

SIEMENS



# SCE 교육 커리큘럼

Siemens Automation Cooperates with Education | 05/2017

## TIA Portal Module 032-300

### IEC 타이머 및 IEC 카운터

Cooperates  
with Education

Automation

SIEMENS

## 교육 커리큘럼에 따른 적합한 SCE 트레이너 패키지

### SIMATIC 컨트롤러

- **SIMATIC ET 200SP Open Controller CPU 1515SP PC F 및 HMI RT SW**  
주문 번호: 6ES7677-2FA41-4AB1
- **SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety**  
주문 번호: 6ES7512-1SK00-4AB2
- **SIMATIC CPU 1516F PN/DP Safety**  
주문 번호: 6ES7516-3FN00-4AB2
- **SIMATIC S7 CPU 1516-3 PN/DP**  
주문 번호: 6ES7516-3AN00-4AB3
- **SIMATIC CPU 1512C PN(소프트웨어 장착) 및 PM 1507**  
주문 번호: 6ES7512-1CK00-4AB1
- **SIMATIC CPU 1512C PN(소프트웨어 장착), PM 1507 및 CP 1542-5 (PROFIBUS)**  
주문 번호: 6ES7512-1CK00-4AB2
- **SIMATIC CPU 1512C PN(소프트웨어 장착)**  
주문 번호: 6ES7512-1CK00-4AB6
- **SIMATIC CPU 1512C PN(소프트웨어 장착) 및 CP 1542-5 (PROFIBUS)**  
주문 번호: 6ES7512-1CK00-4AB7

### 교육용 SIMATIC STEP 7 소프트웨어

- **SIMATIC STEP 7 Professional V14 SP1 - 단일 라이선스**  
주문 번호: 6ES7822-1AA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - 강의실 라이선스 (최대 인원 6명)**  
주문 번호: 6ES7822-1BA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - 업그레이드 라이선스 (최대 인원 6명)**  
주문 번호: 6ES7822-1AA04-4YE5
- **SIMATIC STEP 7 Professional V14 SP1 - 학생 라이선스 (최대 인원 20명)**  
주문 번호: 6ES7822-1AC04-4YA5

위 트레이너 패키지는 필요 시 후속 모델 패키지로 대체가 된다는 점에 유의하십시오. 현재 출시된 SCE 패키지에 대한 개요는 [siemens.com/sce/tp](https://www.siemens.com/sce/tp)에서 제공됩니다.

### 보충 교육

지멘스의 지역별 SCE 보충 교육에 대한 내용은 해당 지역의 SCE 고객 센터로 문의하시기 바랍니다.  
[siemens.com/sce/contact](https://www.siemens.com/sce/contact)

### SCE 관련 추가 정보

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

## 사용 관련 정보

통합 자동화 솔루션인 TIA(Totally Integrated Automation)를 위한 SCE 교육 커리큘럼은 공교육 시설 및 R&D 기관 교육 목적의 "SCE(Siemens Automation Cooperates with Education) 프로그램을 위해 마련된 것입니다. Siemens AG는 프로그램의 내용을 보증하지 않습니다.

본 문서는 지멘스 제품/시스템을 초기 교육하는 용도로만 사용되어야 합니다. 따라서 교육 범위 내에서의 사용 목적으로 전체 또는 일부를 복사하여 교육생들에게 제공할 수 있습니다. 본 문서는 공공 교육 및 고등 교육 시설 내에서의 교육을 위한 목적으로의 배포, 복사 및 내용의 공유가 가능합니다.

예외적인 경우에는 Siemens AG 담당자의 서면 동의가 필요합니다. Roland Scheuerer  
[roland.scheuerer@siemens.com](mailto:roland.scheuerer@siemens.com).

해당 규정의 위반 시에는 그에 대한 책임이 부과될 수 있습니다. 특히 특허가 부여되었거나 실용신안 또는 의장등록이 된 경우, 번역을 포함한 제반 권리는 지멘스의 소유입니다.

산업체 고객을 위한 교육 과정의 사용은 명시적으로 금지됩니다. 지멘스는 교육 커리큘럼의 상업적 이용을 거부합니다.

드레스덴공대(TU Dresden), 특히 공학 박사 Leon Urbas 교수와 Michael Dziallas Engineering Corporation, 그리고 본 교육 커리큘럼을 준비하는 과정에서 도움을 주신 모든 관계자들에게 감사의 말씀을 전합니다.

## 목차

TOC

# SIMATIC S7-1500을 위한 IEC 타이머 및 IEC 카운터 다중 인스턴스

## 1 목표

이 챕터에서는 TIA Portal 프로그래밍 툴을 통해 SIMATIC S7-1500 프로그래밍에서 단일 인스턴스 및 다중 인스턴스를 사용하는 방법에 대해 배워보겠습니다.

이 모듈에는 다양한 유형의 인스턴스 데이터 블록이 설명되어 있으며, IEC 타이머 및 IEC 카운터를 프로그램 블록에 추가하는 방법이 단계별로 나와 있습니다.

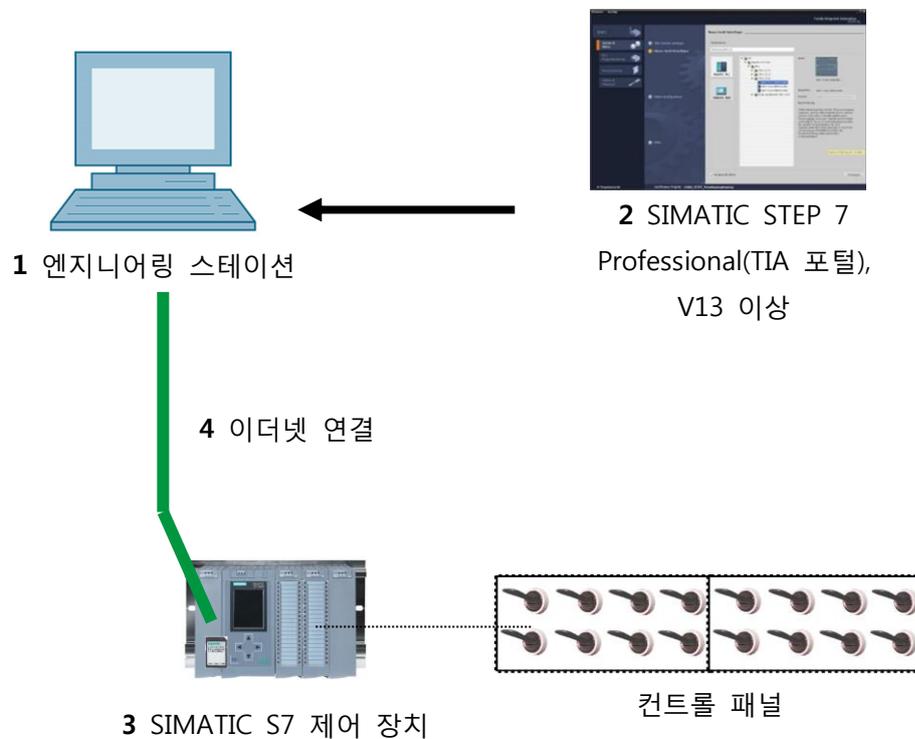
제3장에 기술된 SIMATIC S7 제어 장치를 사용할 수 있습니다.

## 2 전제 조건

이 챕터에서는 SIMATIC S7 CPU1516F-3 PN/DP에서 FB 프로그래밍을 해보겠습니다. 예를 들어 032-200\_FBProgramming\_R1503.zip 같은 프로젝트를 이 챕터에서 사용할 수 있습니다.

### 3 필요한 하드웨어 및 소프트웨어

- 1 엔지니어링 스테이션: 하드웨어 및 운영 시스템이 필요합니다(자세한 정보는 TIA 포털의 설치 DVD Readme/Liesmich를 참조하세요).
- 2 TIA 포털의 소프트웨어 SIMATIC STEP 7 Professional – V13부터
- 3 SIMATIC S7-1500/S7-1200/S7-300 제어 장치, 예: CPU 1516F-3 PN/DP – 펌웨어 버전 V1.6 이상, 메모리 카드와 16DI/16DO 및 2AI/1AO 포함 참고: 디지털 입력은 컨트롤 패널에서 실행되어야 합니다.
- 4 엔지니어링 스테이션과 제어 장치 간 이더넷 연결



## 4 이론

### 4.1 SIMATIC S7-1500의 인스턴스 및 다중 인스턴스

평선 블록의 호출을 인스턴스라고 합니다. 인스턴스는 평선 블록을 호출할 때마다 지정이 되며 데이터 메모리의 역할을 합니다. 인스턴스는 평선 블록의 실제(actual) 파라미터와 정적 데이터를 저장합니다.

평선 블록에서 선언된 태그에 따라 인스턴스 데이터 블록의 구조를 결정합니다.

#### 단일 인스턴스 및 다중 인스턴스 사용

인스턴스는 다음과 같이 지정할 수 있습니다.

단일 인스턴스로서 호출:

- 평선 블록의 각 인스턴스마다 별도의 인스턴스 데이터 블록이 존재

다중 인스턴스로서 호출:

- 1개 이상의 평선 블록의 여러 인스턴스에 하나의 인스턴스 데이터 블록만 존재

#### 4.1.1 인스턴스 데이터 블록/단일 인스턴스

자체 인스턴스 데이터 블록이 지정된 평선 블록의 호출을 단일 인스턴스라고 합니다.

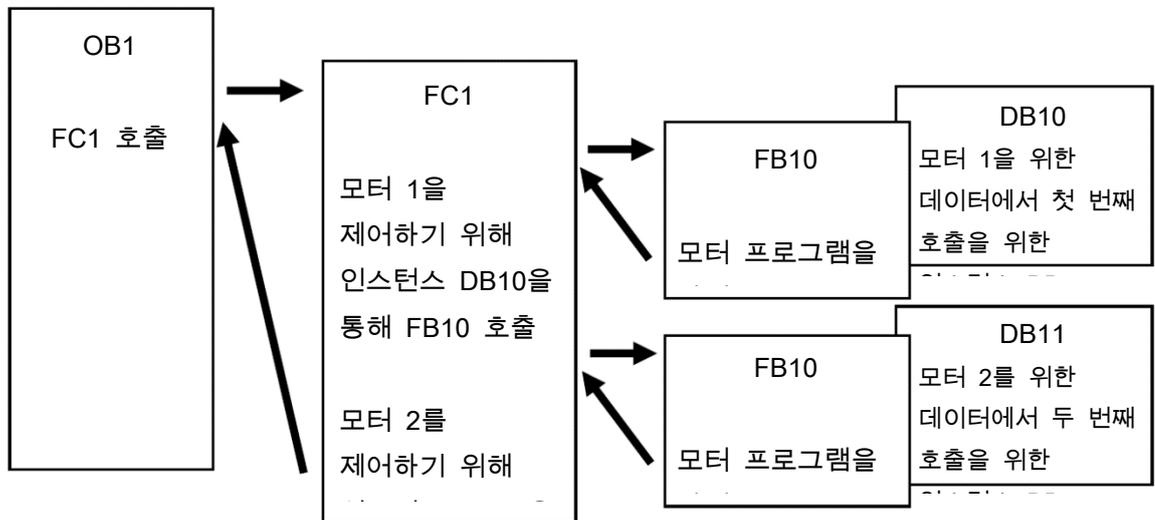
라이브러리 호환 표준 블록을 위한 규칙에 따라 평선 블록이 생성된 경우에는 여러 차례 호출이 가능합니다.

그러나, 단일 인스턴스를 각각 호출시에 서로 다른 인스턴스 데이터 블록을 지정해야 합니다.

단일 인스턴스의 예:

아래 그림에는 1개의 평선 블록 FB10을 사용하는 2개의 모터와 2개의 데이터 블록을 제어하는 방법이 나와 있습니다.

속도, 가속 시간 및 총 동작 시간 같이 각 모터별로 다른 데이터가 인스턴스 데이터 블록 DB10 및 DB11에 저장됩니다.



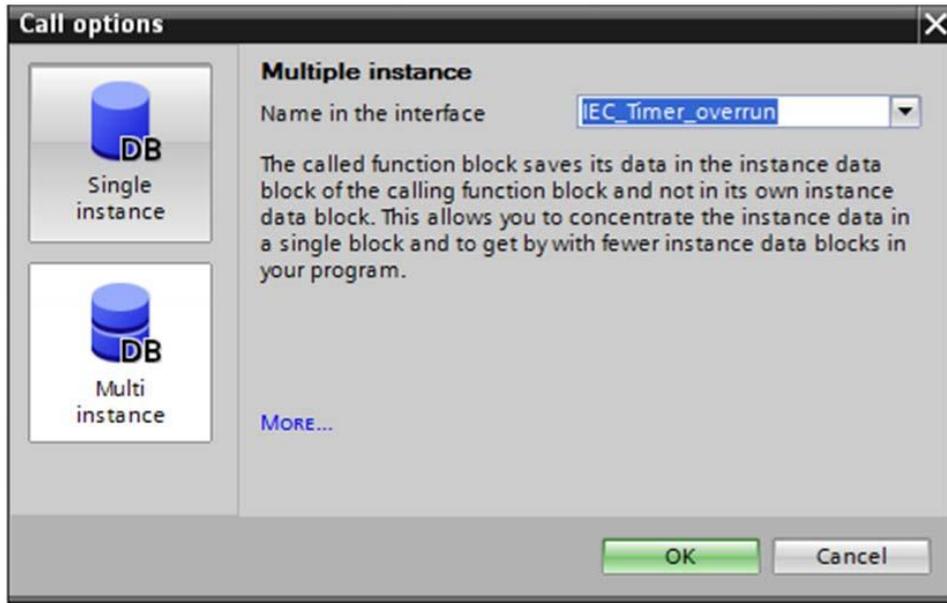
*참고: 타이머나 카운터 같은 일부 명령어들 평선 블록처럼 동작합니다. 이들 명령어를 호출하려면 지정된 메모리 영역(예를 들어 인스턴스 데이터 블록의 형태)이 필요합니다.*

### 4.1.2 다중 인스턴스

사용하는 CPU에서 메모리가 부족하여 인스턴스가 사용하는 데이터 블록의 수를 제한해야 하는 경우가 있을 수 있습니다.

사용자 프로그램의 평션 블록에서 이미 존재하는 기타 평션 블록, 타이머, 카운터 등은 호출이 될 경우에는 별도의 추가적인 인스턴스 DB 없이도 이들을 호출할 수 있습니다.

Call options에서 'Multi-instance'를 선택하면 됩니다.



*참고: 다중 인스턴스는 호출된 평션 블록이 호출한 평션 블록의 인스턴스 데이터 블록에 데이터를 저장하도록 해줍니다.*

*이 경우 호출하는 블록은 반드시 평션 블록이어야 합니다.*

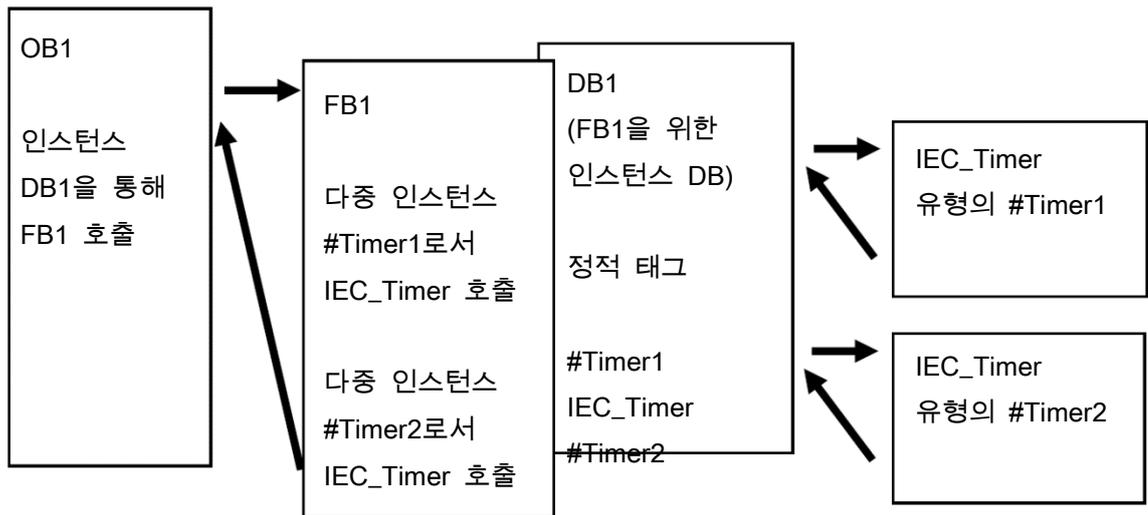
*이렇게 하면 하나의 인스턴스 데이터 블록에 인스턴스 데이터를 집중시켜서 가용 DB의 수가 많아져 더 효과적으로 활용할 수 있습니다.*

*호출하는 블록이 표준 블록으로서 재사용이 가능한 상태로 유지되어야 한다면, 다중 인스턴스가 반드시 필요합니다.*

다중 인스턴스의 예:

아래 그림에는 하나의 평선 블록 내에서 TP(펄스) 유형의 IEC\_Timer를 두 번 호출하는 작업이 나와 있습니다.

두 카운터의 데이터는 호출 평선 블록 FB1의 인스턴스 데이터 블록 DB1에 서로 다른 다중 인스턴스로서 저장됩니다.



## 5 과제

이 챕터에서는 "SCE\_EN\_032-200 FB 프로그래밍" 챕터에서 생성된 평선 블록에 IEC 타이머를 추가해 보겠습니다.

## 6 계획 수립

IEC 타이머는 "032-200\_FBProgramming.zap13" 프로젝트에서 MOTOR\_AUTO [FB1] 평선 블록에 추가되도록 프로그래밍이 되어 있습니다. IEC 타이머 TP(래칭 펄스)를 추가하려면 반드시 이 프로젝트를 사용해야 합니다. 다중 인스턴스는 타이머가 사용하는 메모리로서 생성이 됩니다.

### 6.1 자동 모드 - 타임 평선을 포함한 컨베이어 모터

은 리셋 조건이 충족되지 않을 경우, 시작 시 래치되어 값이 1이 됩니다.

정지 기능이 작동되거나, 안전 전원 차단이 활성화 되어 있거나, 자동 모드가 활성화 되어 있지 않은 경우(수동 모드)에는 Memory\_automatic\_start\_stop이 리셋됩니다.

따라서 Memory\_automatic\_start\_stop이 셋(set)되어 있고 시작 조건이 충족되며 Memory\_conveyor\_start\_stop이 셋되어 있을 때만 Conveyor\_motor\_automatic\_mode 출력이 작동됩니다.

에너지 절약을 위해서는 부품이 있을 때만 컨베이어가 작동해야 합니다.

이러한 이유로 Memory\_conveyor\_start\_stop은 Sensor\_chute\_occupied가 부품을 있음을 알릴 때 셋 되고, Sensor\_end\_of\_conveyor가 네거티브 엷지를 만들어 내거나 안전 전원 차단이 활성화되거나 자동 모드가 활성화되어 있지 않을 때(수동 모드) 리셋됩니다.

타임 평선 추가:

컨베이어 끝에서 Sensor\_end\_of\_conveyor를 직접 마운팅할 수 없기 때문에 Sensor\_end\_of\_conveyor 신호를 더 길게 연장해야 합니다.

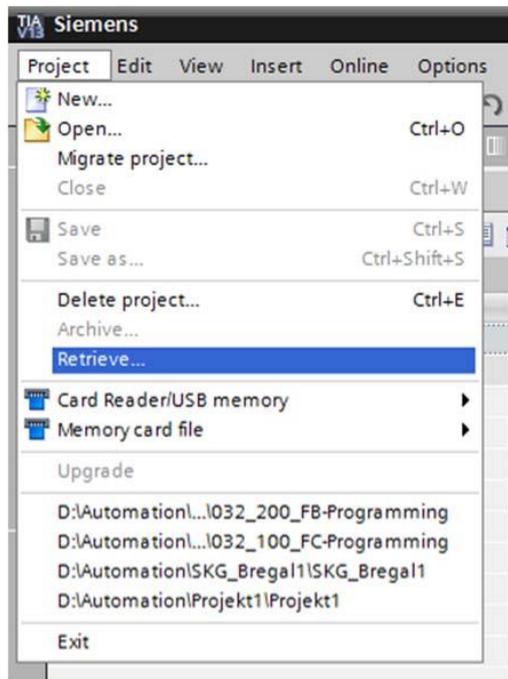
이를 위해 Sensor\_end\_of\_conveyor와 네거티브 엷지 감지 사이에 래칭 펄스가 삽입됩니다.

## 7 단계별 따라 해보기

아래에는 계획을 수립하는 방법에 대한 지침이 나와 있습니다. 모든 내용을 이미 충분히 숙지했다면 숫자가 표시된 단계로 넘어가도 좋습니다. 그렇지 않다면, 아래에 나와 있는 지침의 단계를 따라가면 됩니다.

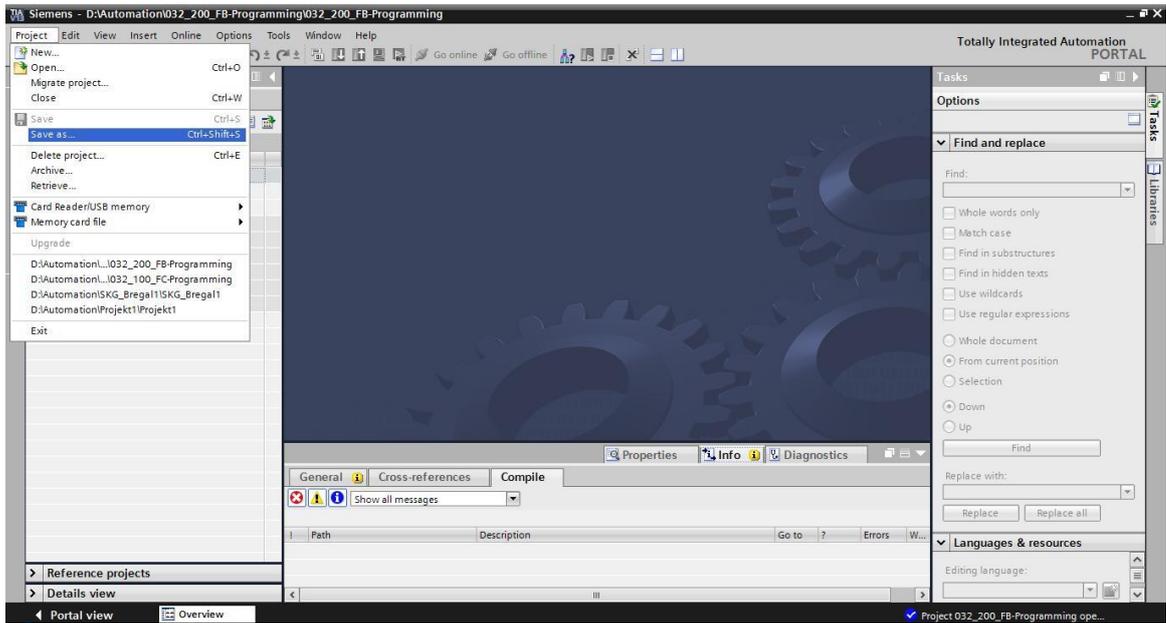
### 7.1 기존 프로젝트 압축풀기

- "MOTOR\_AUTO [FB1]" 평선 블록을 수정하려면, 먼저 "SCE\_EN\_032-200 FBProgramming" 챗터에서 생성된 "032-200\_FBProgramming.zap13" 프로젝트의 압축을 풀어야 합니다. 아카이브된 기존 프로젝트의 압축을 풀려면 "Project"의 "Retrieve..."에서 해당되는 아카이브를 선택해야 합니다. "Open"을 클릭해 선택합니다. (→ Project → Retrieve... → .zap 아카이브 선택 → Open)



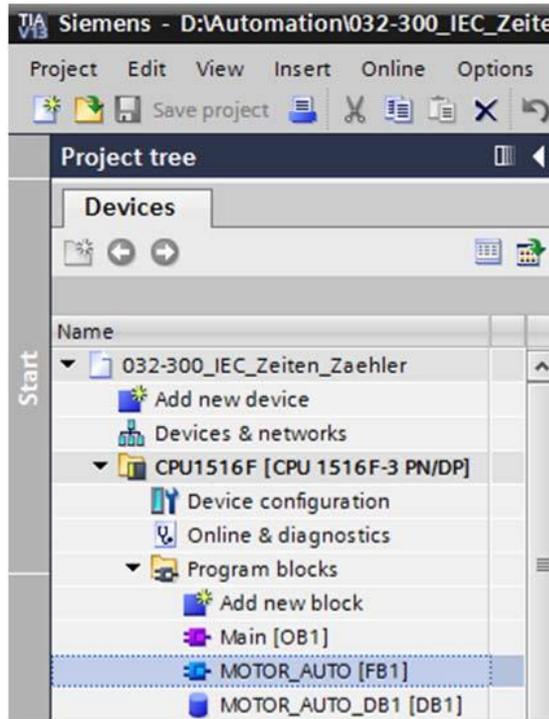
- 그 다음으로 이 프로젝트가 저장될 대상 디렉토리를 선택합니다. "OK"를 눌러 선택합니다. (→ Target directory → OK)

이 프로젝트를 032-300\_IEC\_Timers\_Counters라는 이름으로 저장을 합니다. (→ Project → Save as ... → 032-300-IEC\_Timers\_Counters → Save)

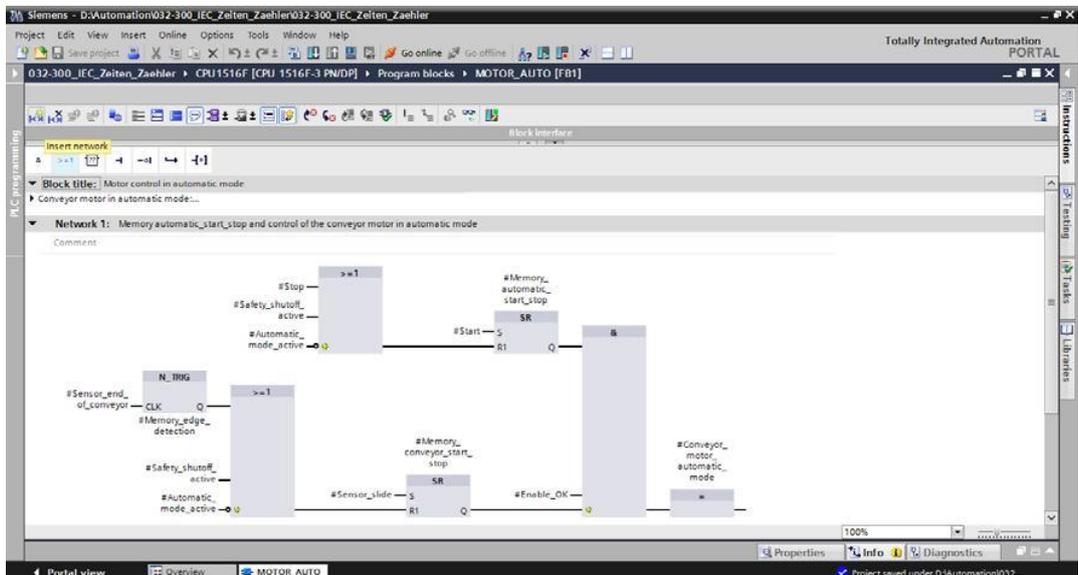


## 7.2 평선 블록 FB1 "MOTOR\_AUTO"에 IEC 타이머 TP 추가

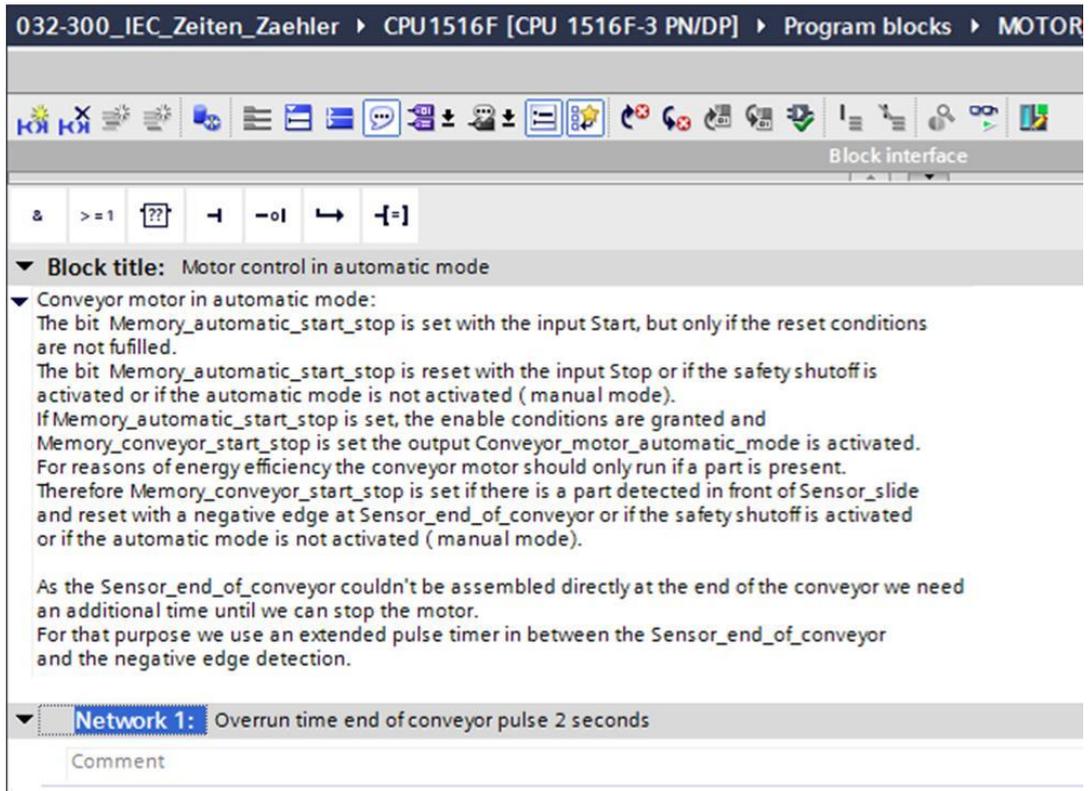
→ 먼저, "MOTOR\_AUTO [FB1]" 평선 블록을 더블클릭해서 엽니다.



→ "Block title"을 선택하고 "Insert network"  아이콘을 클릭해서 "MOTOR\_AUTO [FB1]" 평선 블록의 시작 지점에 새로운 네트워크를 삽입합니다.

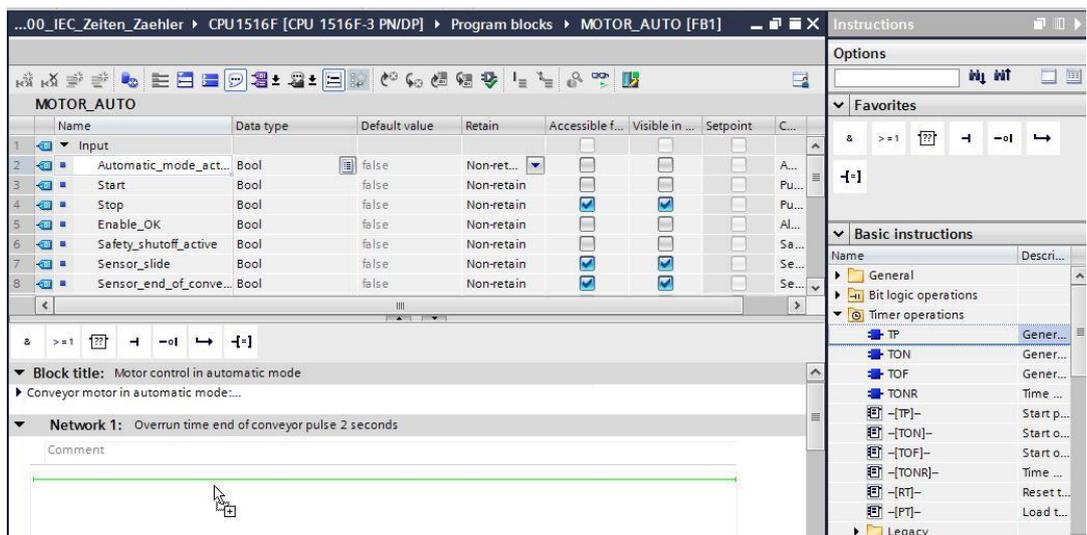


→ 블록 코멘트와 "Network 1:"의 네트워크 타이틀에 도움이 되는 정보를 추가합니다.

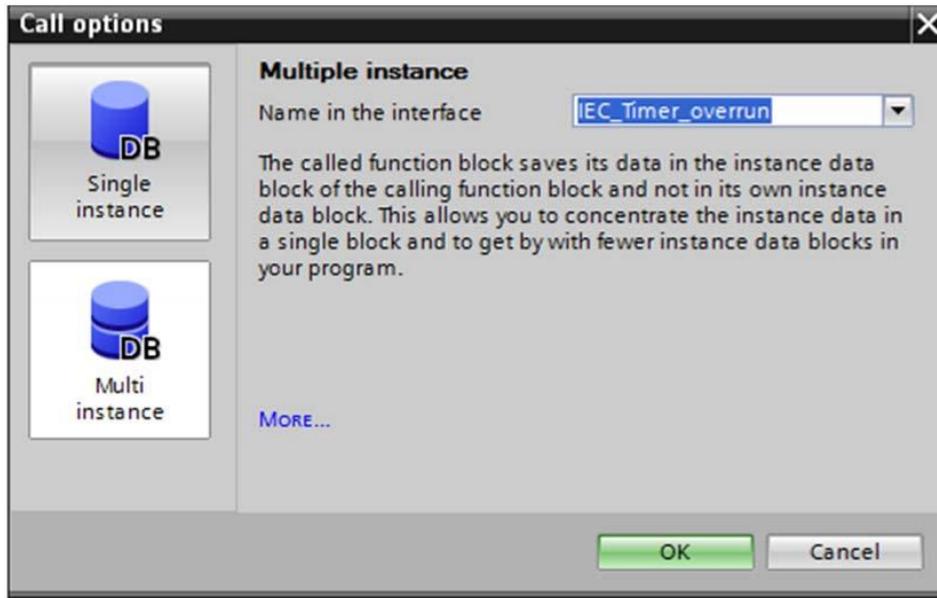


→ 프로그래밍 창 오른쪽의 명령어 목록에 타이머 평선이 나타납니다. "Basic instructions" 아래의 "Timer operations"으로 가서 평선  (Generate pulse)를 찾아 끌어다 놓기 기능을 이용해 Network 1로 이를 이동시킵니다 (녹색 선이 나타나고 마우스 포인터에 + 심볼이 표시됨).

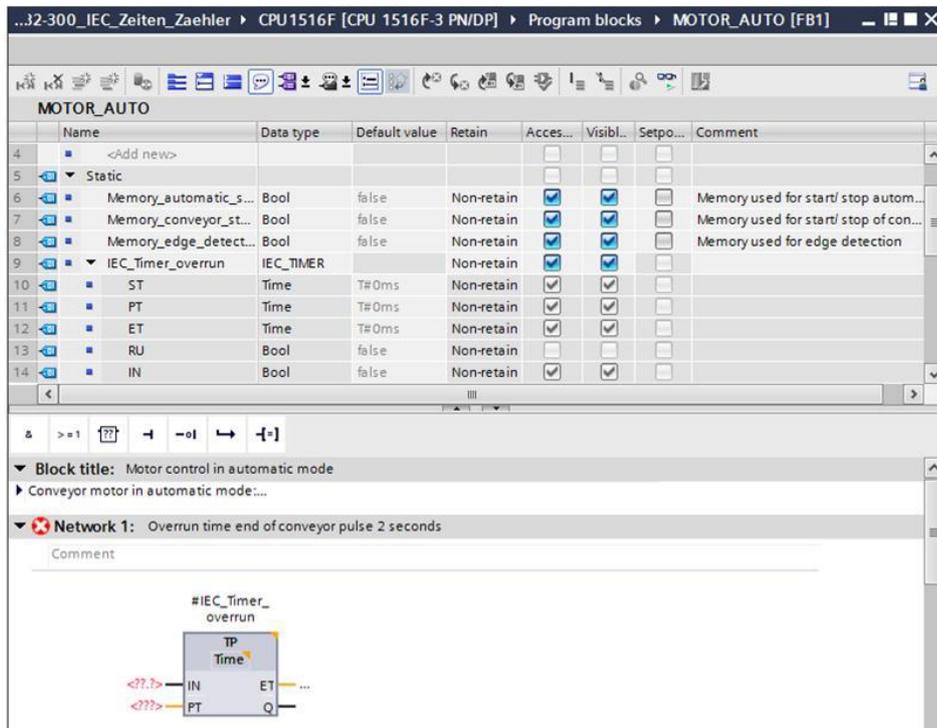
(→Instructions→ Basic instruction → Timer operations →  )



→ 타이머 평선은 메모리를 필요로 합니다. 새로운 인스턴스 데이터 블록을 생성하지 않아도 평선 블록에 의해 인스턴스 데이터 블록 내에서 타이머 평선의 메모리를 사용할 수 있습니다. 이를 위해서 "multi-instance"를 선택합니다. 다중 인스턴스에 대한 이름을 입력하고 "OK"를 클릭합니다. (→ Multi-instance → IEC\_Timer\_overrun → OK)

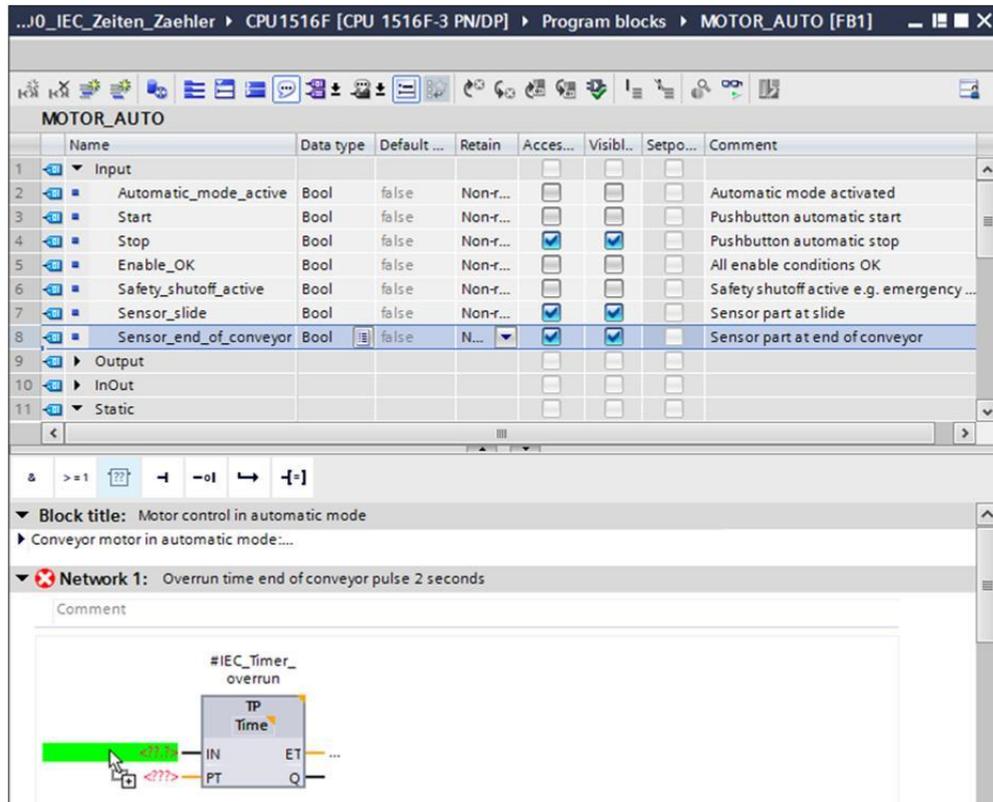


→ 그 결과, 인터페이스 목록에 TP 타이머에 적합한 "Static" 유형의 태그 구조가 생성됩니다.

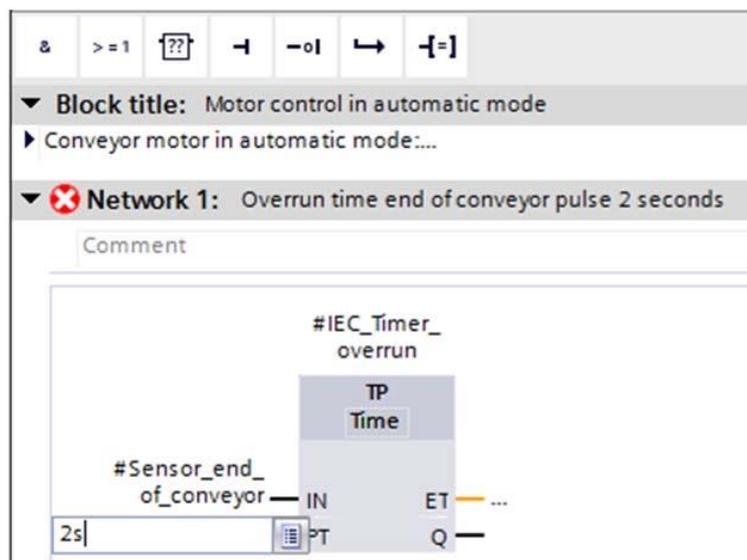


참고: 스태틱 태그는 평선 블록에서만 사용할 수 있기 때문에 다중 인스턴스는 평선 블록에서만 프로그램할 수 있습니다.

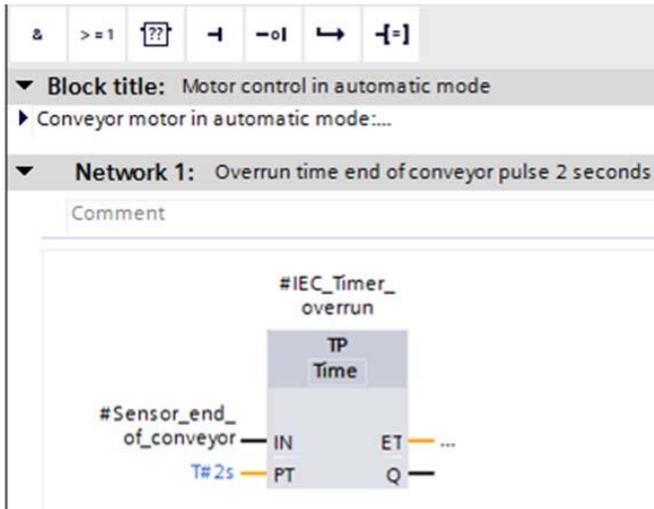
- 입력 #Sensor\_end\_of\_conveyor의 포지티브 엣지에서 타이머가 시작 되도록 하기위해, 입력 파라미터 #Sensor\_end\_of\_conveyor를 TP 타이머의 "IN" 파라미터 앞에 있는 <??>로 끌어다 놓습니다. 인터페이스 목록에서 파라미터를 선택하는 가장 좋은 방법은 파란색 심볼 를 클릭하여 끌어오는 것입니다. (→  Sensor\_end\_of\_conveyor)



- 파라미터 "PT" 앞에 필요한 펄스 유지 시간을 2초로 입력합니다. (→ 2s )

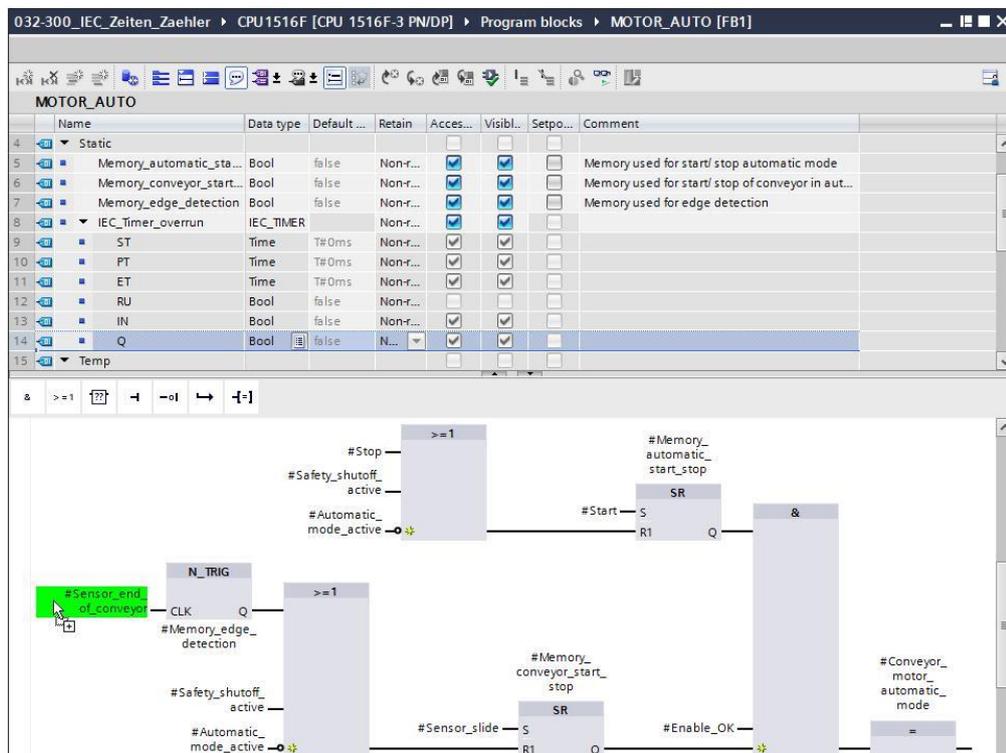


→ 2초라는 입력값이 IEC 타이머에 적합한 IEC-Time 형식으로 자동 변환되어 상수 "T#2s"로 표시가 됩니다.



→ 이제, 태그 구조 "IEC\_Timer\_ouerrun"에서 나온 출력 "Q"를 네트워크 2의 네거티브 엣지 "N\_TRIG"의 입력 "CLK"로 이동시킵니다. 이 값이 이전에 사용했던 #Sensor\_end\_of\_conveyor 입력 태그를 대신하게 되며, IEC\_Timer\_ouerrun 펄스의 네거티브 엣지에 의해 컨베이어의 작동이 정지됩니다.

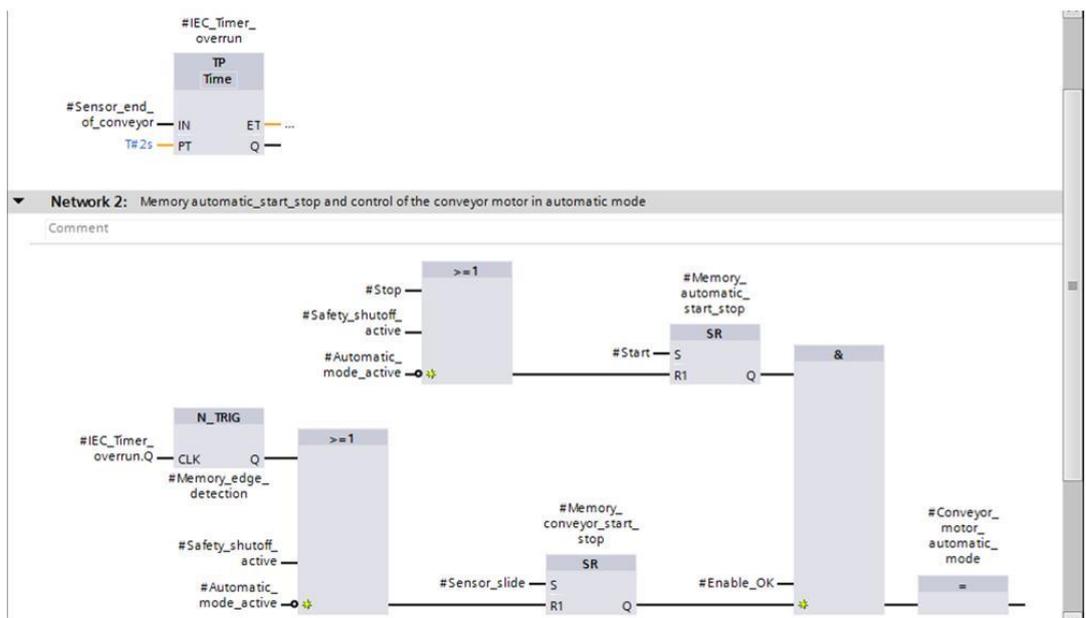
(→ Network 2 → IEC\_Timer\_ouerrun→ Q → #Sensor\_end\_of\_conveyor)



→  Save project 를 클릭하여 프로젝트를 저장합니다. 타이머가 포함되어 작성 완료된 평선 블록 "MOTOR\_AUTO" [FB1]가 아래와 같이 FBD 언어로 나타납니다.

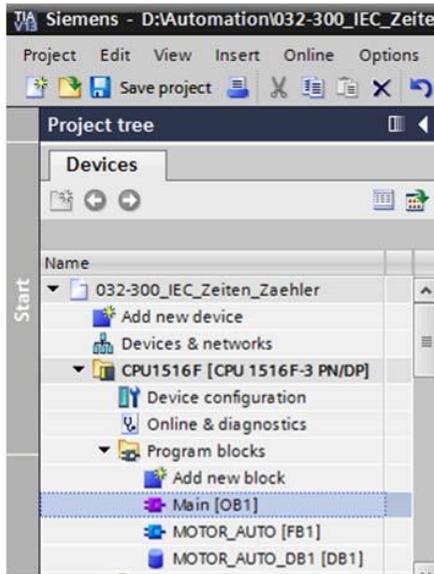
032-300\_IEC\_zeiten\_Zaehler > CPU1516F [CPU 1516F-3 PN/DP] > Program blocks > MOTOR\_AUTO [FB1]

Name	Data type	Default ...	Retain	Acces...	Visibl...	Setpo...	Comment
1	Input						
2	Automatic_mode_active	Bool	false	Non-r...	<input type="checkbox"/>	<input type="checkbox"/>	Automatic mode activated
3	Start	Bool	false	Non-r...	<input type="checkbox"/>	<input type="checkbox"/>	Pushbutton automatic start
4	Stop	Bool	false	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Pushbutton automatic stop
5	Enable_OK	Bool	false	Non-r...	<input type="checkbox"/>	<input type="checkbox"/>	All enable conditions OK
6	Safety_shutoff_active	Bool	false	Non-r...	<input type="checkbox"/>	<input type="checkbox"/>	Safety shutoff active e.g. emergency stop o...
7	Sensor_slide	Bool	false	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor part at slide
8	Sensor_end_of_conveyor	Bool	false	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sensor part at end of conveyor
9	Output						
10	Conveyor_motor_autom...	Bool	false	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Control of the conveyor motor in automati...
11	InOut						
12	Static						
13	Memory_automatic_sta...	Bool	false	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memory used for start/ stop automatic mode
14	Memory_conveyor_start...	Bool	false	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memory used for start/ stop of conveyor in ...
15	Memory_edge_detection	Bool	false	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memory used for edge detection
16	IEC_Timer_overnun	IEC_TIMER		Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	ST	Time	T#0ms	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	PT	Time	T#0ms	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	ET	Time	T#0ms	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	RU	Bool	false	Non-r...	<input type="checkbox"/>	<input type="checkbox"/>	
21	IN	Bool	false	Non-r...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	Q	Bool	false	N...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	Temp						

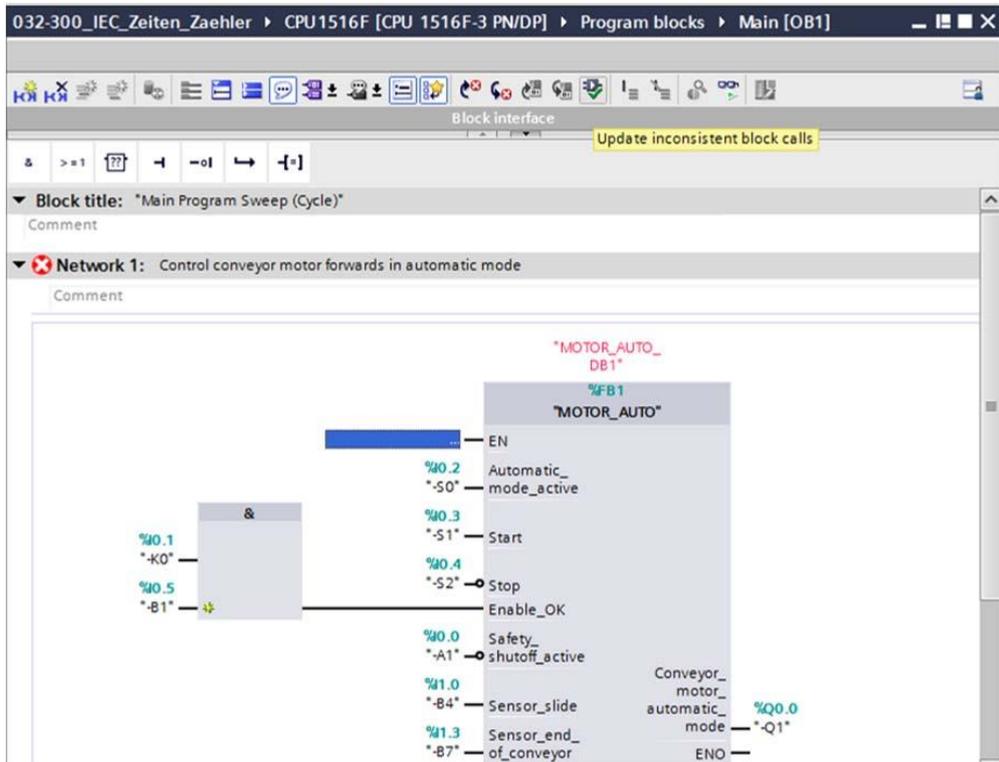


### 7.3 오거나이제이션 블록에서 블록 호출 업데이트

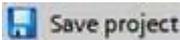
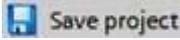
→ "Main [OB1]" 오거나이제이션 블록을 더블클릭해서 엽니다.

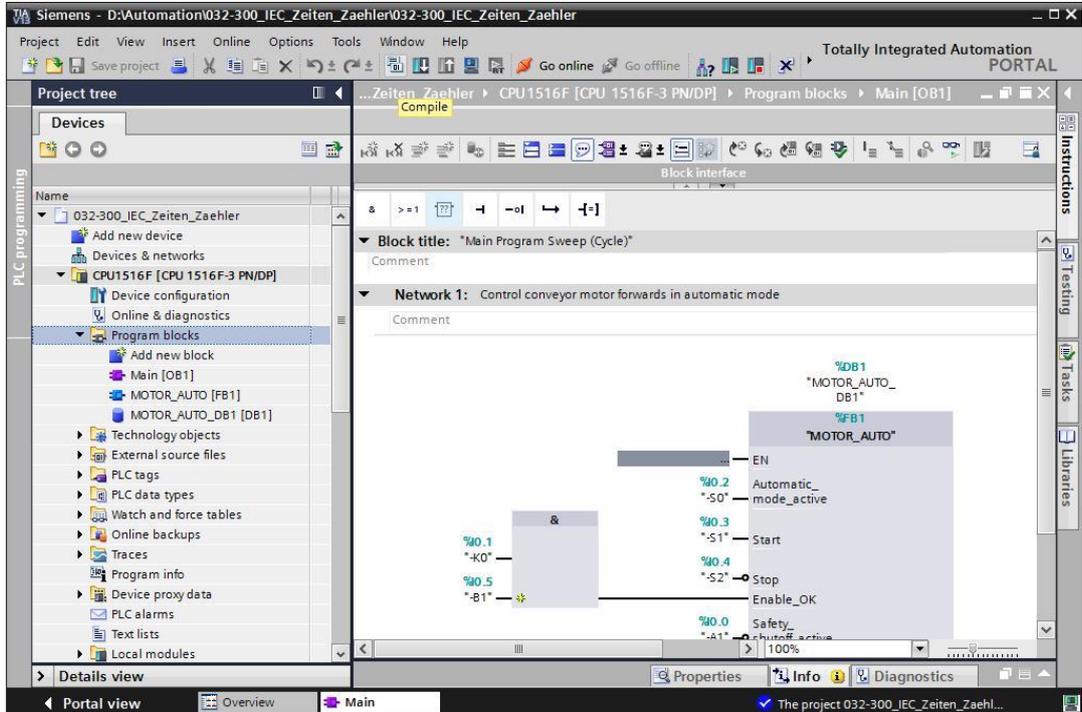


→ TP 타이머의 추가 메모리가 인스턴스 데이터 블록에 아직 추가되지 않았기 때문에 "Main [OB1]" 오거나이제이션 블록의 네트워크 1에서 "MOTOR\_AUTO [FB1]" 평선 블록을 위한 인스턴스 데이터 블록 "MOTOR\_AUTO\_DB1"이 적색으로 올바르게 표시되지 않습니다. "Update inconsistent block calls"  아이콘을 클릭합니다. 이렇게 하면 "MOTOR\_AUTO\_DB1" 인스턴스 데이터 블록이 올바르게 다시 추가가 됩니다. (→ )

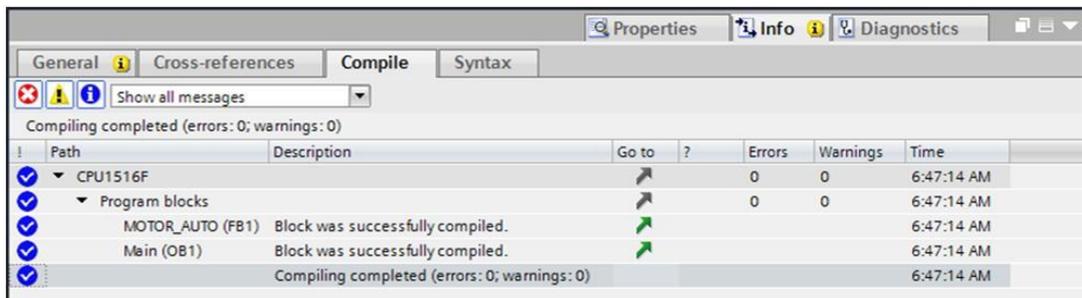


## 7.4 프로그램 저장 및 컴파일

→ 프로젝트를 저장하려면 메뉴에서  Save project 버튼을 선택합니다. 모든 블록을 컴파일하려면 "Program blocks" 폴더를 클릭하고 메뉴에서 컴파일을 위한 아이콘  을 선택합니다. (→  Save project → Program blocks → ).

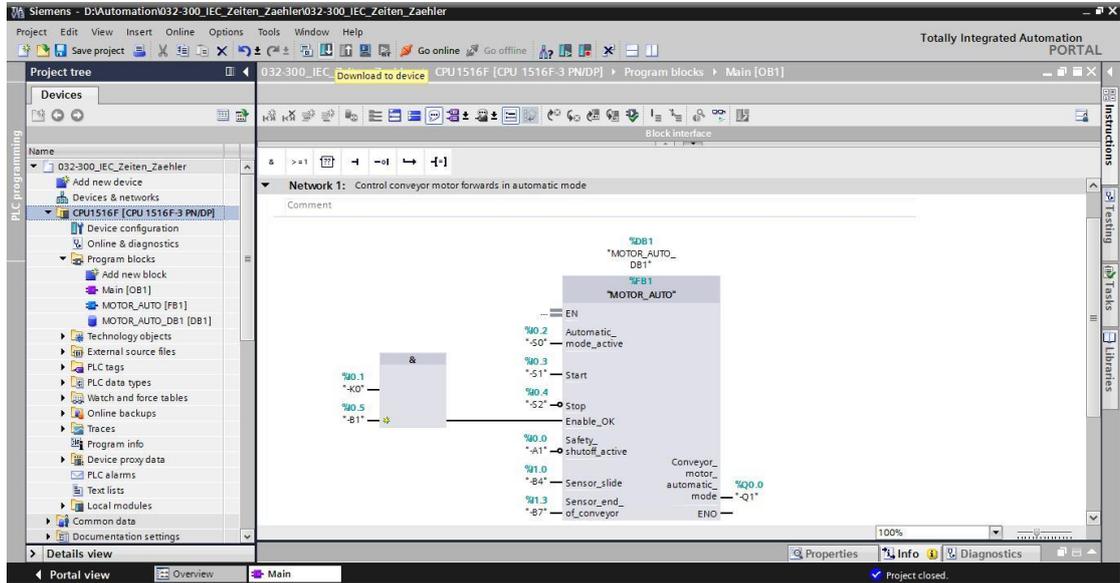


→ "Info" 아래의 "Compile" 영역에 블록이 성공적으로 컴파일되었는지 나타냅니다.



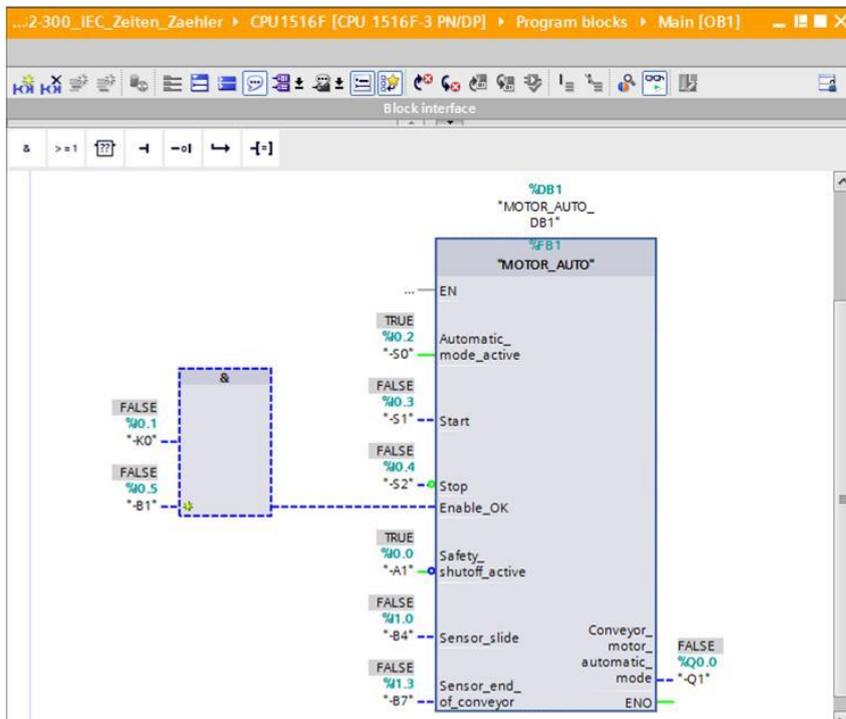
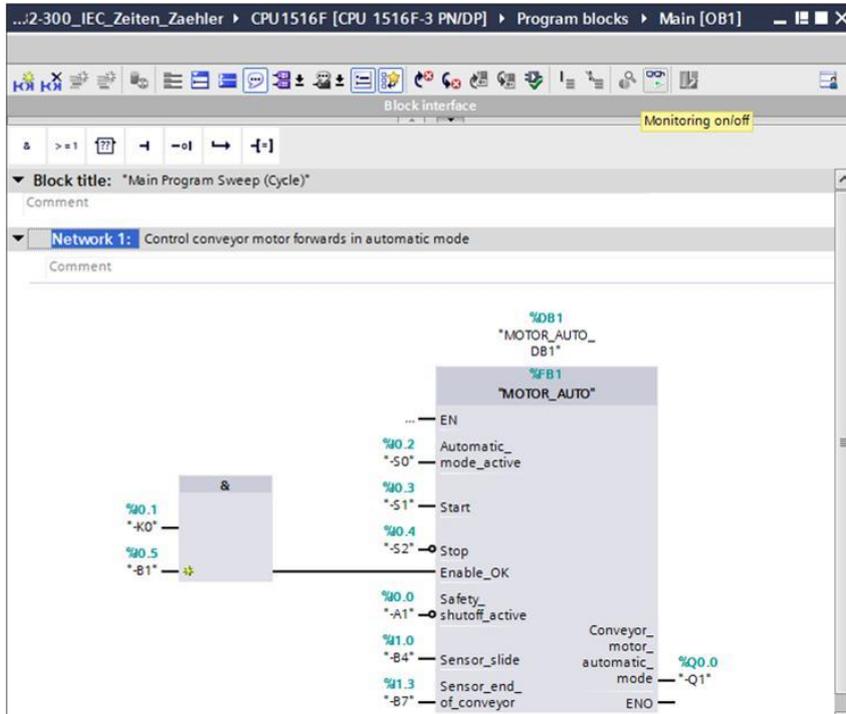
## 7.5 프로그램 다운로드

→ 컴파일이 성공적으로 완료되고 나면 앞서 설명한 하드웨어 구성을 위한 모듈에서와 같이 생성된 프로그램과 함께 전체 컨트롤러를 다운로드할 수 있습니다. (→  )



## 7.6 프로그램 블록 모니터링

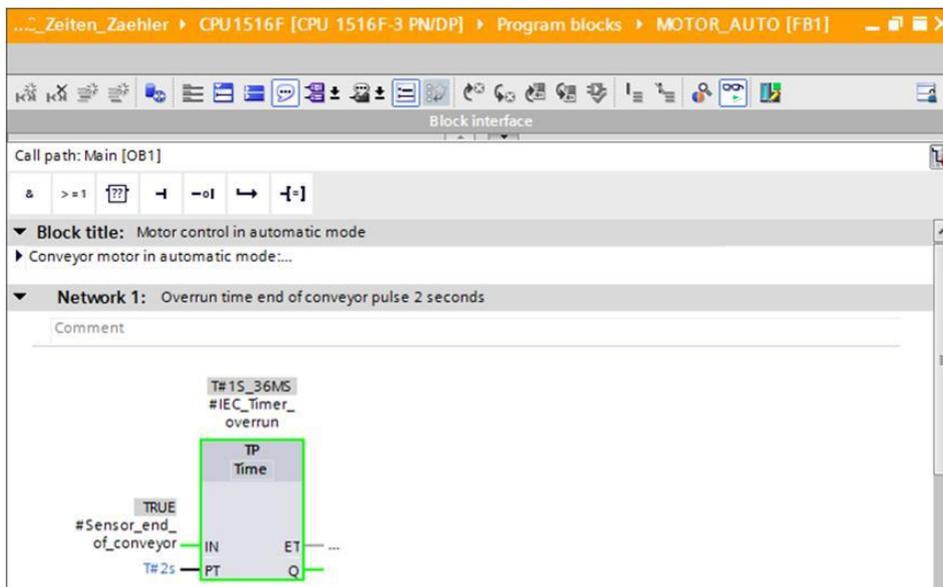
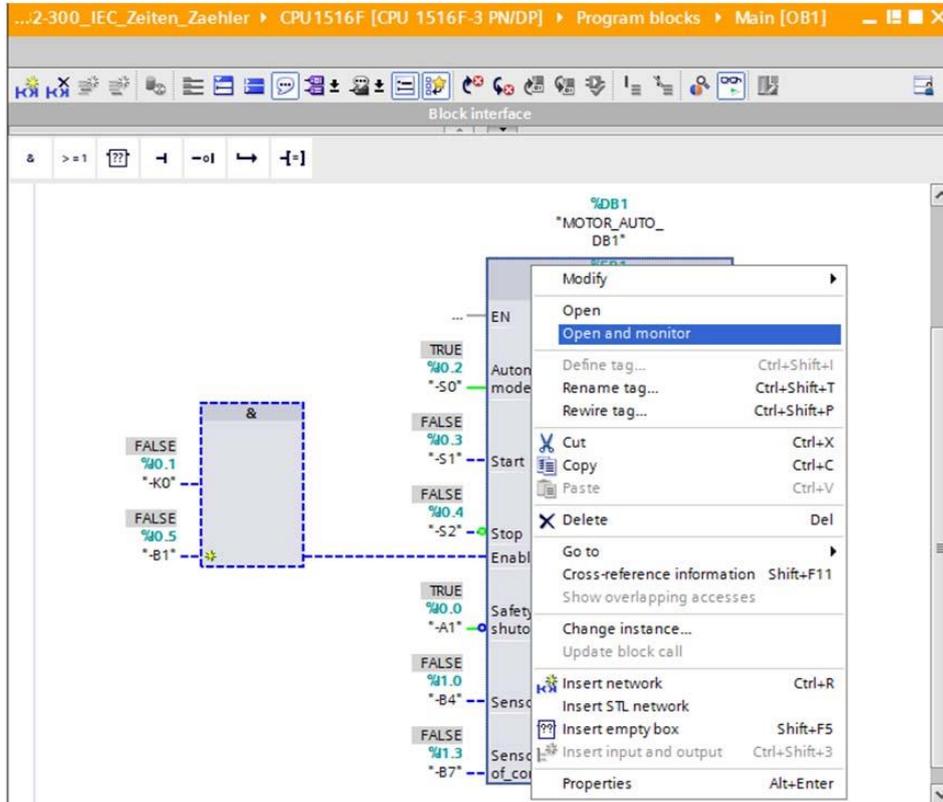
→ 다운로드된 프로그램을 모니터링하려면 원하는 블록을 열어야 합니다.  아이콘을 클릭해서 모니터링을 활성화/비활성화할 수 있습니다. (→ Main [OB1] →  )



참고: 위의 모니터링 상태는 신호값과 컨트롤러에 다르게 보일 수 있습니다. 터미널에서의 신호 상태는 TRUE 또는 FALSE로 표시가 됩니다.

→ 마우스 오른쪽 버튼을 클릭하여 "Open and monitoring"에서 "Main [OB1]" 오거나이제이션 블록에서 호출된 "MOTOR\_AUTO" [FB1] 평선을 직접 선택할 수 있습니다. 따라서 TP 타이머가 있는 평선 블록의 프로그램 코드를 모니터링할 수 있습니다.

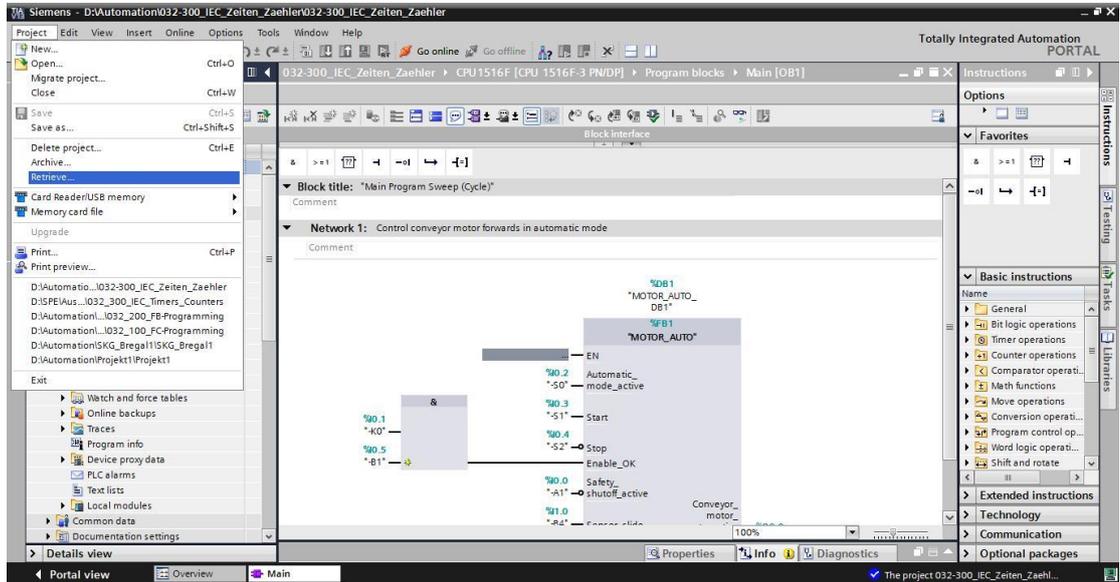
(→ "MOTOR\_AUTO" [FB1] → Open and monitoring)



참고: 위의 모니터링 상태는 평선과 컨트롤러에 따라 다르게 보일 수 있습니다. 센서의 상태와 스테이션 상태가 여기에 TRUE 또는 FALSE로 표시가 됩니다.

## 7.7 프로젝트 아카이브

- 마지막 단계로 전체 프로젝트를 아카이브하기 위해 "Project" 메뉴에서 "Archive ..." 항목을 선택합니다. → 프로젝트를 아카이브하고자 하는 폴더를 선택하고 "TIA Portal project archive" 파일 유형으로 이를 저장합니다. (→ Project → Archive → TIA Portal project archive → 032-300\_IEC\_Timers\_Counters.... → Save)



## 8 체크리스트

번호	설명	완료
1	오류 메시지 없이 성공적으로 컴파일	
2	오류 메시지 없이 성공적으로 다운로드	
3	스테이션 전원 켜기 (-K0 = 1) 실린더 복귀 / 피드백 활성화 (-B1 = 1) 비상 정지 오프 (-A1 = 1)가 활성화되지 않음 자동 모드 (-S0 = 1) 푸시버튼 자동 정지가 구동되지 않음 (-S2 = 1) 자동 시작 푸시버튼을 짧게 누르기 (-S2 = 1) 이송 장치의 센서가 활성화되고 (-B4 = 1) 컨베이어 모터 고정 속도로 전진 (-Q1 = 1) 스위치를 켜 상태로 유지	
4	컨베이어 끝의 센서 활성화 (-B7 = 1) → -Q1 = 0 (2초 후)	
5	자동 정지 푸시버튼을 짧게 누르기 (-S2 = 0) → -Q1 = 0	
6	비상 정지 오프를 활성화 (-A1 = 0) → -Q1 = 0	
7	수동 모드 (-S0 = 0) → -Q1 = 0	
8	스테이션 전원 끄기 (-K0 = 0) → -Q1 = 0	
9	실린더가 복귀되지 않음 (-B1 = 0) → -Q1 = 0	
10	프로젝트가 성공적으로 아카이브 됨	

## 9 연습

### 9.1 과제 – 연습

또한, 이 연습에서는 IEC 카운터를 MOTOR\_AUTO [FB1] 평선 블록에 추가해 보겠습니다.  
그리고 확장된 평선 블록을 계획, 프로그래밍 및 테스트해보겠습니다.

플라스틱 컴포넌트 보관함은 5개의 부품만 수용할 수 있기 때문에 컨베이어 끝에서 부품이 카운팅 됩니다.

5개의 부품이 창고에 저장되면 자동 모드가 정지됩니다.

보관함이 빈 상태가 되면 자동 모드가 다시 재시작됩니다.

Start\_command이 다시 시작되고 카운터가 리셋됩니다.

### 9.2 계획 수립

과제 수행에 대한 계획을 스스로로 수립합니다.

*참고: 온라인 도움말에서 SIMATIC S7-1500에서의 IEC 카운터 사용에 대해 알아보십시오.*

### 9.3 체크리스트 - 연습

번호	설명	완료
1	오류 메시지 없이 성공적으로 컴파일	
2	오류 메시지 없이 성공적으로 다운로드	
3	스테이션 전원 켜기 (-K0 = 1) 실린더 복귀 / 피드백 활성화 (-B1 = 1) 비상 정지 오프 (-A1 = 1)가 활성화되지 않음 자동 모드 (-S0 = 1) 푸시버튼 자동 정지가 구동되지 않음 (-S2 = 1) 자동 시작 푸시버튼을 짧게 누르기 (S1 = 1) 이송 장치의 센서가 활성화되고 (-B4 = 1) 컨베이어 모터 고정 속도로 전진 (-Q1 = 1) 스위치를 켜 상태로 유지	
4	컨베이어 끝의 센서 활성화 (-B7 = 1) → -Q1 = 0 (2초 후)	
5	자동 정지 푸시버튼을 짧게 누르기 (-S2 = 0) → -Q1 = 0	
6	비상 정지 오프를 활성화 (-A1 = 0) → -Q1 = 0	
7	수동 모드 (-S0 = 0) → -Q1 = 0	
8	스테이션 전원 끄기 (-K0 = 0) → -Q1 = 0	
9	실린더가 복귀되지 않음 (-B1 = 0) → -Q1 = 0	
10	창고의 5번째 부품 → -Q1 = 0	
11	프로젝트가 성공적으로 아카이브 됨	

## 10 추가 정보

초기 및 심화 교육에 방향을 제시하는 도구의 차원에서 TIA Portal 모듈에 대한 추가 정보를 활용할 수 있습니다. 시작하기, 동영상, 교재, 앱, 매뉴얼, 프로그래밍 지침, 체험용 소프트웨어/펌웨어 등을 아래 링크에서 찾아보실 수 있습니다.

[www.siemens.com/sce/s7-1500](http://www.siemens.com/sce/s7-1500)