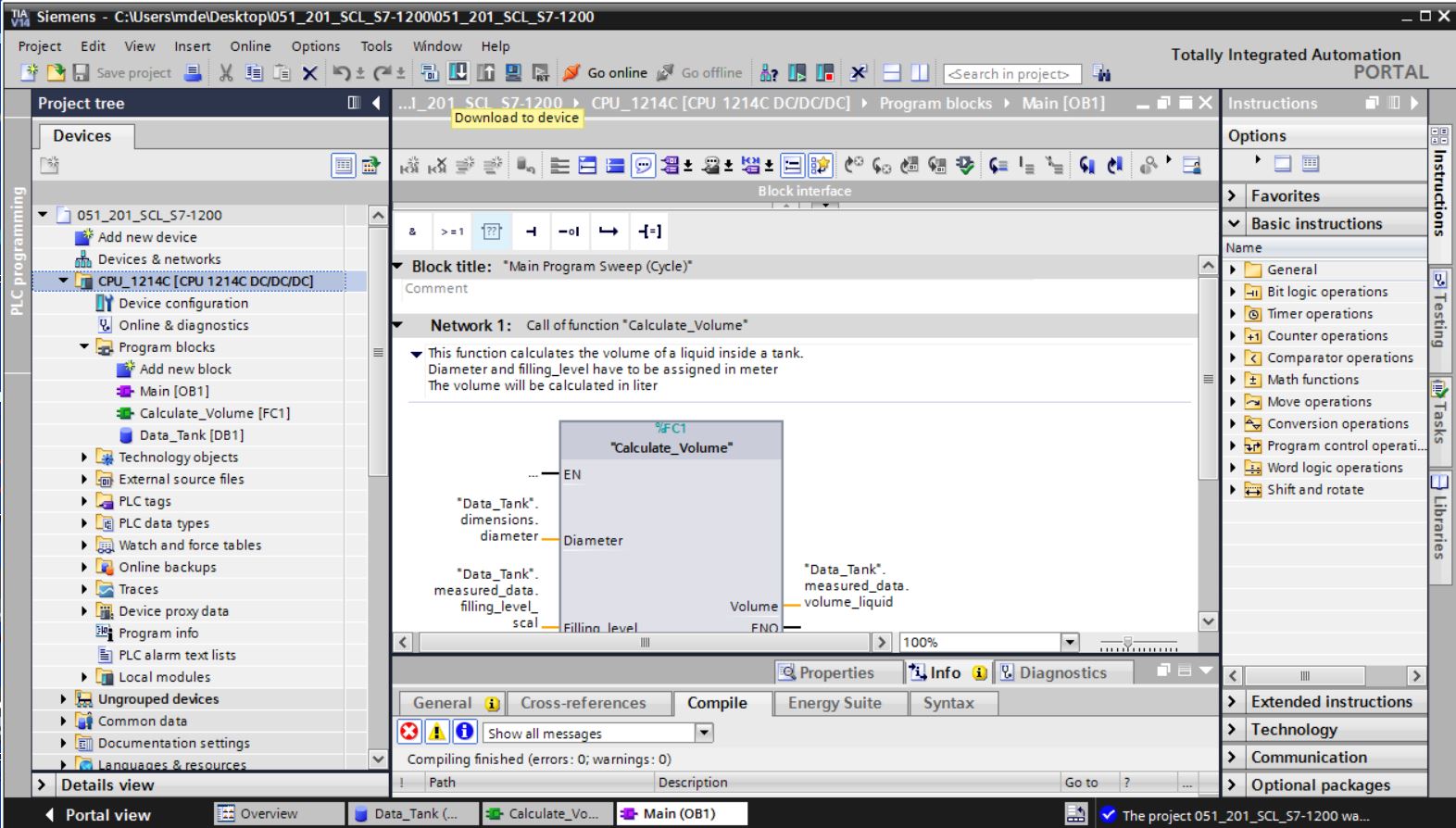
* 첫 번째 네트워크에서 “Calculate\_Volume” 펑션을 호출합니다. 네트워크 제목, 설명을 할당하고 파라미터를 연결합니다. 그런 다음 프로젝트를 저장합니다.   
  (→ “Calculate\_Volume” 호출 → 네트워크 제목 할당 → 네트워크 설명 작성 → 파라미터 연결 → )



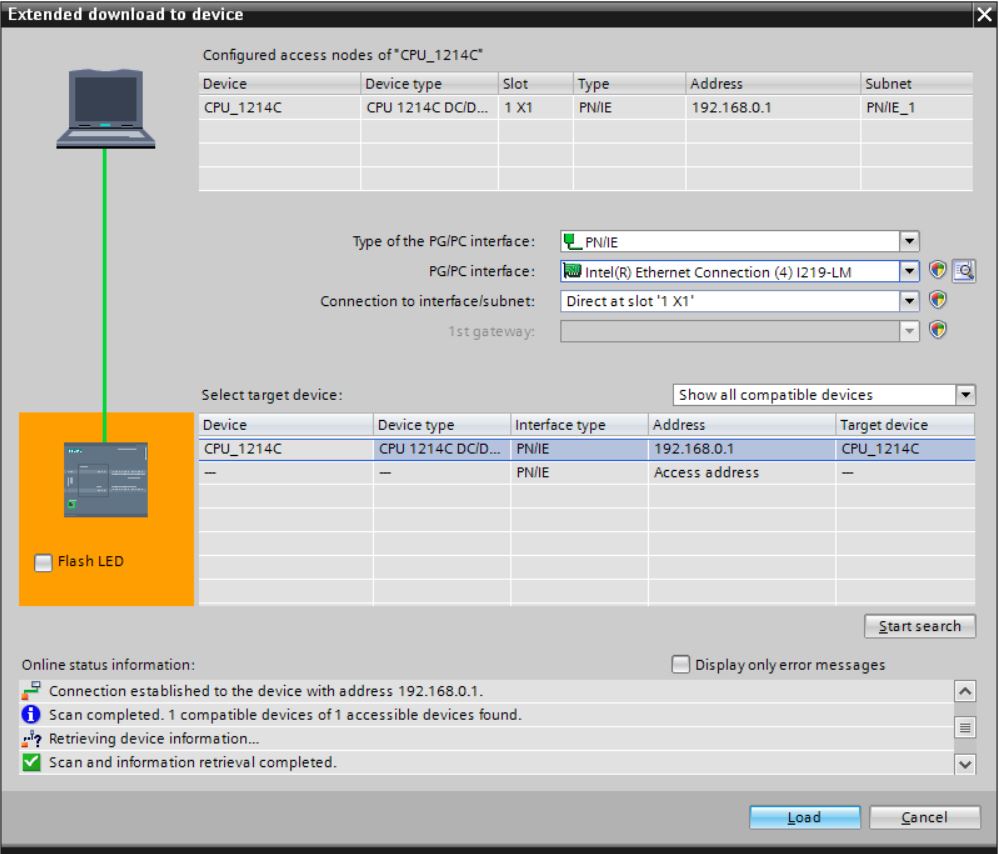


## 프로그램 컴파일 및 다운로드

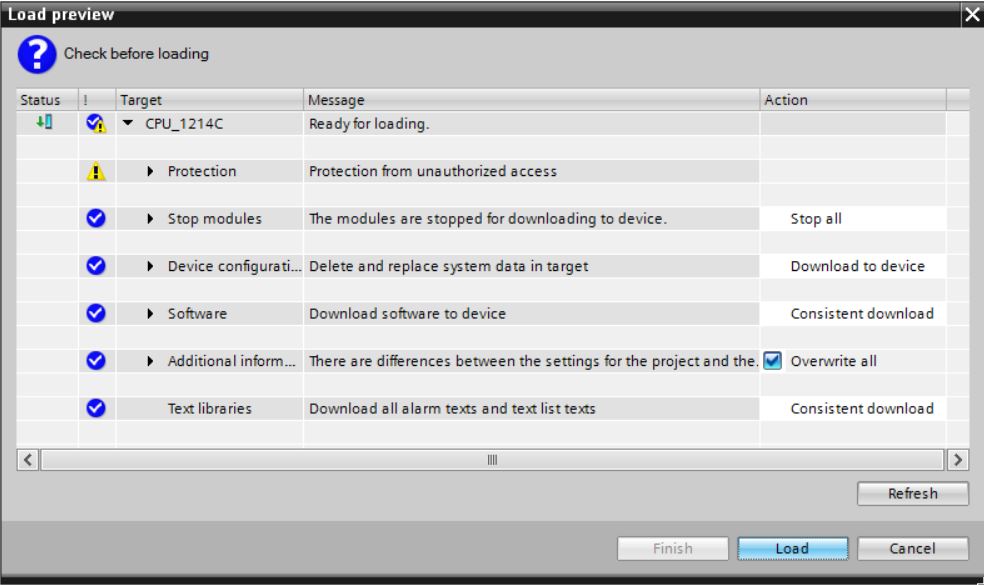
* “프로그램 폴더” 폴더를 클릭하고 전체 프로그램을 컴파일합니다. 성공적으로 컴파일한 후 프로젝트를 PLC로 다운로드합니다. ( →  →  )



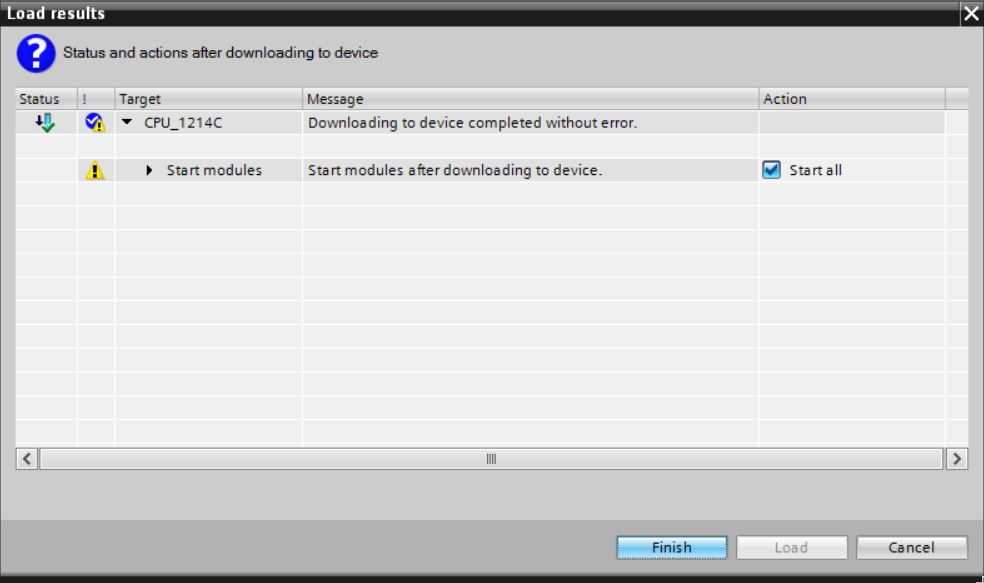
* PG/PC interface 선택 → subnet 선택 → start search → Load



* 필요한 경우 선택 → Load

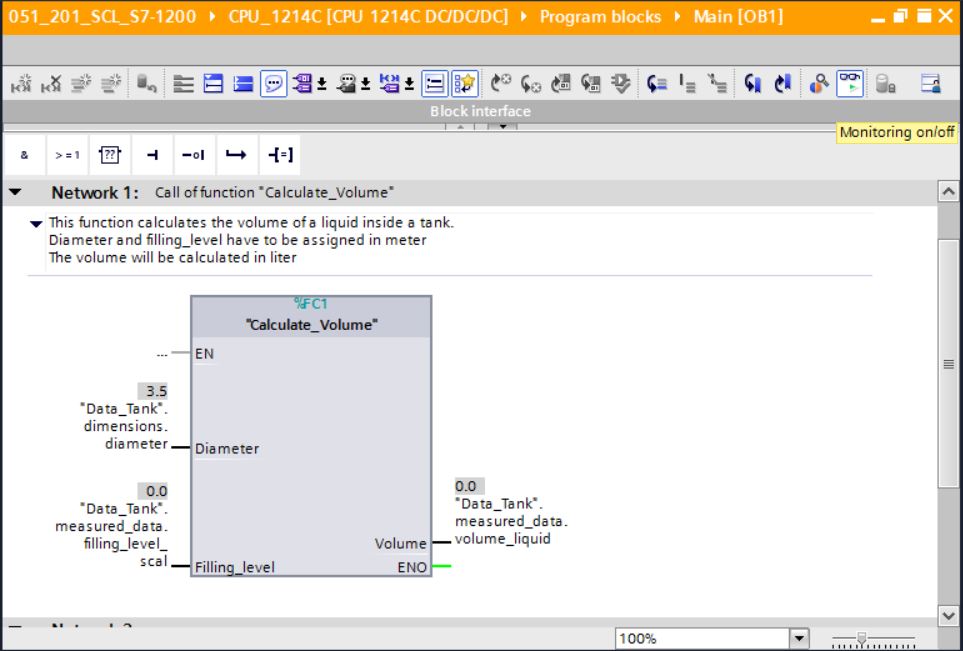


* Finish

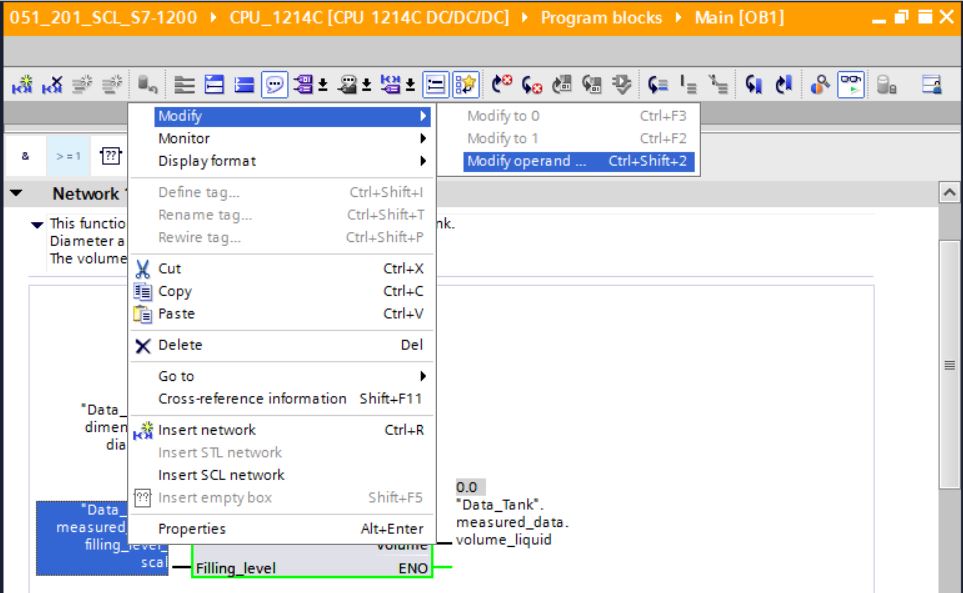


## 오거나이제이션 블록 모니터링 및 테스트

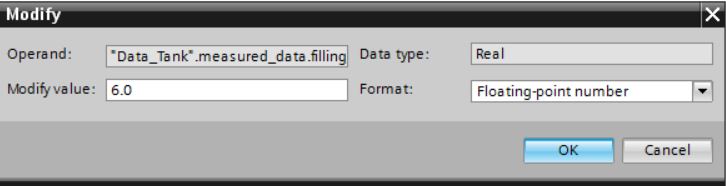
* 열려 있는 OB1에서 D:\00_DATA\SIEMENS\Unterlagen\08_Ausbildungsunterlage_TIA-Portal_R1502_dt\032-100 FC-Programmierung\pics\055b.jpg 아이콘을 클릭하여 블록을 모니터링합니다.



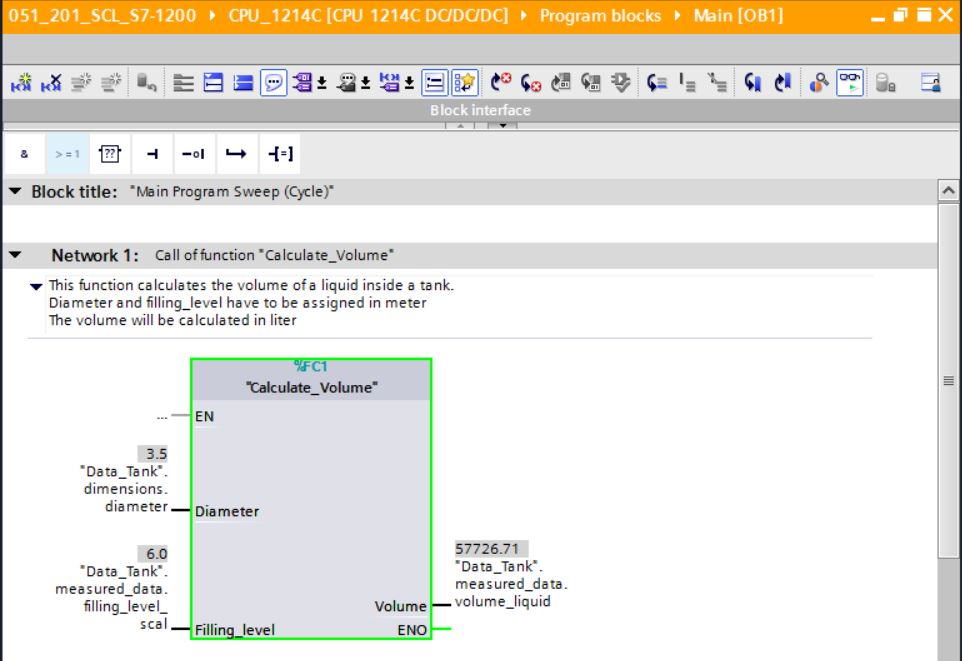
* 데이터 블록에서 “Filling\_level\_scal” 태그에 값을 작성하여 프로그램을 테스트합니다.  
  (→ “Filling\_level\_scal”을 마우스 오른쪽 버튼으로 클릭 → “Modify” 메뉴 → Modify operand...)



* 값 6.0 입력 → OK



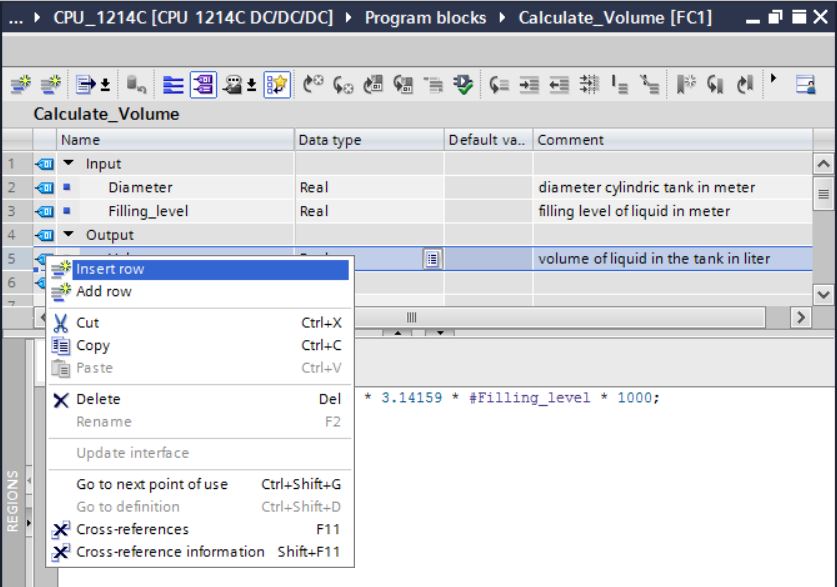
* 결과가 올바른지 검사합니다.



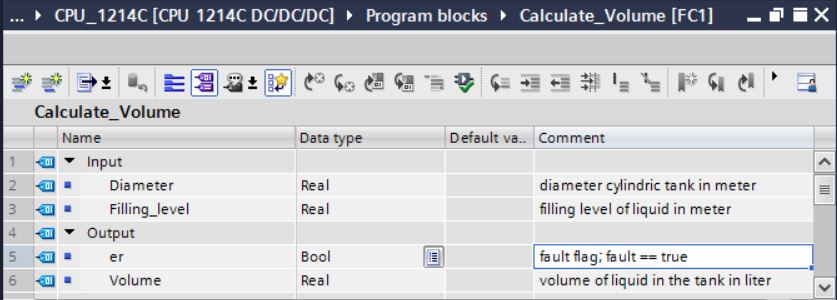
## “Calculate\_Volume” 펑션 확장

* “Calculate\_Volume” 펑션을 열고 인터페이스에서 행을 마우스 오른쪽 버튼으로 클릭하여 출력 파라미터에 행을 삽입합니다.

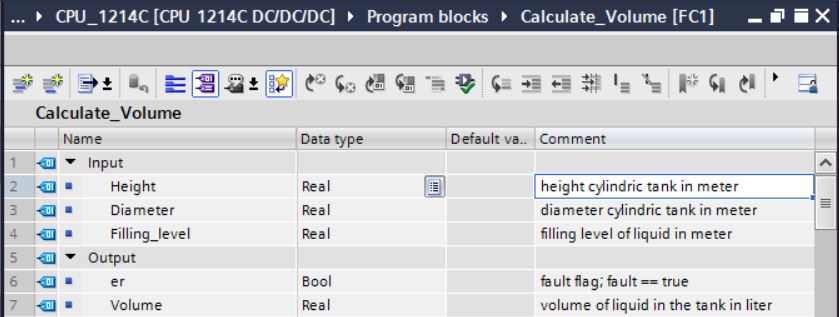
(→ “Calculate\_Volume” 열기 → 행 5를 마우스 오른쪽 버튼으로 클릭 → Insert row)



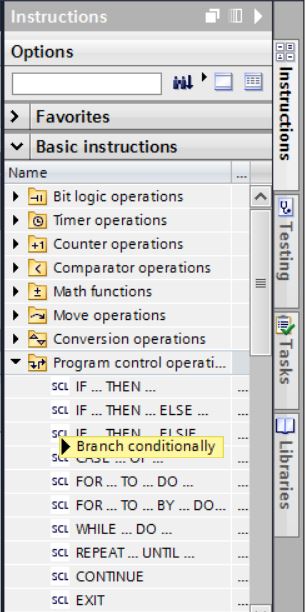
* 데이터 유형 BOOL 및 설명과 함께 파라미터 “er”을 입력합니다.



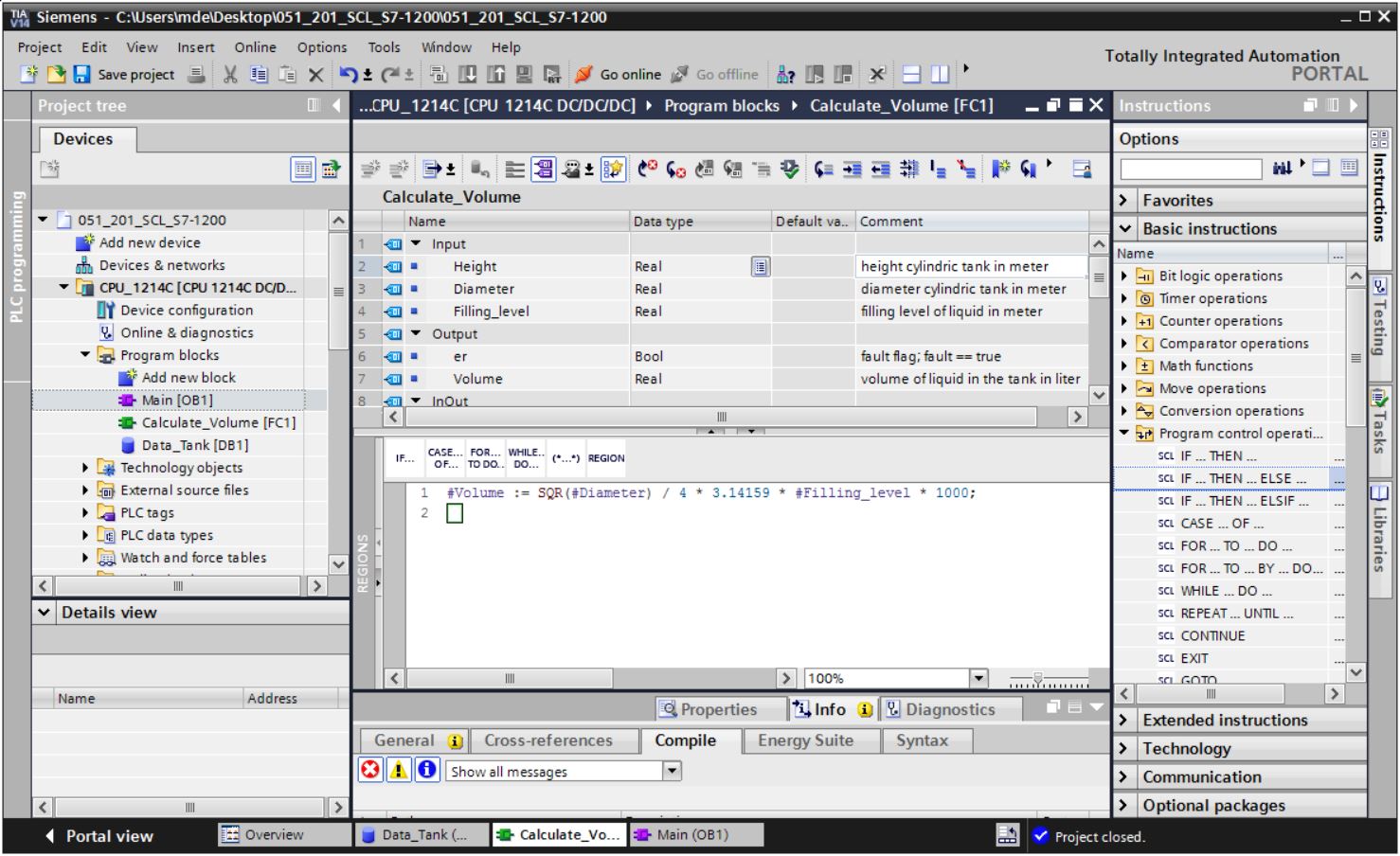
* 같은 단계를 따라 데이터 유형 Real 및 설명과 함께 “Height” 태그를 추가합니다.

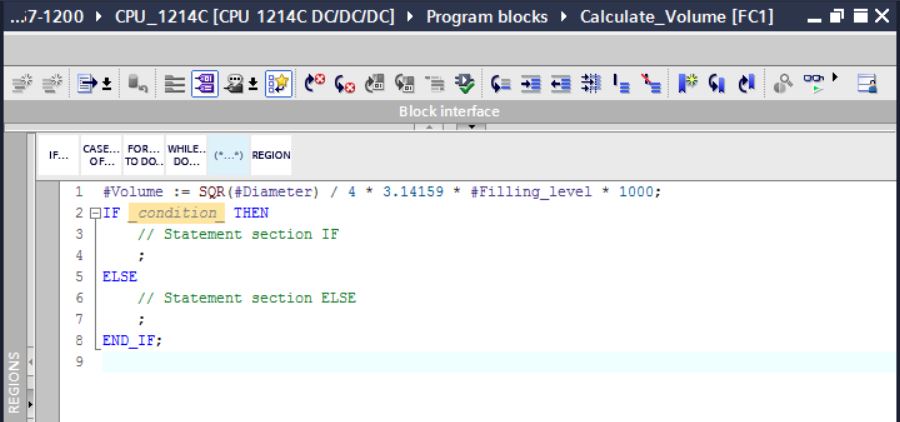


* 그런 다음 기본 명령의 “프로그램 제어 연산”에서 “IF…THEN…ELSE” 제어문으로 이동합니다.   
  (→ 명령 → 기본 명령 → 프로그램 제어 연산 → “IF...THEN…ELSE”)

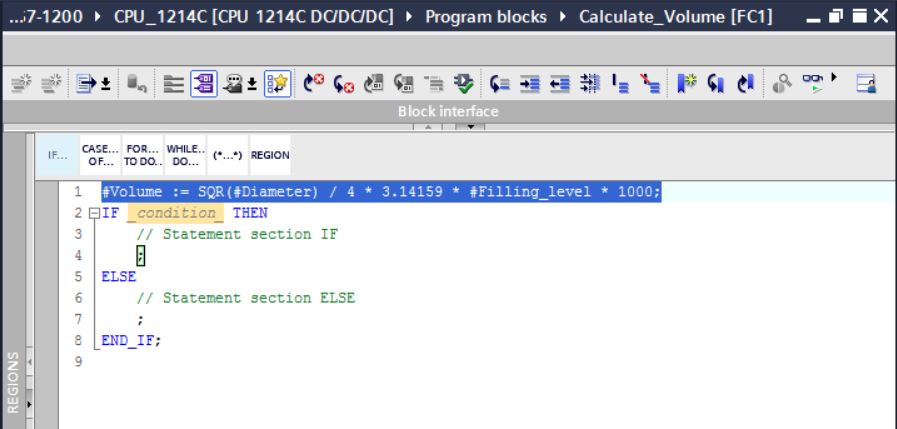


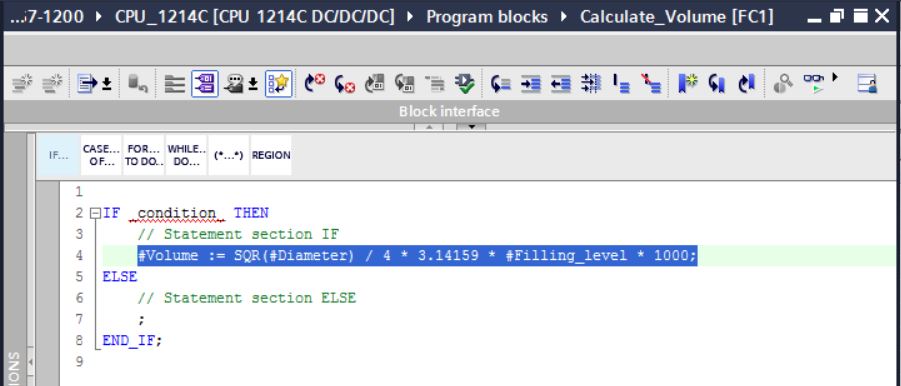
* 그런 다음 “IF...THEN...ELSE” 제어문을 프로그램의 두 번째 행으로 끕니다.   
  (→ “IF…THEN…ELSE” → 끌어다 놓기)



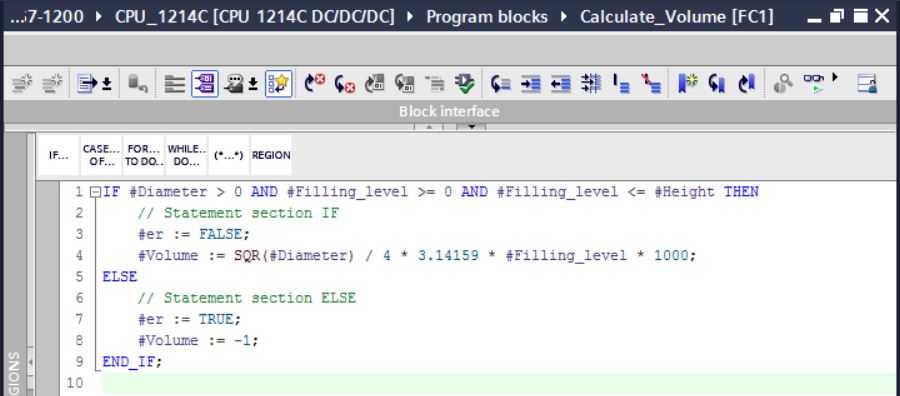


* 수식을 강조 표시하고 ELSE 앞에 있는 세미콜론을 끕니다. (→ 강조 표시 → 끌어다 놓기)



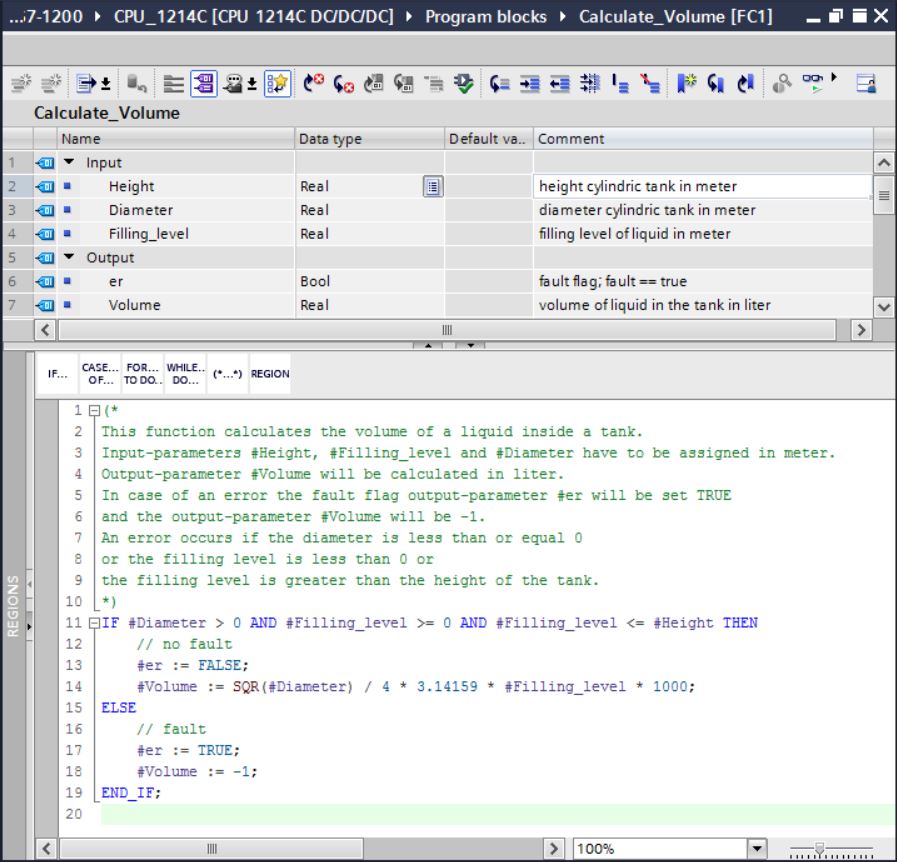


* 펑션을 완료하고 컴파일하여 프로그램을 검사합니다. (→ 프로그램 완료 → )



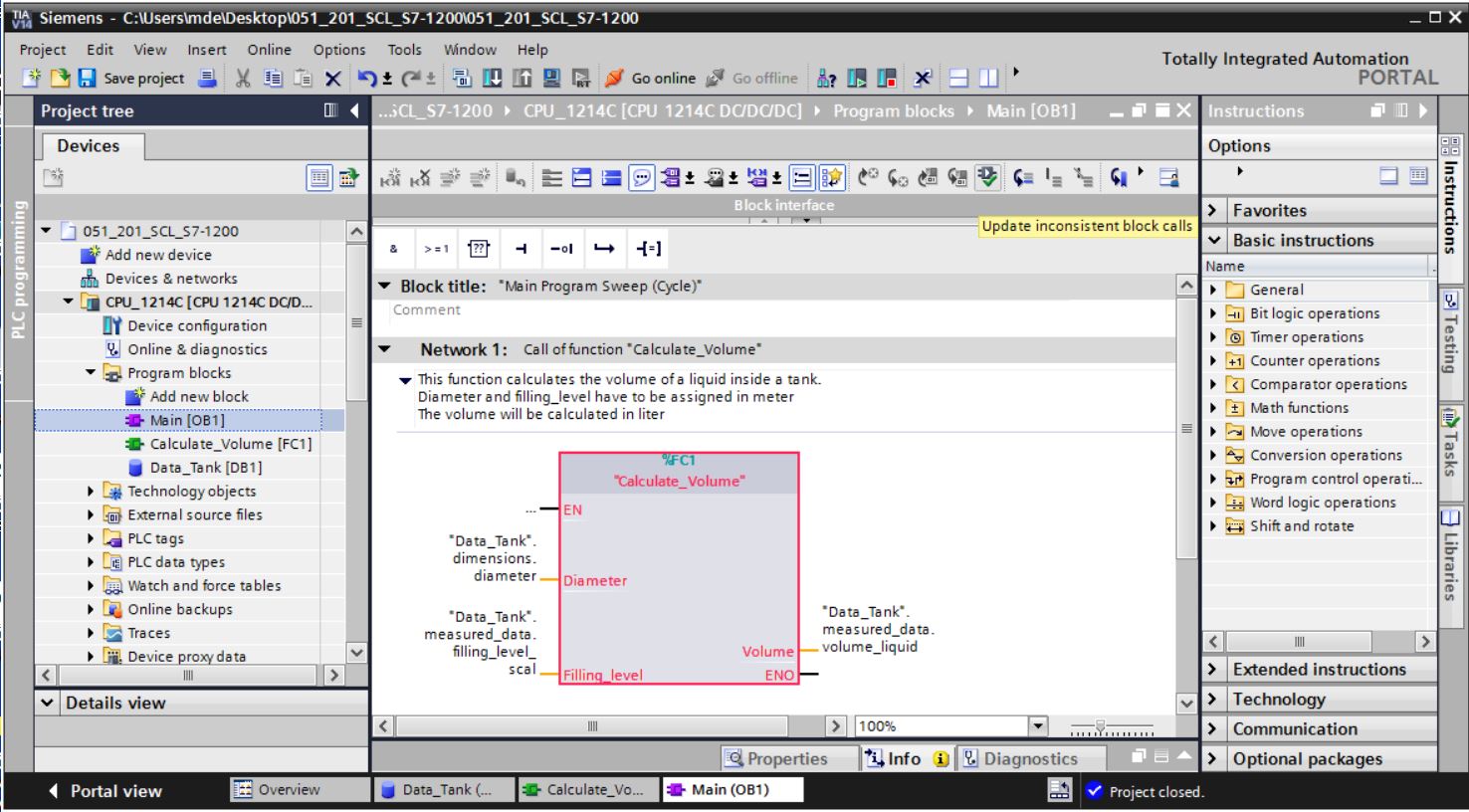
* 블록 설명으로 “(\*\*)”를 사용해 추가하고 행 설명으로 “//”를 사용해 추가합니다.   
  이제 설명을 포함한 프로그램을 완료할 수 있습니다.

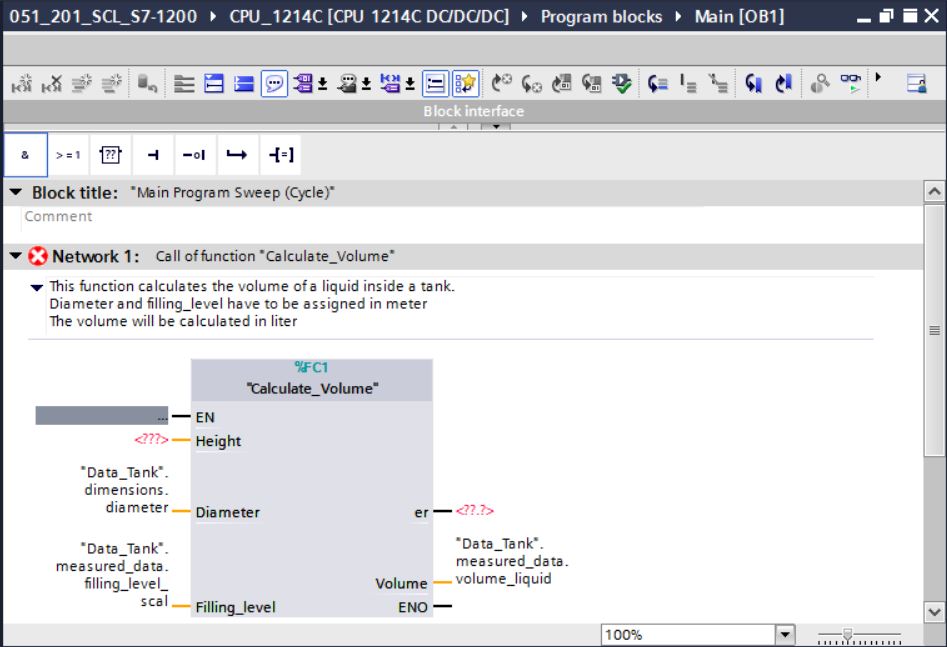
(→ 행 1로 시작하는 블록 설명 추가 → 행 12 몇 16에서 행 설명 추가)



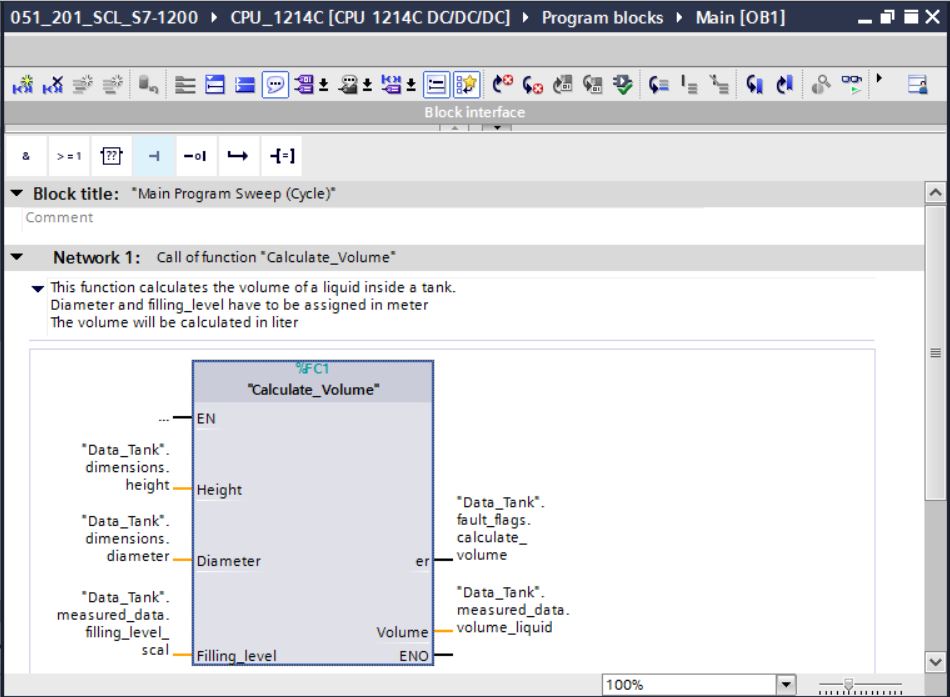
## 오거나이제이션 블록 사용자 정의

* OB1을 열고 을 클릭하여 일치하지 않는 블록 호출을 업데이트합니다.   
  (→ OB1 열기 → )



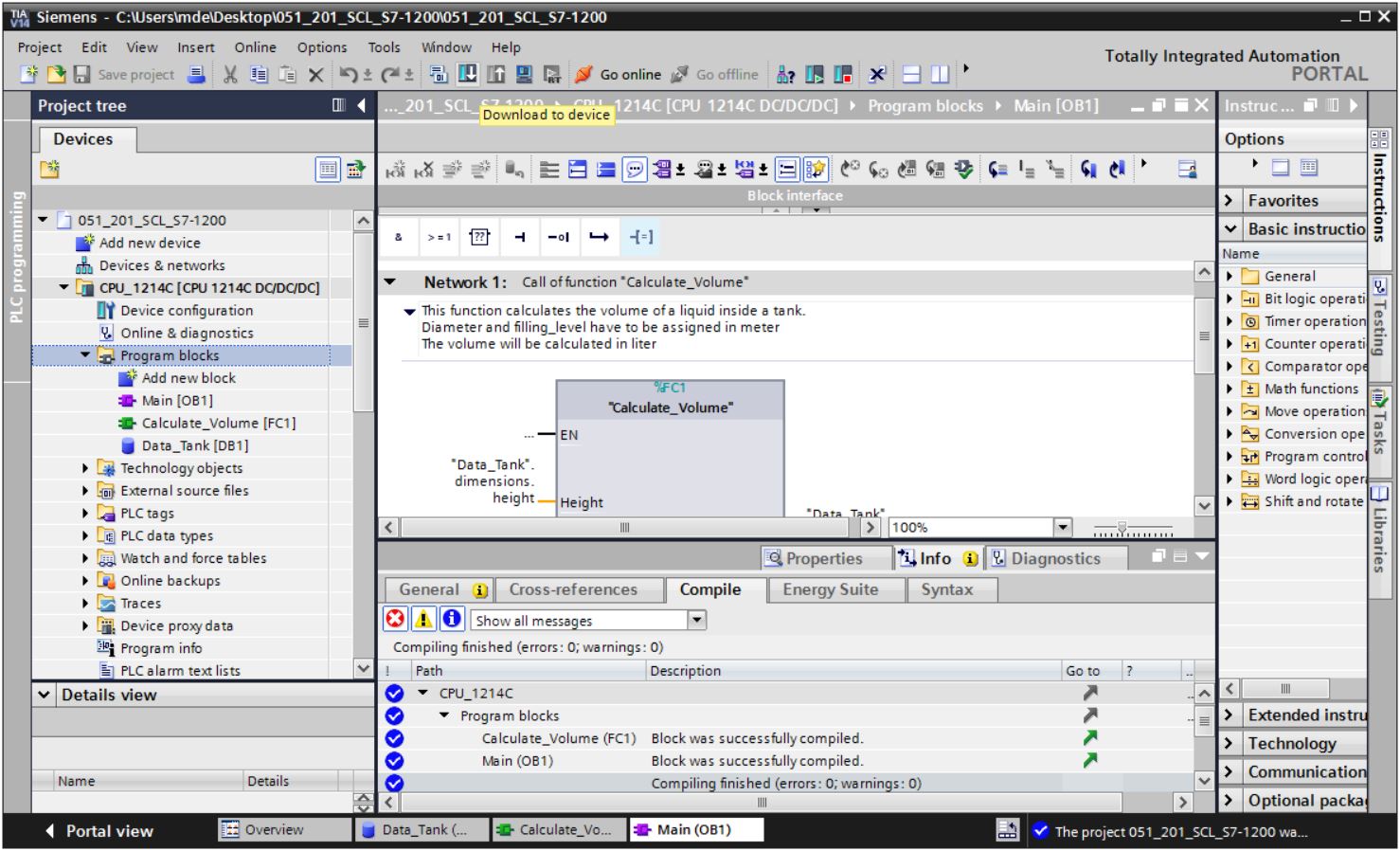


* 이렇게 하려면 파라미터 “er” 및 “Height”를 추가합니다.



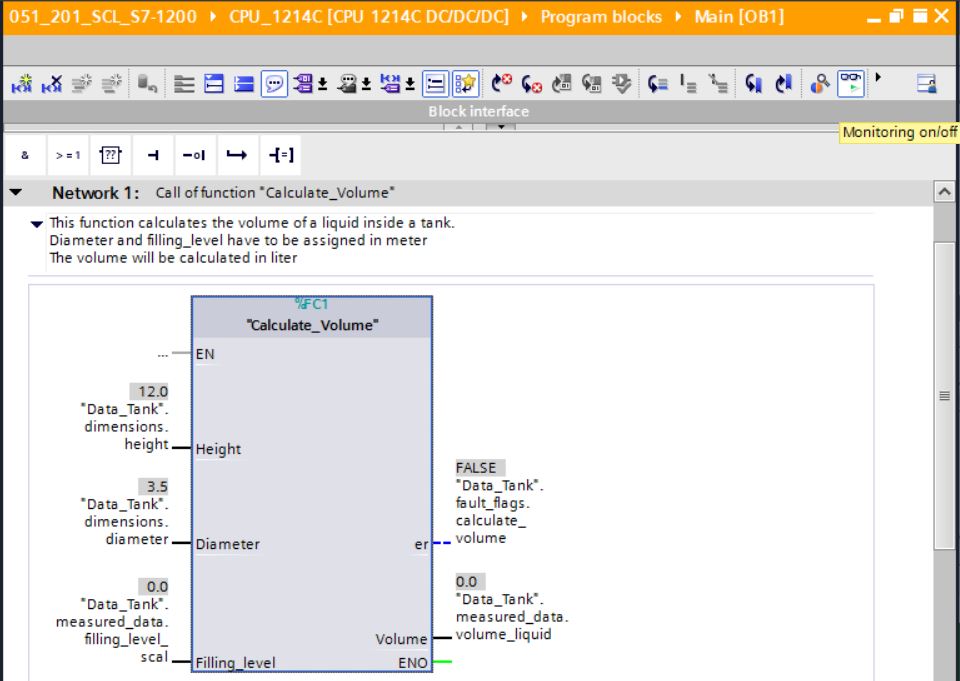
## 프로그램 컴파일 및 다운로드

* “프로그램 블록” 폴더를 클릭하고 전체 프로그램을 컴파일합니다. 성공적으로 컴파일한 후 프로젝트를 PLC로 다운로드합니다. 그런 다음 프로젝트를 저장합니다.   
  ( → 프로그램 블록→  →  →  )

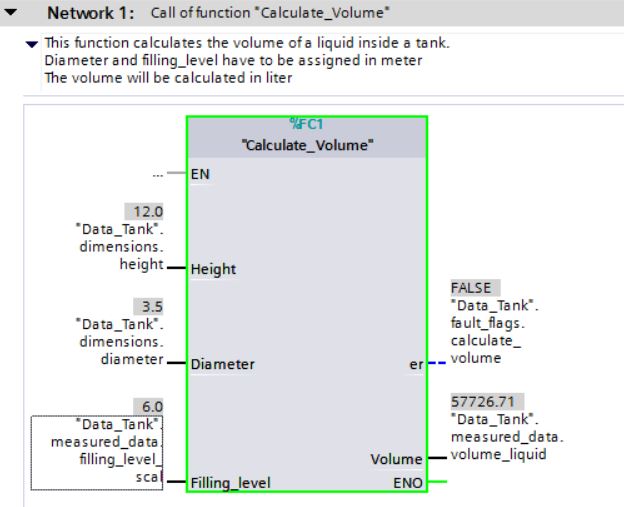


## 오거나이제이션 블록 모니터링 및 테스트

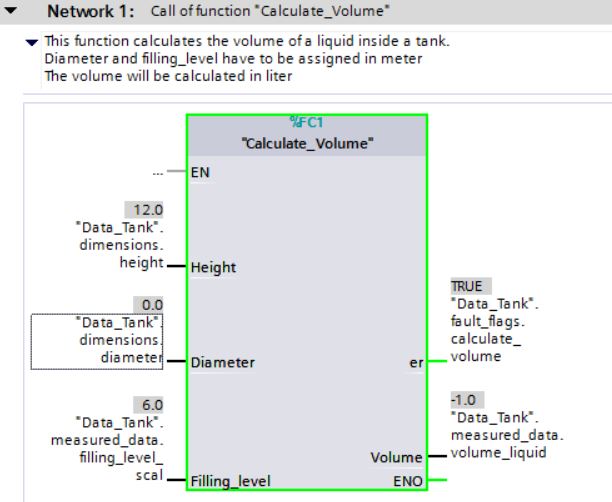
* 열려 있는 OB1에서 D:\00_DATA\SIEMENS\Unterlagen\08_Ausbildungsunterlage_TIA-Portal_R1502_dt\032-100 FC-Programmierung\pics\055b.jpg 아이콘을 클릭하여 블록을 모니터링 합니다.



* 데이터 블록에서 “Filling\_level\_scal” 태그에 값을 작성하여 프로그램을 테스트합니다.  
  (→ “Filling\_level\_scal”을 마우스 오른쪽 버튼으로 클릭 → “수정” 메뉴 → 오퍼랜드 수정 → 값 6.0 입력 → 확인 → 검사)

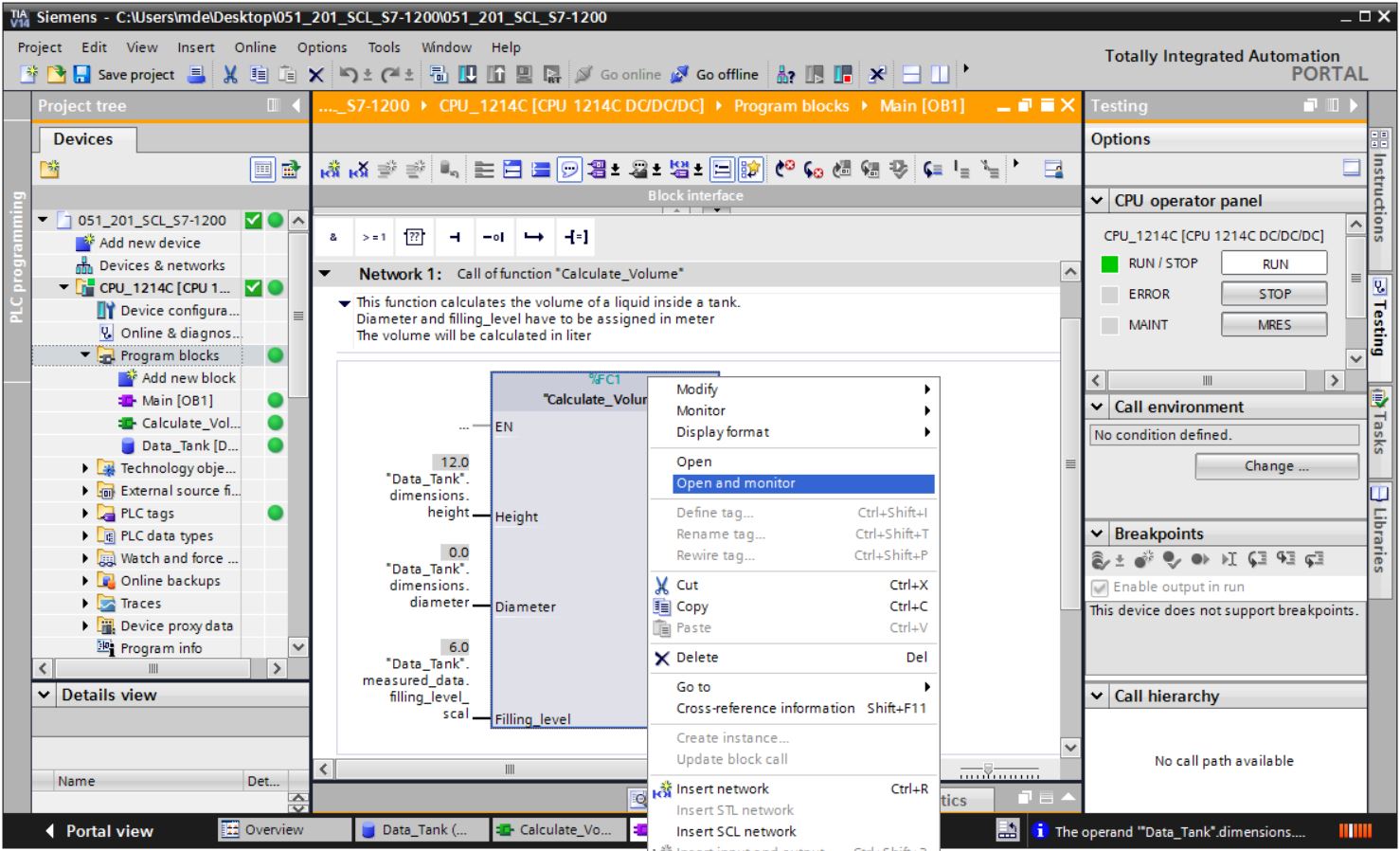


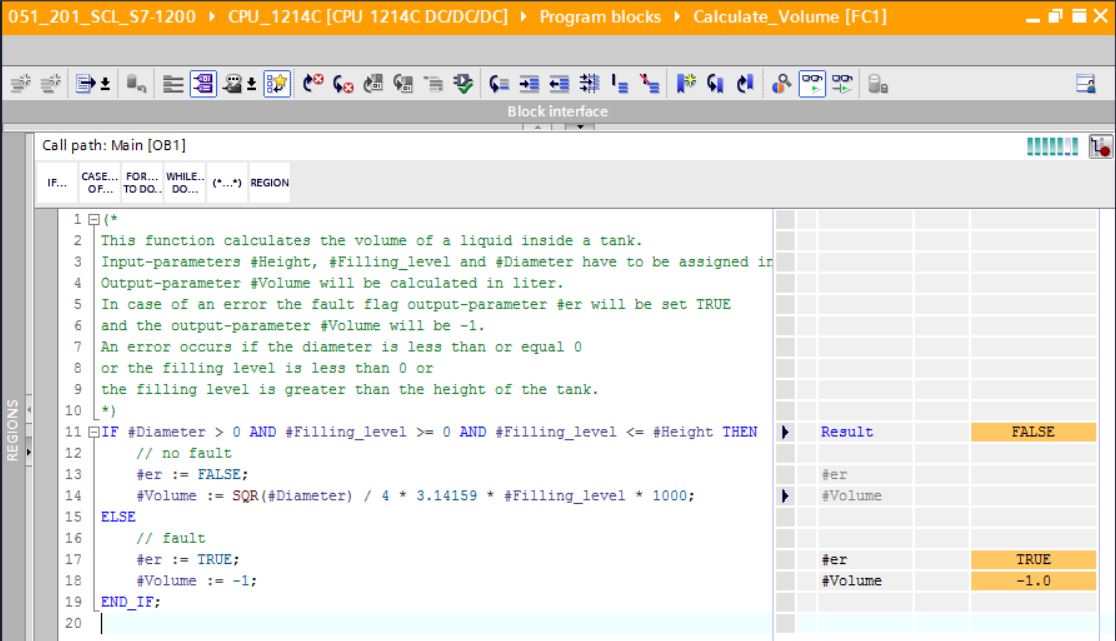
* 이제 직경을 0으로 설정하여 오류가 출력되는지 테스트합니다. (→ “Diameter”를 마우스 오른쪽 버튼으로 클릭 → “수정” 메뉴 → 오퍼랜드 수정 → 값 0.0 입력 → 확인 → 검사)



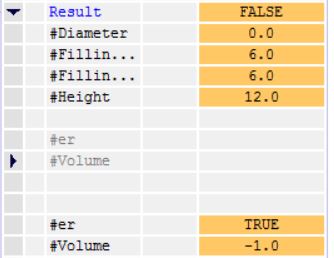
## “Calculate\_Volume” 펑션 모니터링 및 테스트

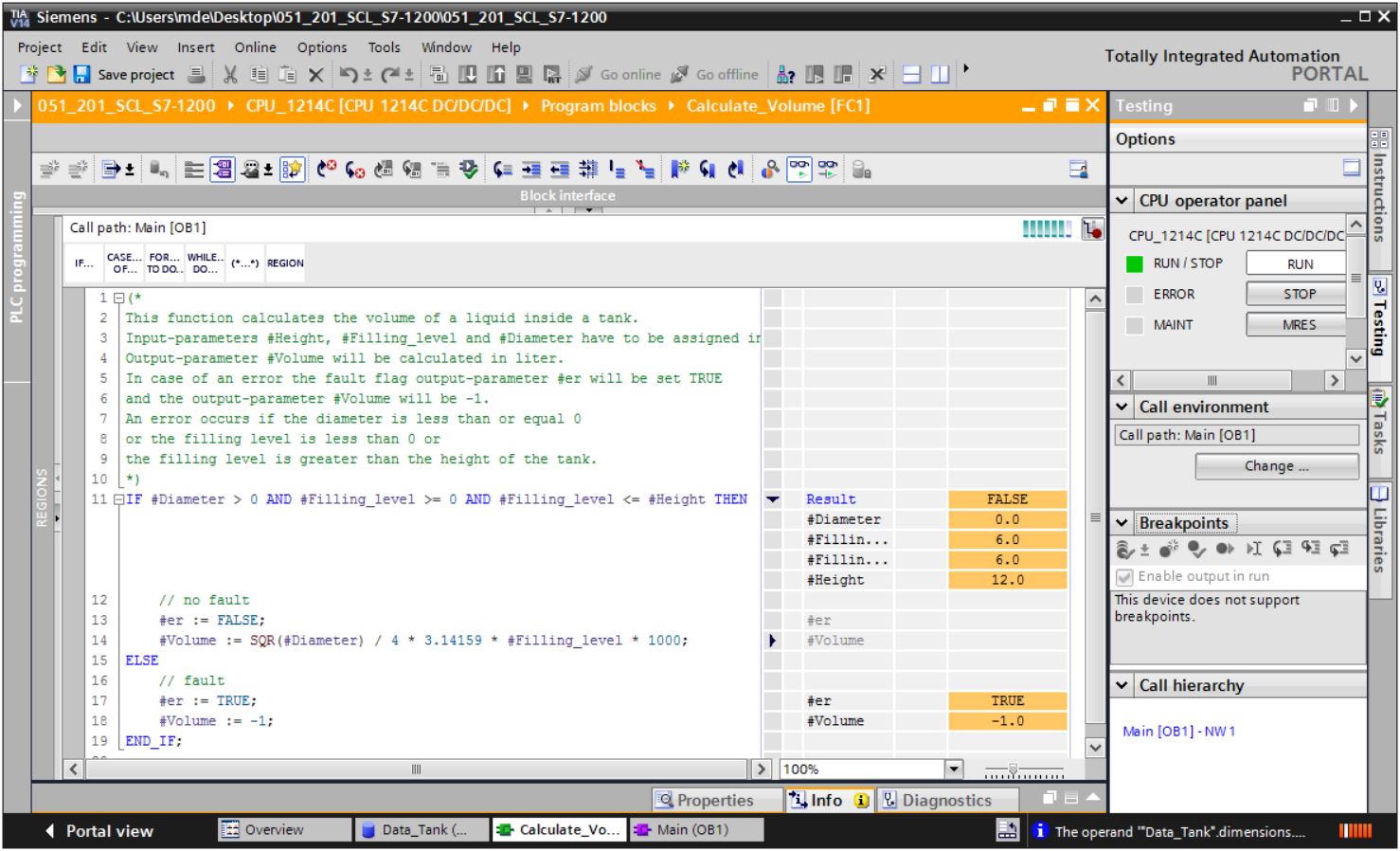
* 마지막으로 펑션에서 마우스 오른쪽 버튼을 클릭하고 “Open and monitor” 메뉴 명령을 선택하여 “Calculate\_Volume” 펑션을 연 다음 모니터링 합니다.   
  (→ 펑션을 마우스 오른쪽 버튼으로 클릭 → Open and monitor)



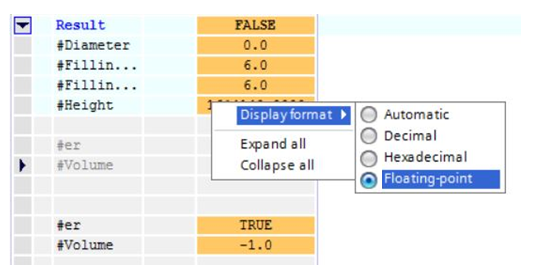


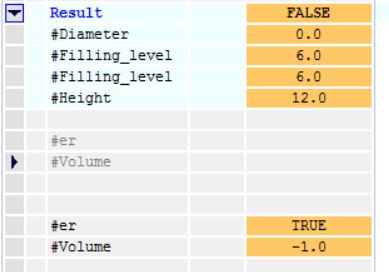
* 검은색 화살표 를 클릭하여 IF 쿼리의 개별 태그 값을 표시할 수 있습니다.   
  ( → )



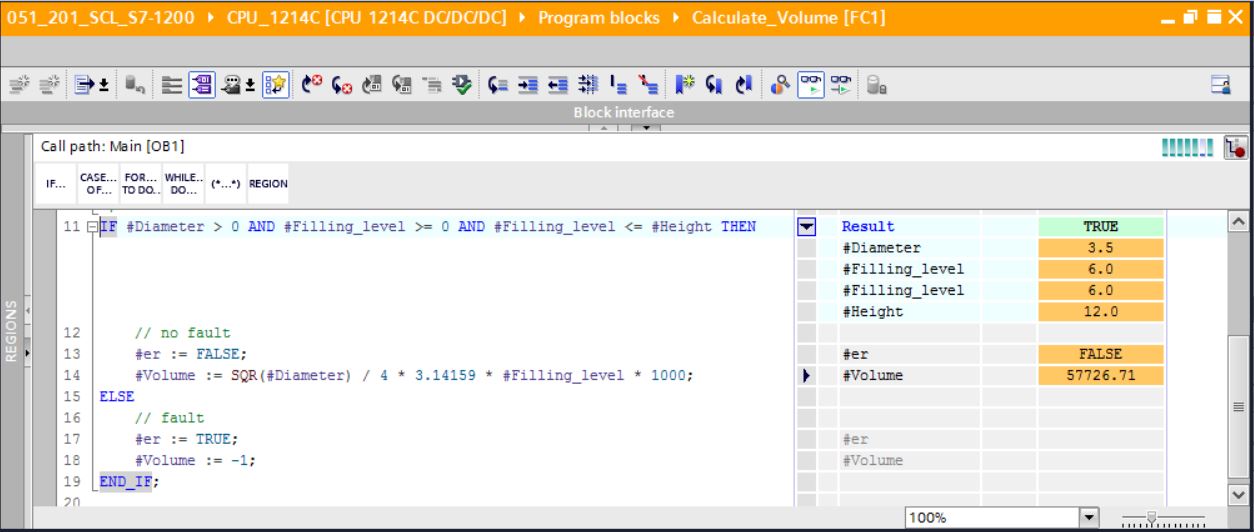


* 태그를 마우스 오른쪽 버튼으로 클릭하여 표시 형식을 조정합니다.   
  (→ 태그를 마우스 오른쪽 버튼으로 클릭 → Display format → Floating-point)





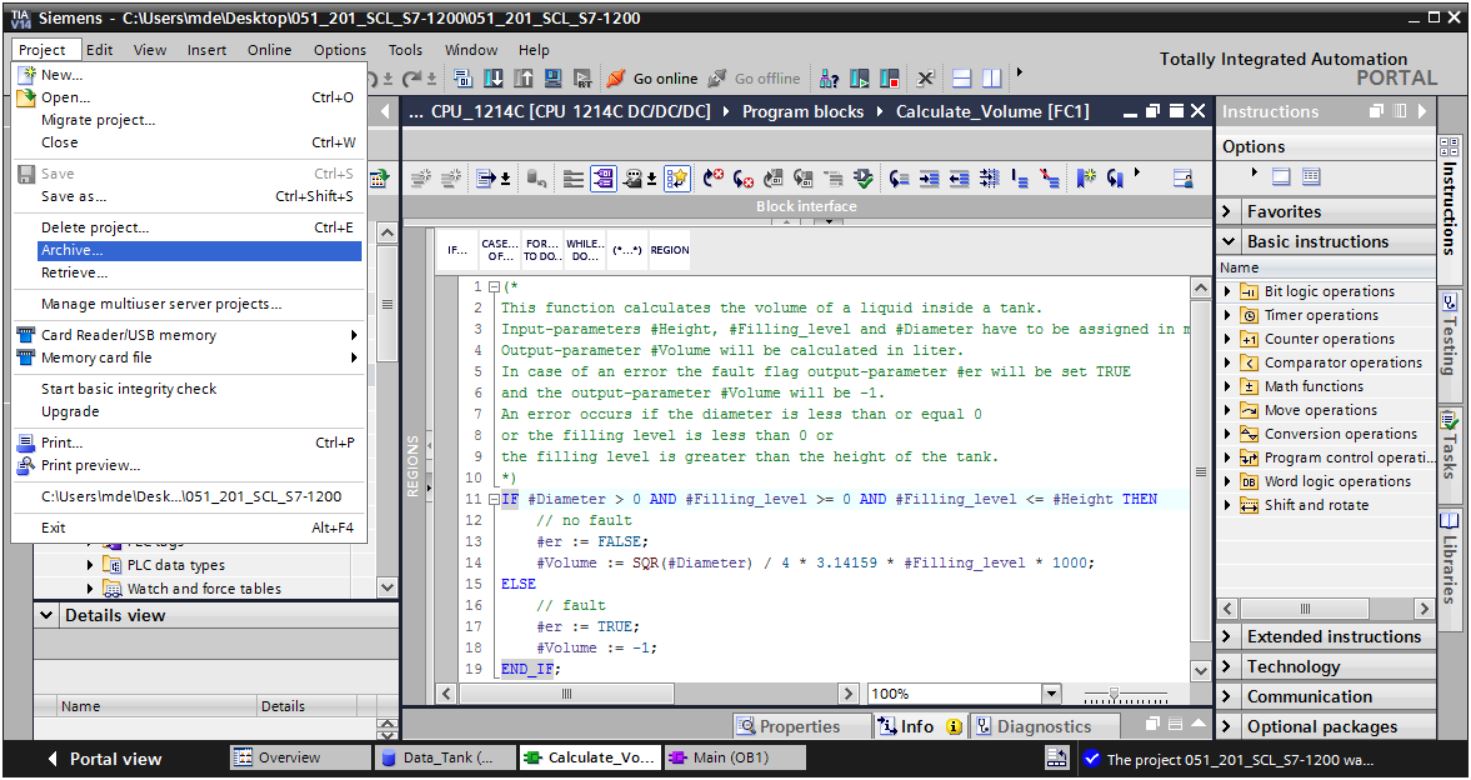
* 이제 OB1에서 직경을 다시 3.5m로 수정하여 IF 분기의 다른 분기를 테스트합니다.   
  (→ OB1 열기 → 직경을 3.5로 수정 → 펑션 열기 및 모니터링)



## 프로젝트 아카이브

* 마지막으로, 이제 전체 프로젝트를 아카이브 하려고 합니다. 메뉴에서 → ‘Project’ → ‘Archive…’를 선택합니다. 프로젝트를 아카이브 하고자 하는 폴더를 열고 “TIA Portal 프로젝트 아카이브” 파일 유형으로 이를 저장합니다.

( → Project → Archive → TIA Portal 프로젝트 아카이브 → SCE\_EN\_052-201 Startup SCL\_S7-1500… → Save )



# 체크리스트

|  |  |  |
| --- | --- | --- |
| No. | 설명 | 검사 완료 |
| 1 | 오류 메시지 없이 성공적으로 컴파일 |  |
| 2 | 오류 메시지 없이 성공적으로 다운로드 |  |
| 3 | 오퍼랜드 수정(직경 = 0.0)  결과 태그 볼륨 = -1  결과 태그 “er” = TRUE |  |
| 4 | 오퍼랜드 수정(직경 = 3.5 및 Filling\_level\_scal = 0)  결과 볼륨 = 0  결과 태그 “er” = FALSE |  |
| 5 | 오퍼랜드 수정(Filling\_level\_scal= 6.0)  결과 볼륨 = 57726.72  결과 태그 “er” = FALSE |  |
| 6 | 오퍼랜드 수정(Filling\_level\_scal= 12.0)  결과 볼륨 = 115453.4  결과 태그 “er” = FALSE |  |
| 7 | 오퍼랜드 수정(Filling\_level\_scal= 14.0)  결과 볼륨 = -1  결과 태그 “er” = TRUE |  |
| 8 | 프로젝트가 성공적으로 아카이브됨 |  |

# 연습

## 과제 설명 – 연습

이 연습에서는 “Scaling” 펑션을 프로그래밍합니다. 프로그램은 일반적으로 모든 양의 아날로그 값에 적용할 수 있습니다. 예제 태스크 “Tank”에서 아날로그 센서로 filling level판독하고 이 펑션으로 스케일된 값을 데이터 블록에 저장합니다.

오류가 발생한 경우 블록이 오류 플래그 “er”을 TRUE로 설정하므로 결과적으로 파라미터 “Analog\_scal”이 0으로 설정됩니다. “mx” 파라미터가 “mn”보다 작거나 같으면 오류가 발생합니다.

펑션에는 다음 파라미터가 포함되어야 합니다.

|  |  |  |
| --- | --- | --- |
| 입력 | 데이터 유형 | 코멘트 |
| Analog\_per | INT | 0..27648 사이 IO의 아날로그 값 |
| mx | REAL | 새 확장의 최대값 |
| mn | REAL | 새 확장의 최소값 |
| 출력 |  |  |
| er | BOOL | 오류 플래그, 오류 없음 = 0, 오류 = 1 |
| Analog\_scal | REAL | mn..mx 사이 확장된 아날로그 값  오류 발생 시 = 0 |

과제를 해결하는 데 다음 식을 사용합니다.



이 과제에는 아날로그 신호가 필요합니다. 이 과제에 사용되는 오퍼랜드를 PLC 태그 테이블에 입력해야 합니다.

|  |  |  |  |
| --- | --- | --- | --- |
| 이름 | 데이터 유형 | 주소 | 코멘트 |
| B1 | INT | %IW64 | 0..27648 사이 Filling level |

## 계획 수립

이제 이 과제를 스스로 해결합니다.

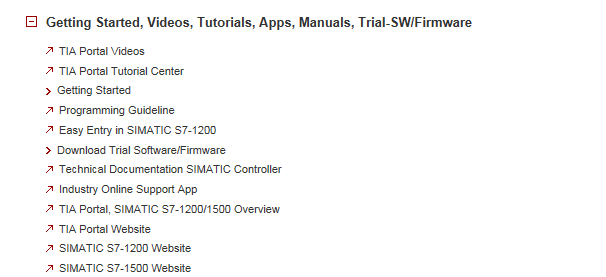
## 체크리스트 – 연습

|  |  |  |
| --- | --- | --- |
| No. | 설명 | 검사 완료 |
| 1 | PLC 태그 테이블에 오퍼랜드 추가 |  |
| 2 | 펑션 FC: “Scaling” 생성 |  |
| 3 | 인터페이스 정의 |  |
| 4 | 펑션 프로그래밍 |  |
| 5 | OB1의 네트워크 1에 “Scaling” 펑션 추가 |  |
| 6 | 입력 태그 연결 |  |
| 7 | 출력 태그 연결 |  |
| 8 | 오류 메시지 없이 성공적으로 컴파일 |  |
| 9 | 오류 메시지 없이 성공적으로 다운로드 |  |
| 10 | 채움 수준에 대한 아날로그 값이 0으로 설정됨  결과 Filling\_level\_scal = 0  결과 er = FALSE |  |
| 11 | 채움 수준에 대한 아날로그 값이 27648로 설정됨  결과 Filling\_level\_scal = 12.0  결과 er = FALSE |  |
| 12 | 채움 수준에 대한 아날로그 값이 13824로 설정됨  결과 Filling\_level\_scal = 6.0  결과 er = FALSE |  |
| 13 | 오퍼랜드 수정(mx = 0.0)  결과 Filling\_level\_scal = 0  결과 태그 “er” = TRUE |  |
| 14 | 프로젝트가 성공적으로 아카이브됨 |  |

# 추가 정보

초기 및 심화 교육에 방향을 제시하는 도우미로서 예를 들어 시작하기, 동영상, 교재, 앱, 매뉴얼, 프로그래밍 지침, 체험용 소프트웨어/펌웨어와 같은 추가 정보를 아래 링크에서 찾아보실 수 있습니다.   
  
[www.siemens.com/sce/s7-1200](http://www.siemens.com/sce/s7-1200)

**"추가 정보" 미리보기**



추가 정보

Siemens Automation Cooperates with Education  
**siemens.com/sce**

SCE 교육 커리큘럼  
**siemens.com/sce/documents**

SCE 교육 담당자 패키지  
**siemens.com/sce/tp**

SCE 담당 파트너   
**siemens.com/sce/contact**

Digital Enterprise  
**siemens.com/digital-enterprise**

Industrie 4.0   
**siemens.com/future-of-manufacturing**

완전히 통합된 자동화 시스템 (TIA)  
**siemens.com/tia**

TIA Portal  
**siemens.com/tia-portal**

SIMATIC 컨트롤러  
**siemens.com/controller**

SIMATIC 기술 문서   
**siemens.com/simatic-docu**

산업 온라인 지원  
**support.industry.siemens.com**

제품 카탈로그 및 온라인 주문 시스템 산업 몰   
**mall.industry.siemens.com**

Siemens AG  
Digital Factory   
P.O. Box 4848  
90026 뉘른베르크  
독일

오류는 제외되며 사전 통보없이 변경될 수 있습니다.  
© Siemens AG 2018

**siemens.com/sce**