



SIEMENS



교육-/학습 문서

Siemens Automation Cooperates with Education
(SCE) | 버전 V14 SP1 부터

TIA Portal Module 031-100

FC 프로그래밍 기초 SIMATIC S7-1200

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

SIEMENS

Global Industry
Partner of
WorldSkills
International



worldskills

이러한 학습-/교육 문서와 일치하는 SCE 교육 담당자 패키지

- **SIMATIC S7-1200 AC/DC/RELAY(6 개 세트) "TIA Portal"**
주문 번호: 6ES7214-1BE30-4AB3
- **SIMATIC S7-1200 DC/DC/DC(6 개 세트) "TIA Portal"**
주문 번호: 6ES7214-1AE30-4AB3
- **업그레이드 SIMATIC STEP 7 BASIC V14 SP1(S7-1200 용)(6 개 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](https://www.siemens.com/sce/tp)

교육 연장

지역별 Siemens SCE 교육 연장은 지역 SCE 담당자에게 문의하십시오. [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 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 AG 는 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 Corporation 을 비롯한 모든 관계자들에게 이 학습-/교육 문서를 준비하는 동안 보내주신 성원에 대해 감사를 표하고자 합니다.

목차

1	목적.....	5
2	사전조건.....	5
3	필수 하드웨어와 소프트웨어	6
4	이론.....	7
4.1	운영체제 및 애플리케이션 프로그램.....	7
4.2	오거나이제이션 블록.....	8
4.3	프로세스 이미지 및 주기적 프로그램 처리	9
4.4	평선.....	11
4.5	평선 블록 및 인스턴스 데이터 블록	12
4.6	글로벌 데이터 블록.....	13
4.7	라이브러리 호환 로직 블록	14
4.8	프로그래밍 언어	15
5	과제.....	16
6	계획 수립.....	16
6.1	비상 정지(EMERGENCY STOP).....	16
6.2	수동 모드 – 수동 모드의 컨베이어 모터	16
6.3	기술 다이어그램	17
6.4	참조 목록	18
7	단계별 따라 해보기.....	19
7.1	기존 프로젝트 압축 풀기.....	19
7.2	새 태그 테이블 생성.....	20
7.3	태그 테이블 내에서 새 태그 생성.....	22
7.4	“Tag_table_sorting_station” 가져오기.....	23
7.5	수동 모드의 컨베이어 모터를 위한 평선 FC1 “MOTOR_MANUAL” 생성	27
7.6	평선 FC1 “MOTOR_MANUAL”에 대한 인터페이스 정의.....	29
7.7	프로그램 FC1: MOTOR_MANUAL	32
7.8	오거나이제이션 블록 OB1 프로그래밍 – 수동 모드에서 컨베이어 전진 트래킹 제어	39
7.9	프로그램 저장 및 컴파일.....	44

7.10	프로그램 다운로드	45
7.11	프로그램 블록 모니터링	46
7.12	프로젝트 아카이브	48
7.13	체크리스트	49
8	연습	50
8.1	과제 – 연습	50
8.2	기술 다이어그램	50
8.3	참조 목록	51
8.4	계획 수립	51
8.5	체크리스트 – 연습	52
9	추가 정보	53

FC 프로그래밍 기본 사항

1 목적

이 장에서는 **조직 블록(OB)**, **함수(FC)**, **함수 블록(FB)** 및 **데이터 블록(DB)**과 같은 컨트롤 프로그램의 기본 요소를 학습합니다. 또한, **라이브러리-호환 가능** 함수 및 함수 블록 프로그래밍을 소개합니다. **함수 블록 다이어그램(FBD)** 프로그래밍 언어 및 함수(FC1) 및 조직 블록(OB1)을 프로그래밍하는데 사용하는 방법을 학습합니다.

3 장에 나열된 SIMATIC S7 컨트롤러를 사용할 수 있습니다.

2 사전조건

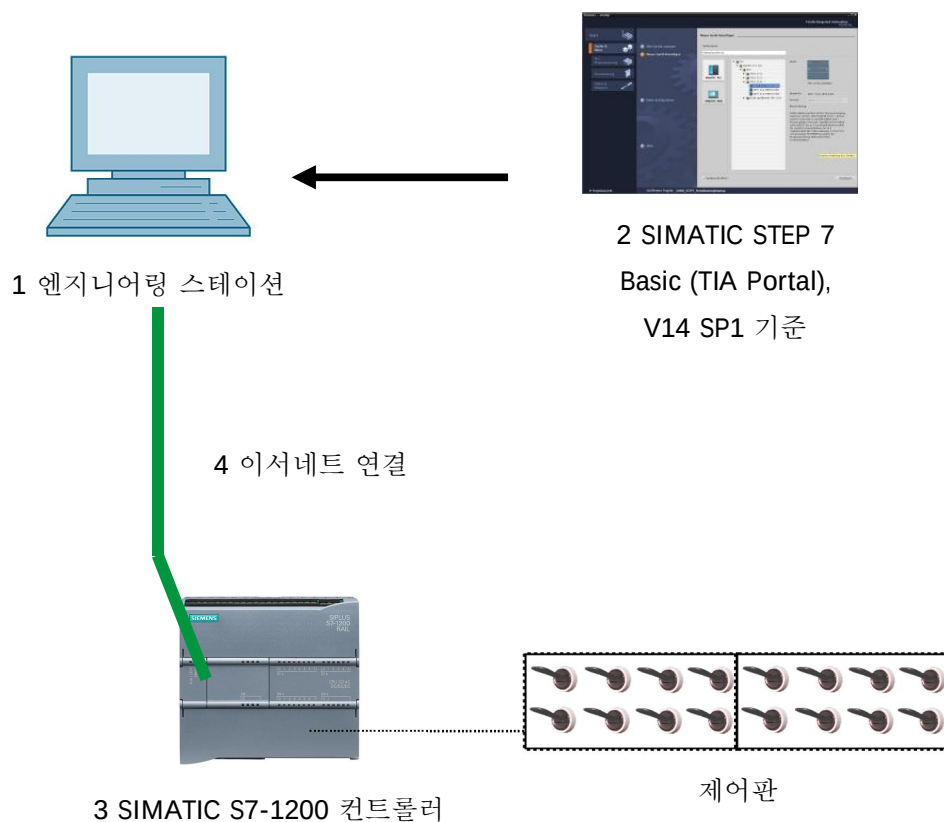
이 장은 SIMATIC S7 CPU1214C의 하드웨어 컨피규레이션을 기반으로 합니다. 그러나 디지털 입력 및 출력 카드가 포함된 다른 하드웨어 컨피규레이션을 사용할 수 있습니다. 이 장의 경우 다음 프로젝트를 사용할 수 있습니다. 예:

SCE_EN_011_101_Hardware_Configuration_CPU1214C.zap14

3 필수 하드웨어와 소프트웨어

- 1 엔지니어링 스테이션: 전제조건에는 하드웨어 및 운영 체제가 포함됩니다.
(추가 정보는 TIA Portal 설치 DVD 에 추가 정보 참조).
- 2 TIA Portal 의 SIMATIC STEP 7 Basic 소프트웨어 – V14 SP1 기준
- 3 SIMATIC S7-1200 컨트롤러, 예: CPU 1214C DC/DC/DC 및 ANALOG OUTPUT SB1232
시그널 보드, 1 AO – V4.2.1 기준 펌웨어

참조: 디지털 입력 및 아날로그 입력과 출력은 제어판으로 전달되어야 합니다.
- 4 엔지니어링 스테이션과 컨트롤러 간의 이서네트 연결



4 이론

4.1 운영체제 및 애플리케이션 프로그램

모든 컨트롤러(CPU)에는 **운영체제**가 포함되어 있는데, CP 의 모든 기능과 시퀀스를 조직하는 역할을 합니다. 운영체제는 다음과 같은 작업을 수행합니다.

- 워م 리스타트 수행
- 입력 및 출력의 프로세스 이미지 업데이트
- 주기적 사용자 프로그램 호출
- 인터럽트 감지 및 인터럽트 OB 호출
- 오류 감지 및 처리
- 메모리 영역 관리

운영체제는 CPU 를 구성하는 핵심 요소로서 사전 설치가 되어 있습니다.

사용자 프로그램에는 특정한 자동화 작업을 실행하는 데 필요한 모든 기능들이 포함되어 있습니다. 사용자 프로그램은 다음과 같은 작업을 수행합니다.

- 스타트업 OB 를 이용한 워م 리스타트를 위해 기본적인 요구사항을 확인
- 프로세스 데이터 처리, 즉 입력 신호 상태의 한 기능으로서 출력 신호 활성화
- 인터럽트 및 인터럽트 입력에 반응
- 프로그램이 정상 실행되는 동안 오류 처리

4.2 오거나이제이션 블록

오거나이제이션 블록(OB)은 컨트롤러(CPU)의 운영체제와 애플리케이션 프로그램 사이의 인터페이스를 형성합니다. 이들 블록은 운영체제에서 호출이 되어 다음과 같은 작업들을 제어합니다.

- 주기적 프로그램 처리(예: OB1)
- 컨트롤러의 스타트업 특성
- 인터럽트 중심(interrupt-driven)의 프로그램 처리
- 오류 처리

프로젝트는 **주기적 프로그램 처리를 위한 최소한 하나의 오거나이제이션 블록**은 가져야 합니다. 그림 1 에서와 같이 **시작 이벤트**에 의해 OB 가 호출됩니다. 뿐만 아니라 각각의 OB 에는 우선순위가 정의되어 있는데, 예를 들어 오류 처리를 위한 OB82 는 주기적 OB1 을 인터럽트 할 수 있습니다.

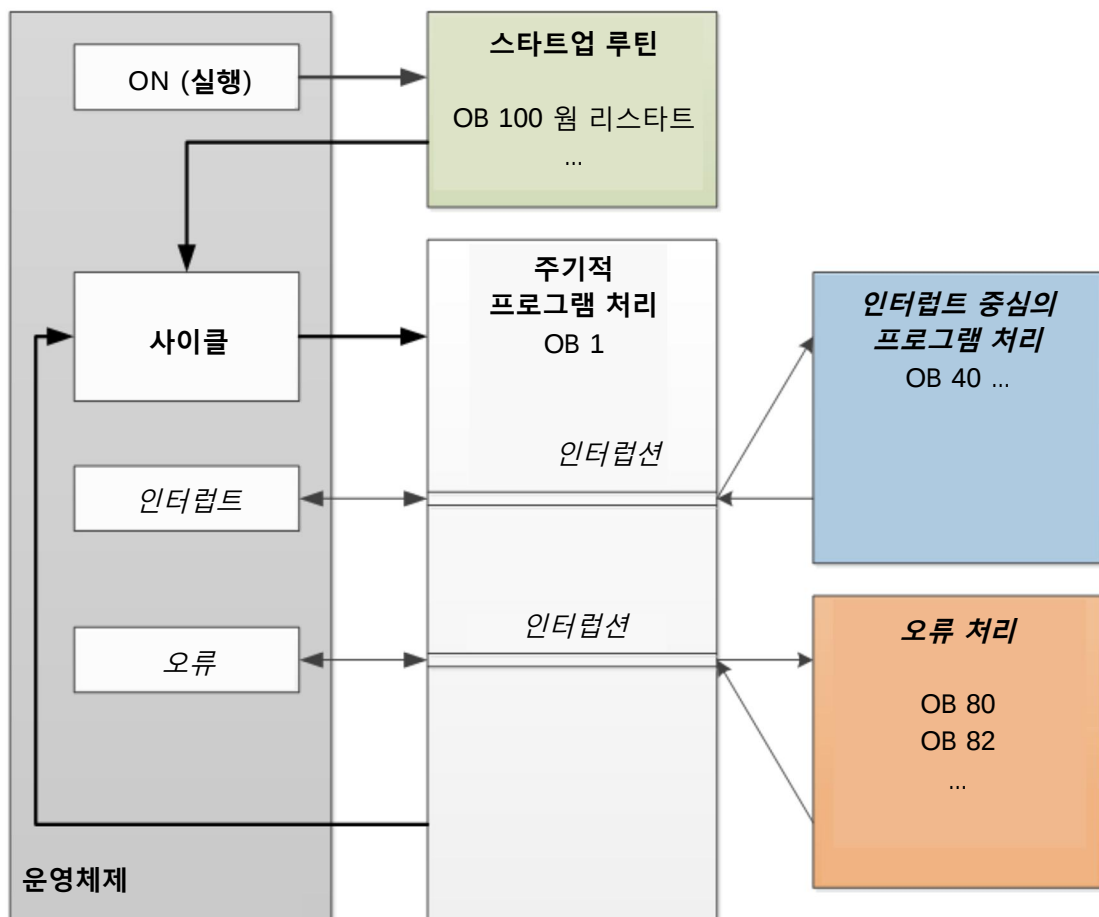


그림 1: 운영체제 및 OB 호출에서의 시작 이벤트

시작 이벤트가 발생하면 다음과 같이 반응할 수 있습니다.

- OB 가 이벤트에 지정이 된 경우에는 이 이벤트가 지정된 OB 의 실행을 개시합니다. 지정된 OB 의 우선순위가 현재 실행된 OB 의 우선순위보다 높으면 이 OB 가 즉시 실행됩니다(인터럽트). 그렇지 않은 경우에는 우선순위가 더 높은 OB 가 끝까지 실행이 될 때까지 지정된 OB 가 대기합니다.
- OB 가 이벤트에 지정되지 않은 경우에는 기본 설정된 시스템 반응이 수행됩니다.

표 1 에는 SIMATIC S7-1200 에 대한 다양한 시작 이벤트의 예가 나와 있습니다. 또한 가능한 OB 번호 및 각각의 오거나이제이션 블록(OB)이 컨트롤러에 존재하지 않은 경우 나타나는 기본 시스템 반응을 볼 수 있습니다.

시작 이벤트	가능한 OB 번호	기본 시스템 반응
스타트업	100, ³ 123	무시
주기적 프로그램	1, ³ 123	무시
TOD(Time-Of-Day) 인터럽트	10 ~ 11	-
업데이트 인터럽트	56	무시
1 번 초과된 주기적 모니터링 시간 스캔	80	무시
2 번 초과된 주기적 모니터링 시간 스캔	80	정지
진단 인터럽트	82	무시

표 1: 다양한 시작 이벤트에 대한 OB 번호

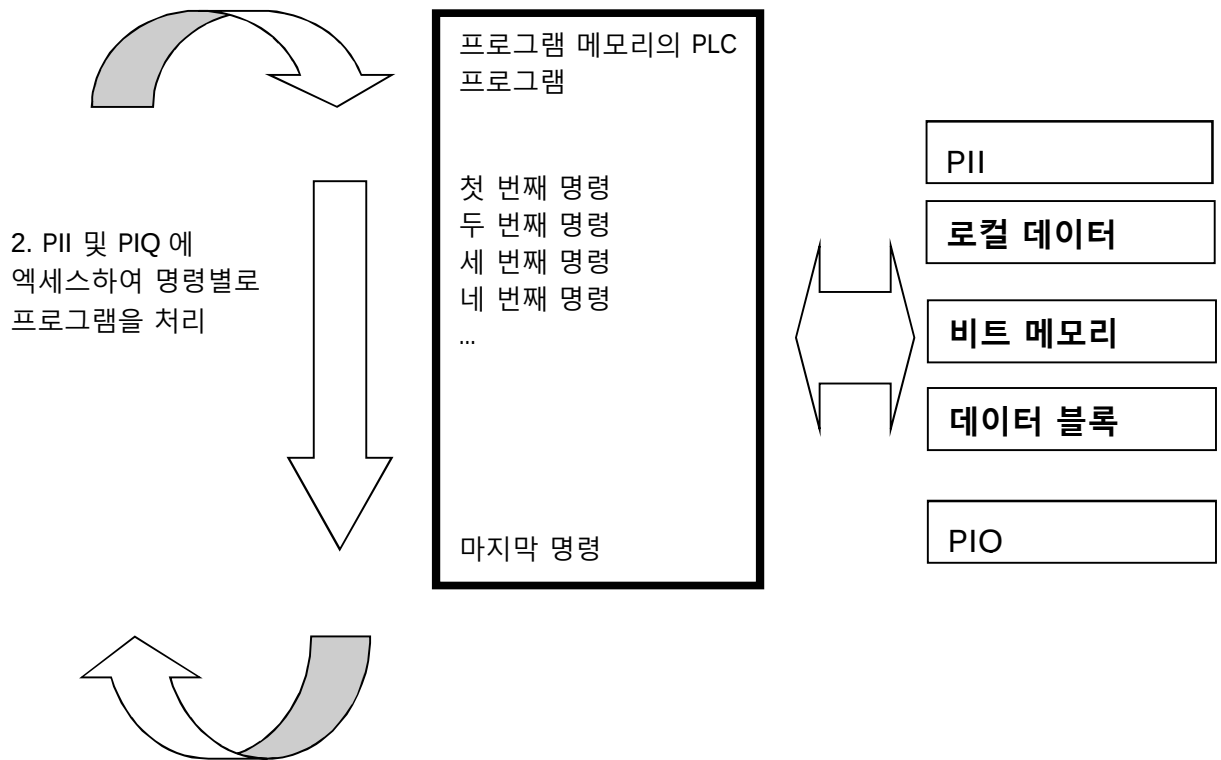
4.3 프로세스 이미지 및 주기적 프로그램 처리

주기적 사용자 프로그램은 입력(I) 및 출력(O) 주소를 지정할 때 입력/출력 모듈에서 직접 신호 상태를 조회하지 않습니다. 그 대신 CPU 의 메모리 영역을 액세스합니다. 이러한 메모리 영역에는 신호 상태의 이미지가 포함되어 있는데, 이를 **프로세스 이미지**라고 합니다.

주기적 프로그램의 처리 순서는 다음과 같습니다.

1. 주기적 프로그램이 시작되면 각각의 입력에 전압이 인가되었는지 아닌지를 확인하기 위한 쿼리가 전송됩니다. 이러한 입력의 상태는 **입력의 프로세스 이미지(PII)**에 저장됩니다. 이 과정에서 전압이 인가된 입력에 대해서는 정보 **1** 또는 “높음”이 저장되고 무전압 상태의 입력에 대해서는 정보 **0** 또는 “낮음”이 저장됩니다.
2. 그런 다음 CPU 는 주기적 오거나이제이션 블록에 저장된 프로그램을 실행합니다. CPU 는 이전에 읽어 들인 **입력의 프로세스 이미지(PII)**를 액세스하여 필요한 입력 정보를 얻고, 논리 연산의 결과(RLO)가 소위 **출력의 프로세스 이미지(PIQ)**에 기록됩니다.
3. 사이클이 끝나면 **출력의 프로세스 이미지(PIQ)**가 신호 상태로서 출력 모듈로 전달되고, 전원이 인가되거나 끊깁니다. 이제 항목 1 부터 다시 시퀀스가 계속 진행됩니다.

1. PII 에서 입력 상태를 저장



3. PIQ 에서 출력으로 상태 전달

그림 2: 주기적 프로그램 처리

참고: 이 시퀀스를 위해 CPU 에서 필요로 하는 시간을 **사이클 시간**이라고 합니다. 사이클 시간은 순서대로 명령의 수와 유형, 그리고 컨트롤러의 프로세서 성능에 따라 결정됩니다.

4.4 평선

평선(FC)은 메모리가 없는 로직 블록입니다. 이들은 블록 파라미터의 값들을 저장할 수 있는 **데이터 메모리가 없습니다**. 따라서, 평선이 호출될 때 모든 인터페이스 파라미터를 연결해 주어야 합니다. 데이터를 영구적으로 저장하려면 먼저 글로벌 데이터 블록을 미리 생성해야 합니다.

하나의 평선에는 또 다른 로직 블록에서 평선이 호출될 때마다 실행되는 하나의 프로그램이 포함되어 있습니다.

평선은 예를 들어 다음과 같은 용도로 사용할 수 있습니다.

- 수학 평선 - 입력 값에 따라 결과를 반환
- 기술 평선 - 바이너리 논리 연산을 통한 개별 제어 등의 평선

또한 평선은 한 프로그램 내의 서로 다른 위치에서 여러 차례 호출이 가능합니다.

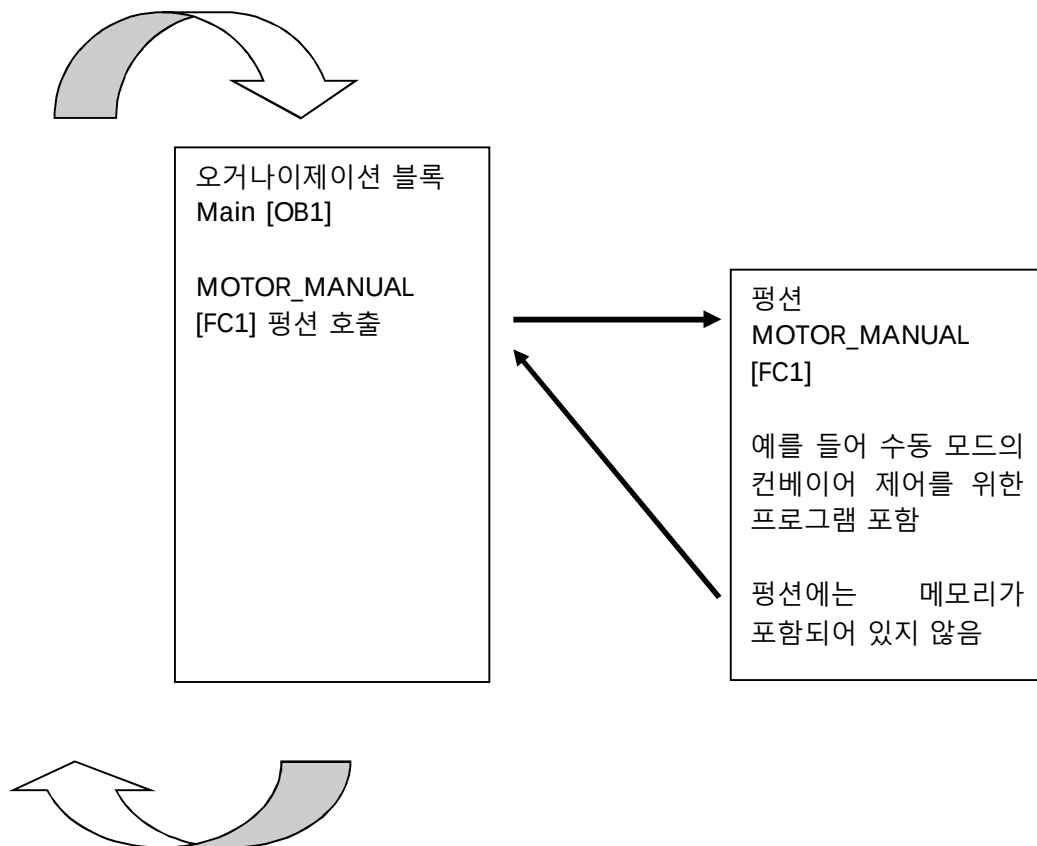


그림 3: 오거나이제이션 블록 Main [OB1]에서 호출되는 평선

4.5 평선 블록 및 인스턴스 데이터 블록

평선 블록은 입력, 출력 및 입력-출력 태그를 비롯해 정적 태그를 인스턴스 데이터 블록에 영구적으로 저장하는 로직 블록이기 때문에 **블록이 실행된 이후에도 사용이 가능합니다**. 이러한 이유로 평선 블록은 “메모리”를 갖춘 블록으로 불리기도 합니다.

평선 블록은 임시 태그에서도 작동이 가능합니다. 그러나 임시 태그는 인스턴스 DB에 저장되지 않습니다. 대신에 한 사이클 동안에만 사용이 가능합니다.

평선 블록은 다음과 같은 평선을 통해 구현할 수 없는 작업에 사용됩니다.

- 블록에서 타이머와 카운터가 필요한 모든 경우
- 예를 들어 버튼을 통한 운전 모드의 사전 선택과 같이 프로그램에 정보를 저장해야 하는 경우

평선 블록은 다른 로직 블록에서 호출이 될 때마다 항상 실행됩니다. 또한 평선 블록은 한 프로그램 내의 서로 다른 위치에서 여러 차례 호출이 가능합니다. 이는 자주 반복되는 복잡 평선의 프로그래밍을 용이하게 합니다.

평선 블록의 호출을 인스턴스라고 합니다. 평선 블록의 각 인스턴스에는 평선 블록이 사용하는 데이터가 포함된 메모리 영역이 할당되어 있습니다. 이 메모리는 소프트웨어에 의해 자동으로 생성되는 데이터 블록을 통해 사용할 수 있게 됩니다.

또한, **멀티 인스턴스** 형태로 하나의 데이터 블록 내에 여러 개의 인스턴스가 가능하도록 메모리를 제공하는 것이 가능합니다. 인스턴스 데이터 블록의 최대 크기는 CPU에 따라 다릅니다. 평선 블록에서 선언된 태그가 인스턴스 데이터 블록의 구조를 결정합니다.

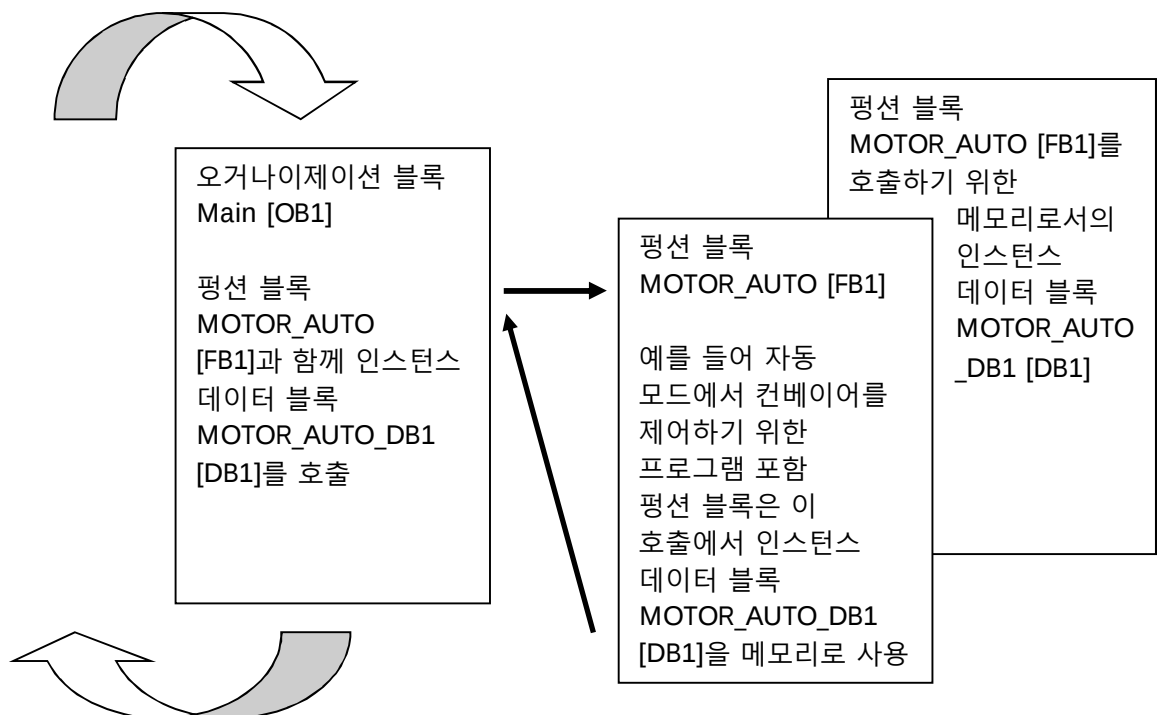


그림 4: 오거나이제이션 블록 Main [OB1]에서 호출된 평선 블록 및 인스턴스

4.6 글로벌 데이터 블록

로직 블록과 대조적으로 데이터 블록에는 명령어가 포함되어 있지 않습니다. 그보다는 사용자 데이터를 위한 메모리 역할을 합니다.

따라서 데이터 블록에는 사용자 프로그램에서 사용되는 데이터가 포함됩니다. 필요에 따라 글로벌 데이터 블록의 구조를 정의할 수 있습니다.

글로벌 데이터 블록에는 **다른 모든 블록들에서** 사용이 가능한 데이터를 저장합니다(그림 5 참조). 인스턴스 데이터 블록에는 관련된 평선 블록만 액세스 할 수 있습니다. 데이터 블록의 최대 크기는 CPU에 따라 다릅니다.

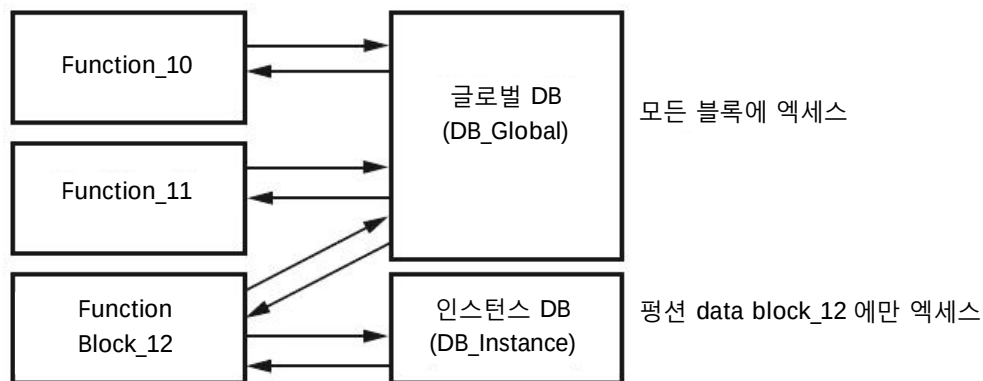


그림 5: 글로벌 DB와 인스턴스 DB 간의 차이

글로벌 데이터 블록의 애플리케이션 예는 다음과 같습니다.

- 스토리지 시스템에 대한 정보 저장 “어떤 제품이 어디에 위치해 있는가?”
- 특정 제품에 대한 레시피 저장

4.7 라이브러리 호환 로직 블록

사용자 프로그램은 선형 또는 구조적 프로그래밍을 통해 생성할 수 있습니다. **선형 프로그래밍**은 사이클 OB 에서 전체 사용자 프로그램을 기록하지만, LOGO! 같이 보다 저비용 제어 시스템을 사용하는 단순한 프로그램에만 적합합니다.

보다 복잡한 프로그램에는 항상 **구조적 프로그래밍**을 권장합니다. 여기에서, 평선과 평선 블록에서의 솔루션 구현을 위해 전체 자동화 작업을 작은 작업단위로 나눌 수 있습니다..

이 경우에는 라이브러리 호환 로직 블록을 우선적으로 생성해야 합니다. 즉, 이것은 평선 또는 평선 블록의 입력 및 출력 파라미터가 일반적으로 정의되었고, 블록이 사용될 때 현재의 글로벌 태그(입력/출력)를 통해서만 제공된다는 것을 의미합니다.

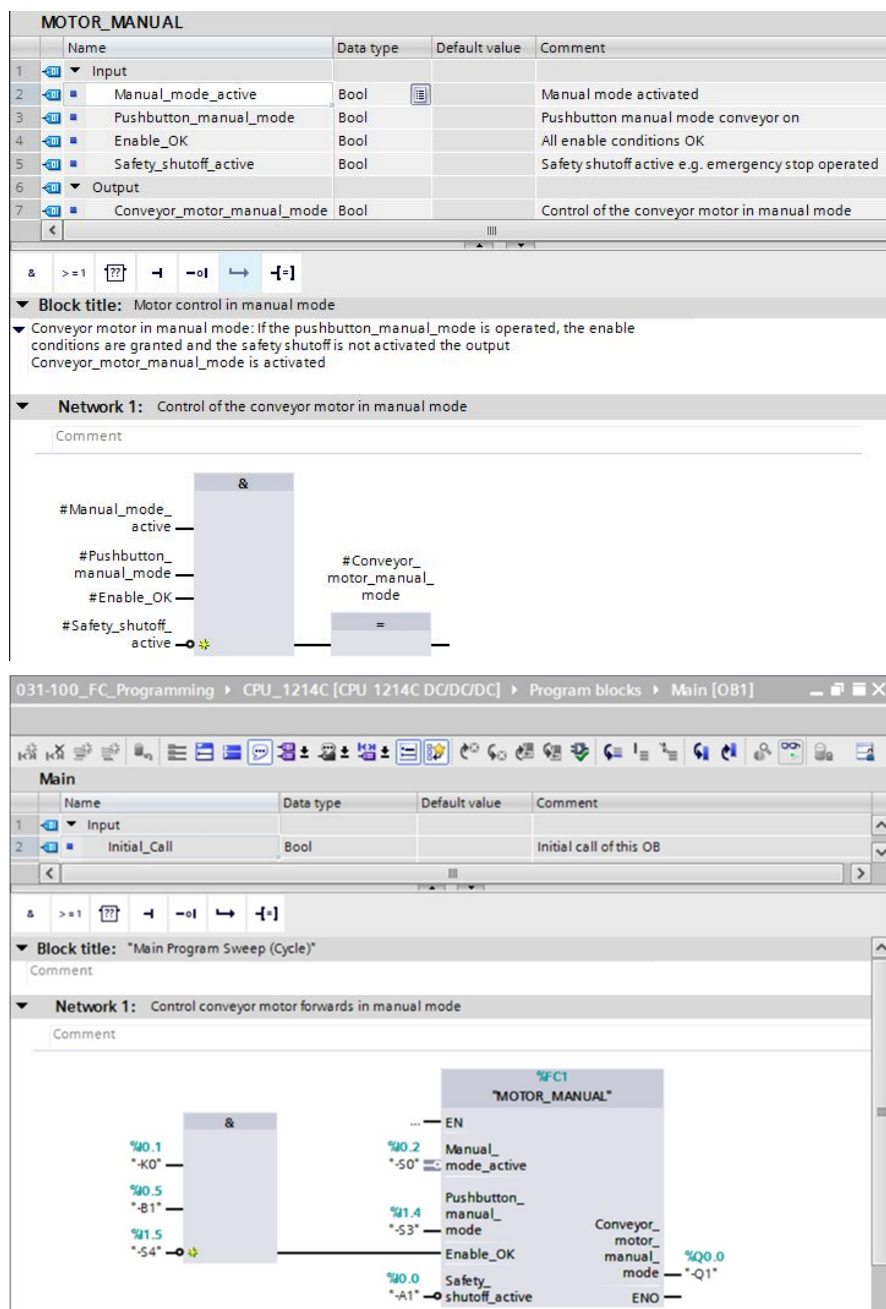


그림 6: OB1 에서 호출된 라이브러리 호환 평선

4.8 프로그래밍 언어

SIMATIC S7-1200 의 경우, 평선 및 평선 블록을 프로그래밍하는 데 사용할 수 있는 프로그래밍 언어로는 평선 블록 다이어그램(Function Block Diagram, FBD), 래더 로직(Ladder Logic, LAD), 구조적 제어 언어(Structured Control Language, SCL)가 있습니다.

평선 블록 다이어그램(FBD) 프로그래밍 언어는 다음과 같습니다.

FBD 는 그래픽 프로그래밍 언어입니다. 전자 스위칭 시스템을 토대로 표현이 됩니다. 프로그램은 네트워크에 매핑됩니다. 네트워크에는 1 개 이상의 논리 연산 경로가 포함되어 있습니다. 바이너리 및 아날로그 신호는 박스로 연결이 됩니다. 불(Bool) 대수로부터 파악된 그래픽 로직 심볼은 바이너리 로직을 표현하는 데 사용됩니다.

바이너리 평선을 이용해 바이너리 오퍼랜드에 대해 쿼리를 수행하고 이들의 신호 상태를 논리적으로 결합할 수 있습니다. 바이너리 평선의 예로는 “AND 연산”, “OR 연산” 및 “EXCLUSIVE OR 연산”과 같은 명령이 있습니다. 이들은 그림 7 에 나와 있습니다.

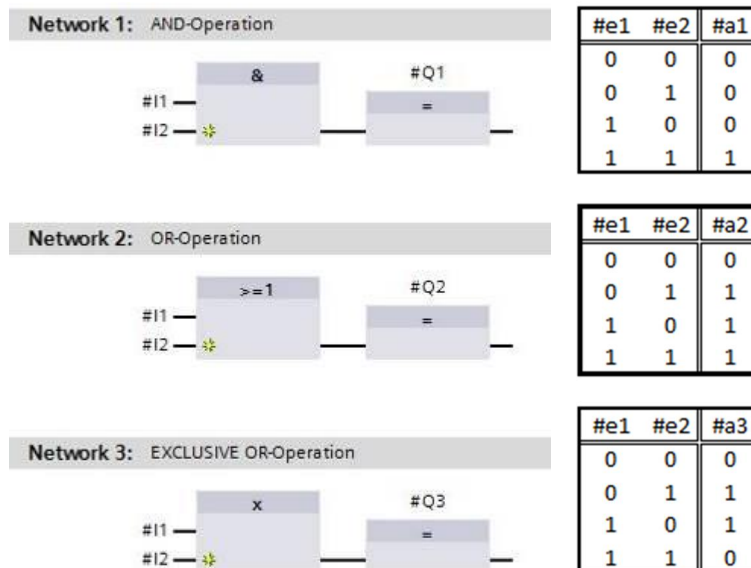


그림 7: FBD 의 바이너리 함수 및 관련 로직 테이블

그렇게 해서 당신은 프로그램에서 바이너리 출력을 제어하고 엣지를 평가하며 점프 기능을 실행하는 것과 같은 간단한 명령을 사용할 수 있습니다.

IEC 타이머 및 IEC 카운터 같은 프로그램 요소들은 복합 명령을 제공합니다.

빈 박스는 필요한 명령을 선택할 수 있는 플레이스홀더 역할을 합니다.

입력 활성화 EN (enable)/출력 활성화 ENO (enable output) 메커니즘:

- EN/ENO 메커니즘이 없는 명령은 박스 입력의 신호 상태에 무관하게 실행
- EN/ENO 메커니즘이 있는 명령은 활성화 입력 “EN” 입력의 신호 상태가 “1”인 경우에만 실행됩니다. 박스가 올바르게 처리되면 활성화 출력 “ENO”의 신호 상태가 “1”이 됩니다. 처리 동안 오류가 발생하면 활성화 출력 “ENO”가 리셋됩니다. 활성화 입력 EN 이 연결되지 않은 경우에는 항상 박스가 실행됩니다.

5 과제

이 챕터에서는 다음과 같이 선별기 프로세스 설명에 대한 평선들을 계획, 프로그래밍 및 테스트해보겠습니다.

- 수동 모드 - 수동/조그 모드에서 컨베이어 전진 트래킹 제어

6 계획 수립

명확성 및 재사용성을 이유로 **OB1** 내에서 모든 평선을 프로그래밍하는 것은 권장하지 않습니다. 따라서 프로그램 코드의 대다수는 평선(FC) 및 평선 블록(FB)으로 이동됩니다. FC 로 이동시킬 평선과 **OB1** 에서 실행할 평선은 아래와 같이 결정이 됩니다.

6.1 비상 정지(EMERGENCY STOP)

비상 정지를 위해서 별도의 평선이 필요하지 않습니다. 마치 운전 모드처럼 비상 정지 릴레이의 현재 상태를 블록에서 직접 사용할 수 있습니다.

6.2 수동 모드 - 수동 모드의 컨베이어 모터

컨베이어 모터의 수동 모드는 평선(FC) “**MOTOR_MANUAL**”에 캡슐화가 됩니다. 한편으로 캡슐화는 **OB1** 의 명확성을 지켜줍니다. 다른 한편으로는 또 다른 컨베이어 벨트가 스테이션에 추가된 경우에 재사용이 가능하도록 합니다. 표 2 에는 계획된 파라미터들이 나와 있습니다.

입력	데이터 유형	코멘트
Manual_mode_active	BOOL	수동 모드 활성화
Pushbutton_manual_mode	BOOL	수동 모드의 컨베이어 “ON” 전환 푸시버튼
Enable_OK	BOOL	모든 활성화 조건 확인
Safety_shutoff_active	BOOL	안전 전원 차단 활성화(예: 비상 정지 작동)
출력		
Conveyor_motor_manual_mode	BOOL	수동 모드의 컨베이어 모터 제어

표 2: FC “**MOTOR_MANUAL**”을 위한 파라미터

Pushbutton_manual_mode 를 누르고, 수동 모드가 활성화되고, 활성화 조건이 충족되고 안전 차단이 활성화되지 않았을 때 출력 **Conveyor_motor_manual_mode** 가 “ON” 됩니다.

6.3 기술 다이어그램

여기에는, 과제에 대한 기술 다이어그램이 나와 있습니다.

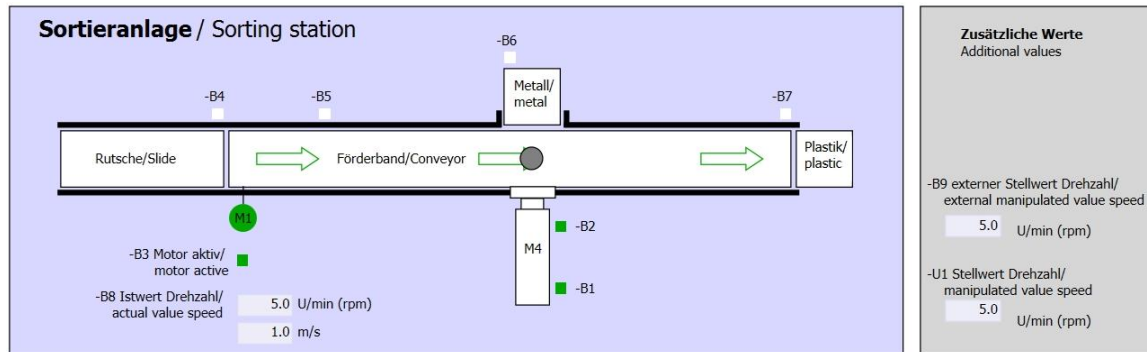


그림 8: 기술 다이어그램



그림 9: 제어 패널

6.4 참조 목록

이 과제를 위한 오퍼랜드로서 아래와 같은 신호들이 필요합니다.

DI	유형	식별자	평선	NC/NO
I 0.0	BOOL	-A1	반환 신호 비상 정지 확인	NC
I 0.1	BOOL	-K0	메인 스위치 "ON"	NO
I 0.2	BOOL	-S0	모드 선택 수동(0)/자동(1)	수동 = 0 자동 = 1
I 0.5	BOOL	-B1	센서 실린더 M4 복귀	NO
I 1.4	BOOL	-S3	푸시버튼 수동 모드 컨베이어 M1 전진	NO
I 1.5	BOOL	-S4	푸시버튼 수동 모드 컨베이어 M1 후진	NO

DO	유형	식별자	평선	
Q 0.0	BOOL	-Q1	컨베이어 모터 M1 고정 속도로 전진	

참조 목록 범위

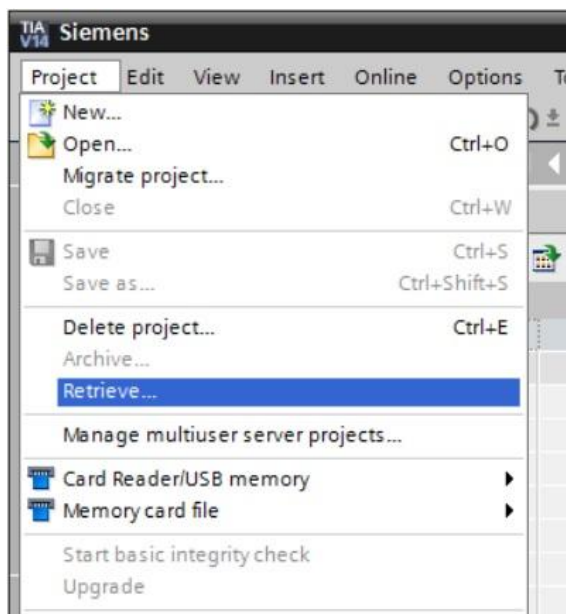
DI	디지털 입력	DO	디지털 출력
AI	아날로그 입력	AO	아날로그 출력
I	입력	Q	출력
NC	상시 닫힘		
NO	상시 열림		

7 단계별 따라 해보기

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

7.1 기존 프로젝트 압축 풀기

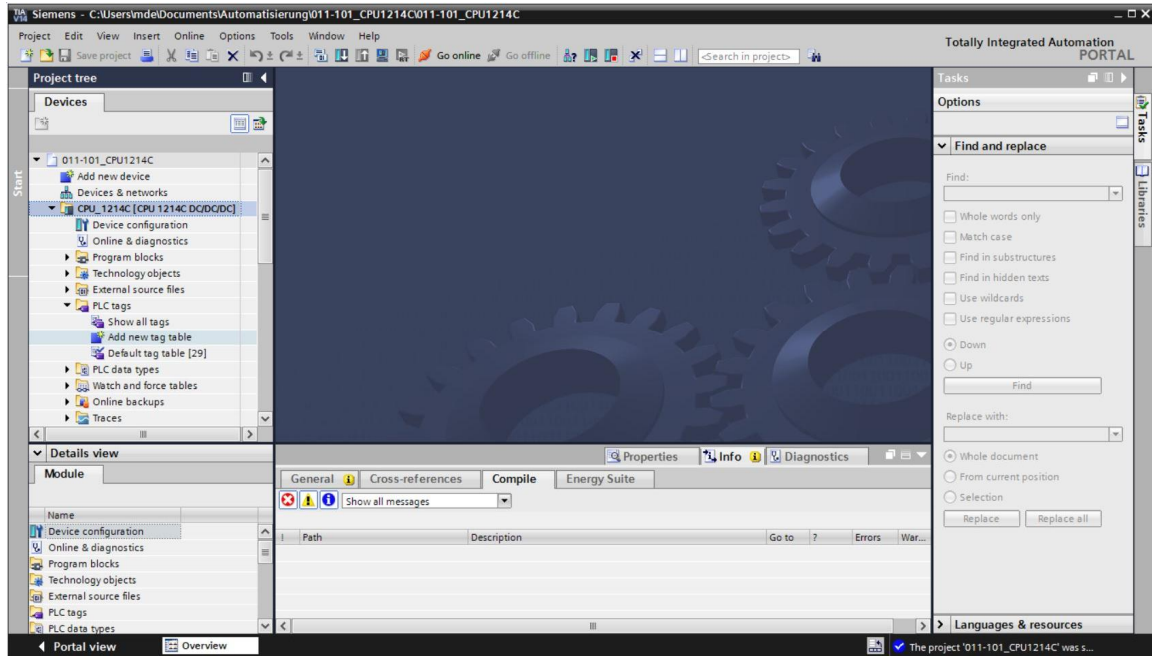
- ⑧ 평선(FC) “MOTOR_MANUAL” 프로그래밍을 시작하기 전에 하드웨어 구성(예: SCE_EN_011_101_Hardware_Configuration_CPU1214C.zap14)을 가진 프로젝트가 필요합니다. 아카이브된 기존 프로젝트의 압축을 풀려면 프로젝트 뷰에서 ⑧ “프로젝트”의 ⑧ “Retrieve”로 가서 해당되는 아카이브를 선택해야 합니다. “Open”를 클릭해 선택을 확정합니다(⑧ Project ⑧ Retrieve ⑧ .zap 아카이브 선택 ⑧ Open).



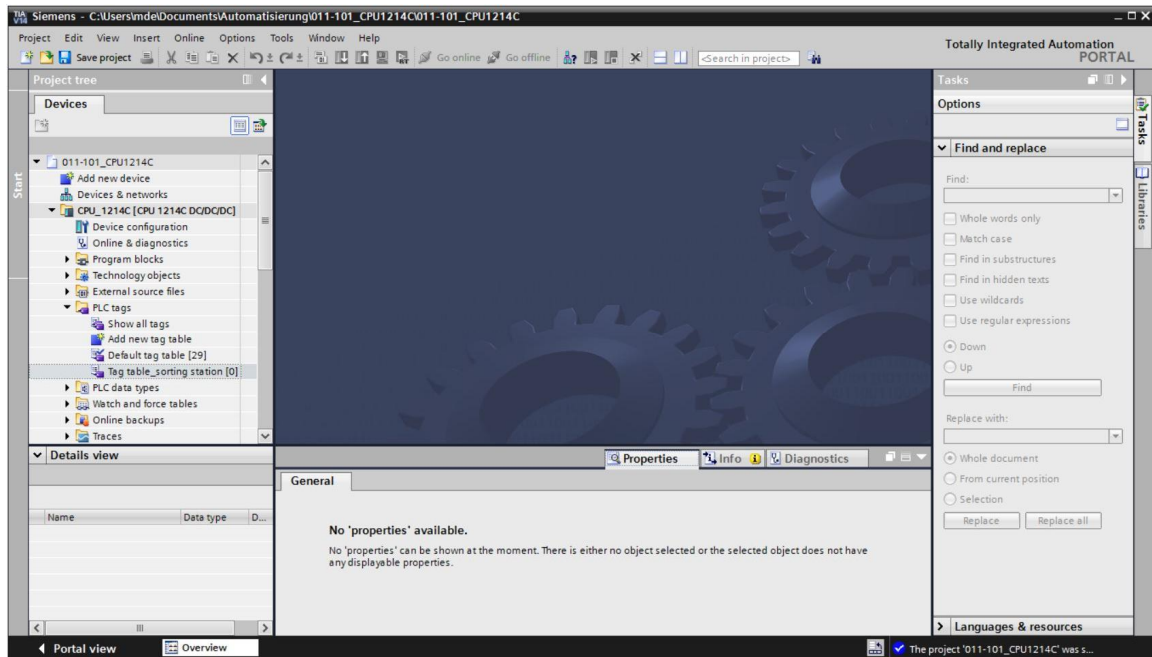
- ⑧ 그 다음으로 압축 풀기한 프로젝트가 저장될 대상 디렉토리를 선택합니다. “OK”을 클릭해 선택을 확정합니다. (⑧ 대상 디렉토리 ⑧ OK)

7.2 새 태그 테이블 생성

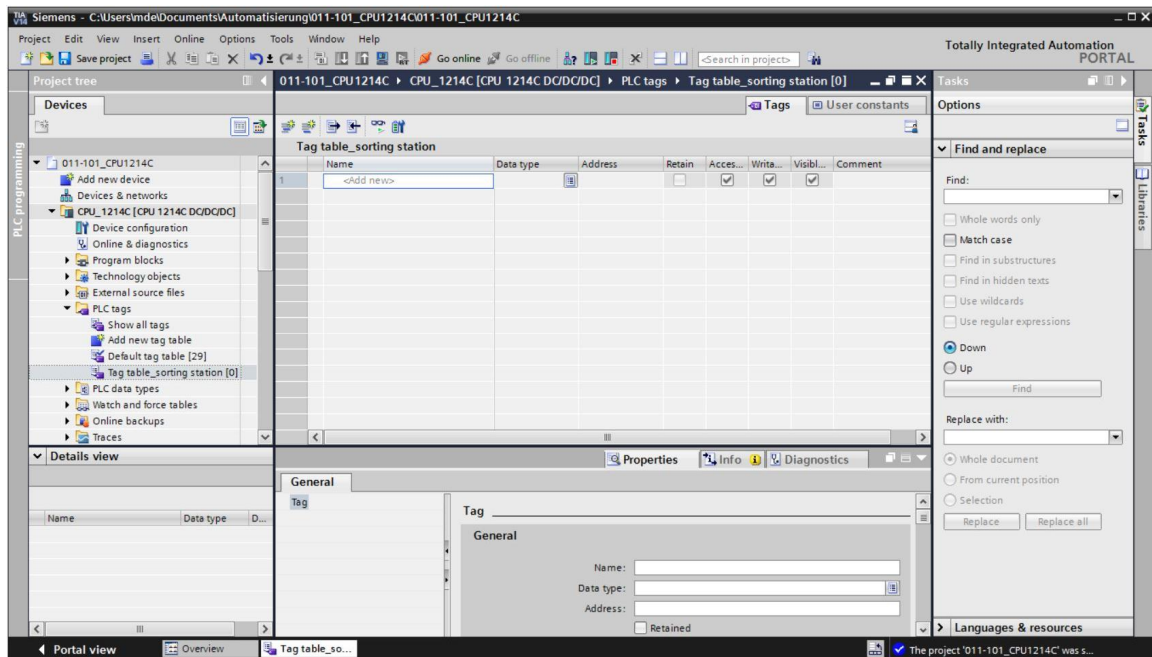
⑧ 프로젝트 뷰에서 ⑧ 컨트롤러의 PLC 태그를 탐색하고 ⑧ “Add new tag table”를 더블 클릭하여 새 태그 테이블을 생성합니다.



- ® 생성된 태그 테이블 이름을 “Tag_table_sorting_station”으로 변경합니다. (® “Tag_table_1”을 마우스 오른쪽 버튼으로 클릭 ® “Rename” ® Tag_table_sorting_station)

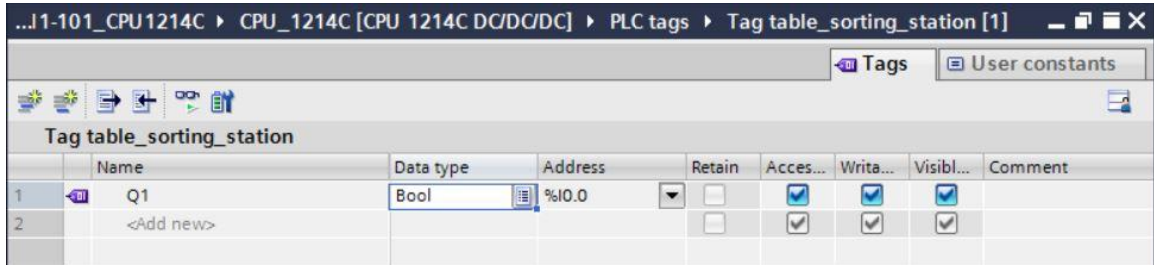


- ® 더블클릭으로 태그 테이블을 엽니다. (® Tag_table_sorting_station)

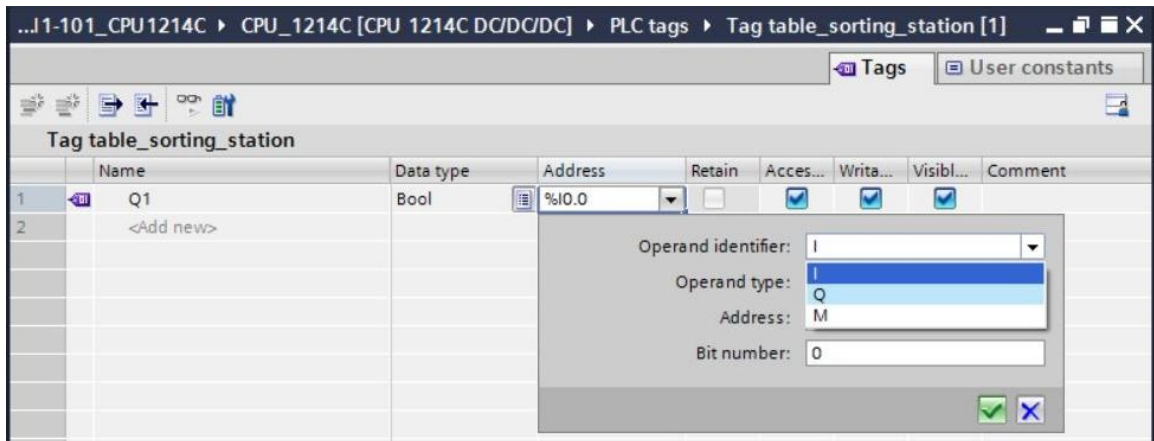


7.3 태그 테이블 내에서 새 태그 생성

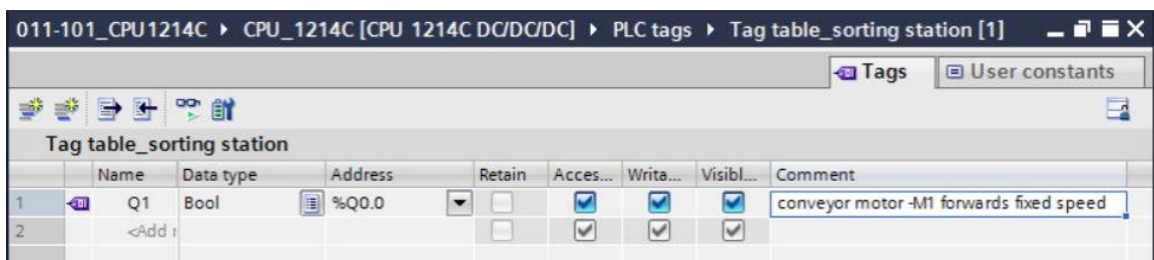
- ⑧ 이름 Q1 을 추가하고 엔터 키를 눌러 입력을 확정합니다. 추가 태그를 아직 생성하지 않은 경우에는 TIA Portal 이 데이터 유형 “Bool”과 주소 %I0.0(I 0.0)을 자동으로 할당합니다. (⑧ <추가> ⑧ Q1 ⑧ 엔터)



- ⑧ 이 값을 직접 입력하거나 드롭다운 화살표를 클릭해 “주소 지정” 메뉴를 열고 주소를 %Q0.0(Q 0.0)으로 변경합니다. 오퍼랜드 식별자를 Q 로 변경한 다음 엔터를 누르거나 체크 표시를 클릭해 이를 확정합니다. (⑧ %I0.0 ⑧ Operand identifier ⑧ Q ⑧ ☒)

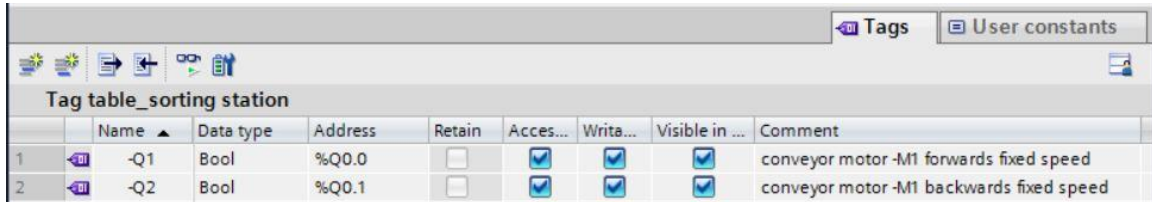


- ⑧ 태그에 대해 “컨베이어 모터 M1 고정 속도로 전진” 코멘트를 입력합니다.



- ⑧ 라인 2 에 새로운 Q2 태그를 추가합니다. TIA Portal 은 라인 1 과 동일한 데이터 유형을 자동으로 지정하고 %Q0.1(Q0.1)의 주소를 1 증가시킵니다. “컨베이어 모터 M1 후진 고정 속도” 코멘트를 입력합니다.

(⑧ <추가> ⑧ Q2 ⑧ 엔터 ⑧ 코멘트 ⑧ 컨베이어 모터 M1 후진 고정 속도)

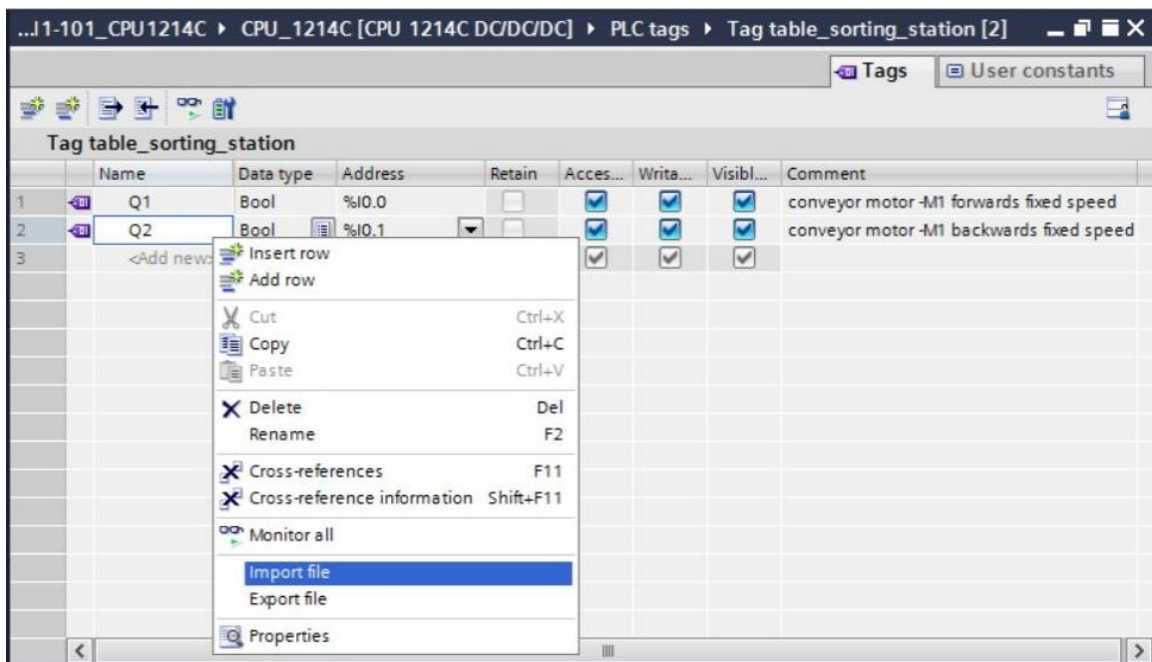


	Name	Data type	Address	Retain	Access...	Write...	Visible in ...	Comment
1	Q1	Bool	%Q0.0	☐	☒	☒	☒	conveyor motor -M1 forwards fixed speed
2	Q2	Bool	%Q0.1	☐	☒	☒	☒	conveyor motor -M1 backwards fixed speed

7.4 “Tag_table_sorting_station” 가져오기

- ⑧ 기존 심볼 테이블을 삽입하려면 생성된 “Tag_table_sorting_station”의 빈 필드를 마우스 오른쪽 버튼으로 클릭합니다. 바로 가기 메뉴에서 “파일 가져오기”를 선택합니다.

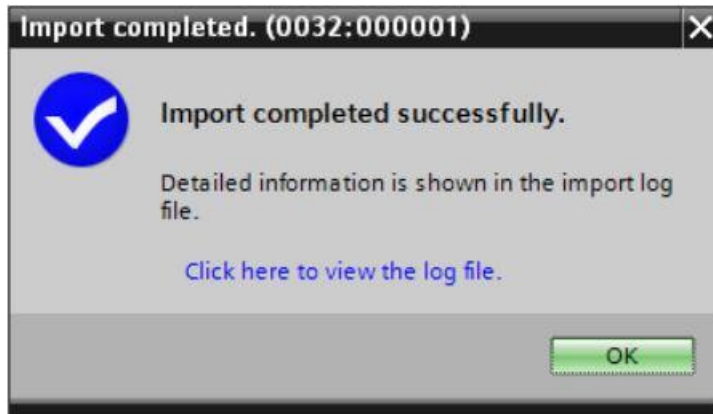
(⑧ 태그 테이블의 빈 필드에서 마우스 오른쪽 버튼 클릭 ⑧ Import file)



⑧ 원하는 심볼 테이블(예: .xlsx 형식)을 선택하고 “Open”를 클릭해 선택을 확정합니다.

(⑧ SCE_EN_020-100_Tag_table_sorting_station... ⑧ Open)

⑧ 가져오기가 끝나면 확인 창이 나타나고, 가져오기에 대한 로그 파일을 볼 수 있습니다. ⑧ “OK”을 클릭합니다.

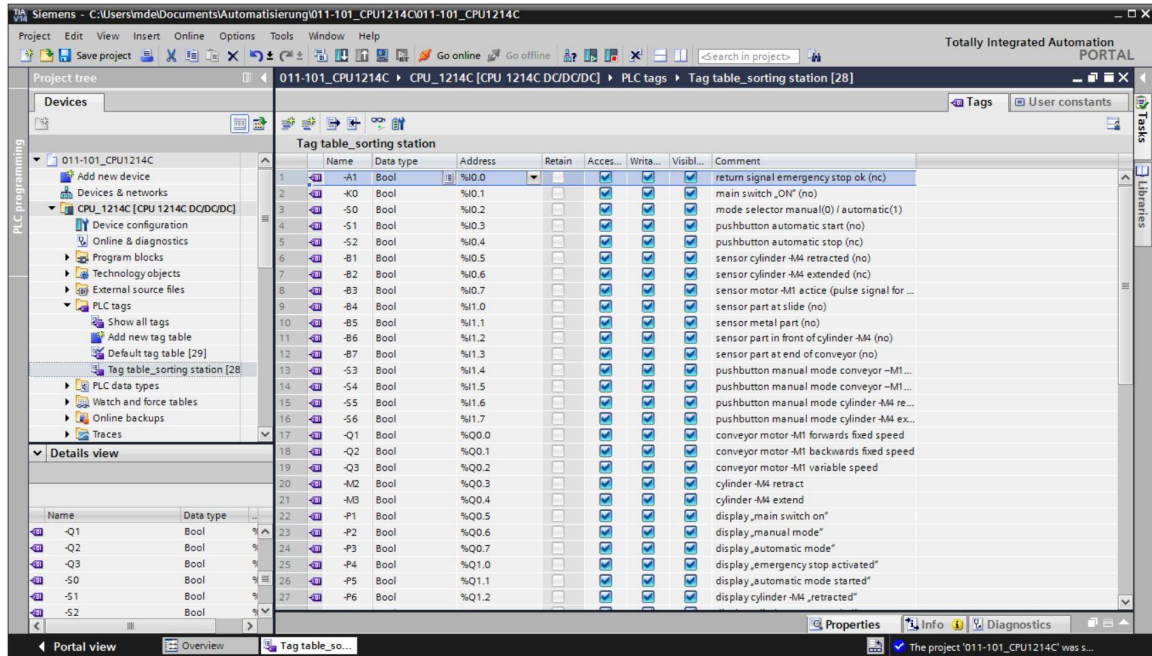


- ⑩ 일부 주소들이 주황색으로 강조 표시된 것을 볼 수 있습니다. 이들은 중복 주소로서, 혼란을 피하기 위해 연관된 태그의 이름에 자동으로 번호가 표시됩니다.
- ⑩ 라인을 선택하고 키보드에서 **Del** 키를 누르거나 바로 가기 메뉴에서 “Delete”를 선택하여 중복 태그를 삭제합니다. (⑩ 선택된 태그에서 마우스 오른쪽 버튼 클릭 ⑩ Delete)

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	Q1	Bool	%Q0.0		☒	☒	☒	conveyor motor -M1 forwards fixed speed
2	Q2	Bool	%Q0.1		☒	☒	☒	conveyor motor -M1 forwards fixed speed
3	-A1	Bool	%I0.0		☒	☒	☒	return signal emergency stop ok (nc)
4	-K0	Bool	%I0.1		☒	☒	☒	main switch „ON“ (no)
5	-S0	Bool	%I0.2		☒	☒	☒	mode selector manual(0) / automatic(1)
6	-S1	Bool	%I0.3		☒	☒	☒	pushbutton automatic start (no)
7	-S2	Bool	%I0.4		☒	☒	☒	pushbutton automatic stop (nc)
8	-B1	Bool	%I0.5		☒	☒	☒	sensor cylinder -M4 retracted (no)
9	-B2	Bool	%I0.6		☒	☒	☒	sensor cylinder -M4 extended (nc)
10	-B3	Bool	%I0.7		☒	☒	☒	sensor motor -M1 actice (pulse signal for ...
11	-B4	Bool	%I1.0		☒	☒	☒	sensor part at slide (no)
12	-B5	Bool	%I1.1		☒	☒	☒	sensor metal part (no)
13	-B6	Bool	%I1.2		☒	☒	☒	sensor part in front of cylinder -M4 (no)
14	-B7	Bool	%I1.3		☒	☒	☒	sensor part at end of conveyor (no)
15	-S3	Bool	%I1.4		☒	☒	☒	pushbutton manual mode conveyor -M1...
16	-S4	Bool	%I1.5		☒	☒	☒	pushbutton manual mode conveyor -M1...
17	-S5	Bool	%I1.6		☒	☒	☒	pushbutton manual mode cylinder -M4 re...
18	-S6	Bool	%I1.7		☒	☒	☒	pushbutton manual mode cylinder -M4 ex...
19	-Q1	Bool	%Q0.0		☒	☒	☒	conveyor motor -M1 forwards fixed speed
20	-Q2	Bool	%Q0.1		☒	☒	☒	conveyor motor -M1 backwards fixed speed
21	-Q3	Bool	%Q0.2		☒	☒	☒	conveyor motor -M1 variable speed
22	-M2	Bool	%Q0.3		☒	☒	☒	cylinder -M4 retract
23	-M3	Bool	%Q0.4		☒	☒	☒	cylinder -M4 extend
24	-P1	Bool	%Q0.5		☒	☒	☒	display „main switch on“
25	-P2	Bool	%Q0.6		☒	☒	☒	display „manual mode“
26	-P3	Bool	%Q0.7		☒	☒	☒	display „automatic mode“

® 디지털 입력 및 출력에 대한 전체 심볼 테이블이 나타납니다. 프로젝트를 031-100_FC Programming 라는 이름으로 저장합니다.

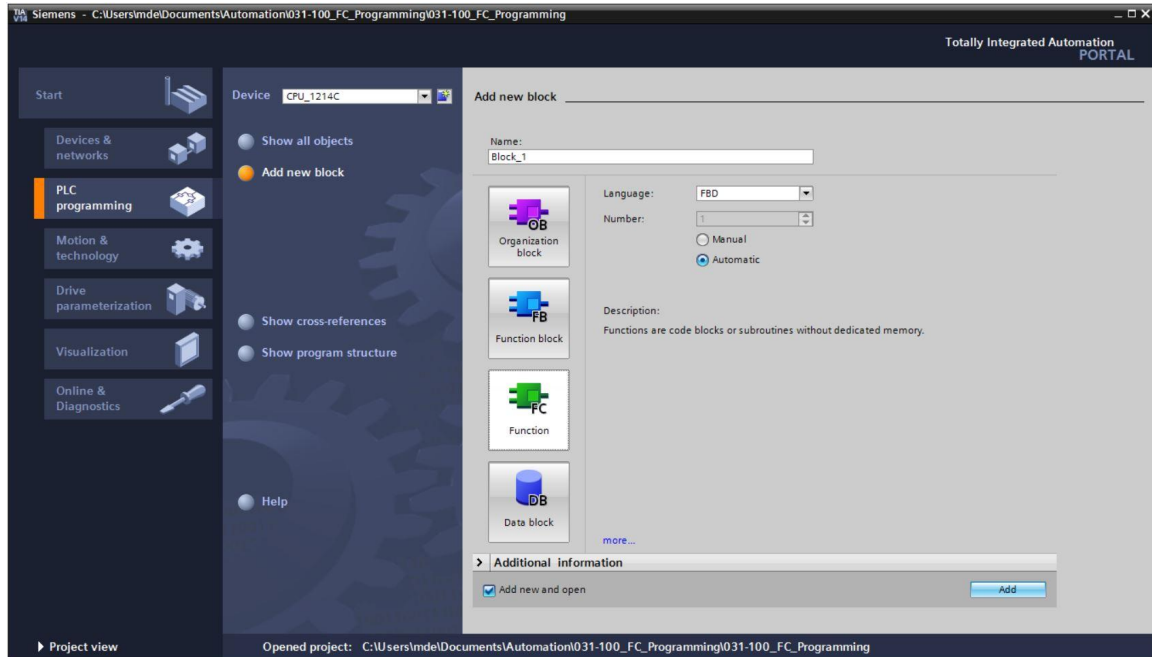
(® Project ® Save as.... ® 031-100_FC Programming ® Save)



7.5 수동 모드의 컨베이어 모터를 위한 평선 FC1 “MOTOR_MANUAL” 생성

④ 포털 뷰의 PLC 프로그래밍 섹션에서 “Add new block”를 클릭하여 새 평선을 생성합니다.

(④ PLC programming ④ Add new block ④ )



- ® 새 블록의 이름을 “MOTOR_MANUAL”로 변경하고 언어를 FBD 로 설정한 다음, 자동 지정된 번호를 그대로 유지합니다. “Add new and open” 체크박스를 선택합니다. 프로젝트 뷰에서 생성된 평선 블록으로 자동으로 이동됩니다. “Add”를 클릭합니다.

(® Name: MOTOR_MANUAL ® Language: FBD ® Number: Automatic ® ☒ Add new and open ® Add)

Add new block

Name:

Language:

Number:

☐ Manual
☒ Automatic

Description:
Functions are code blocks or subroutines without dedicated memory.

[more...](#)

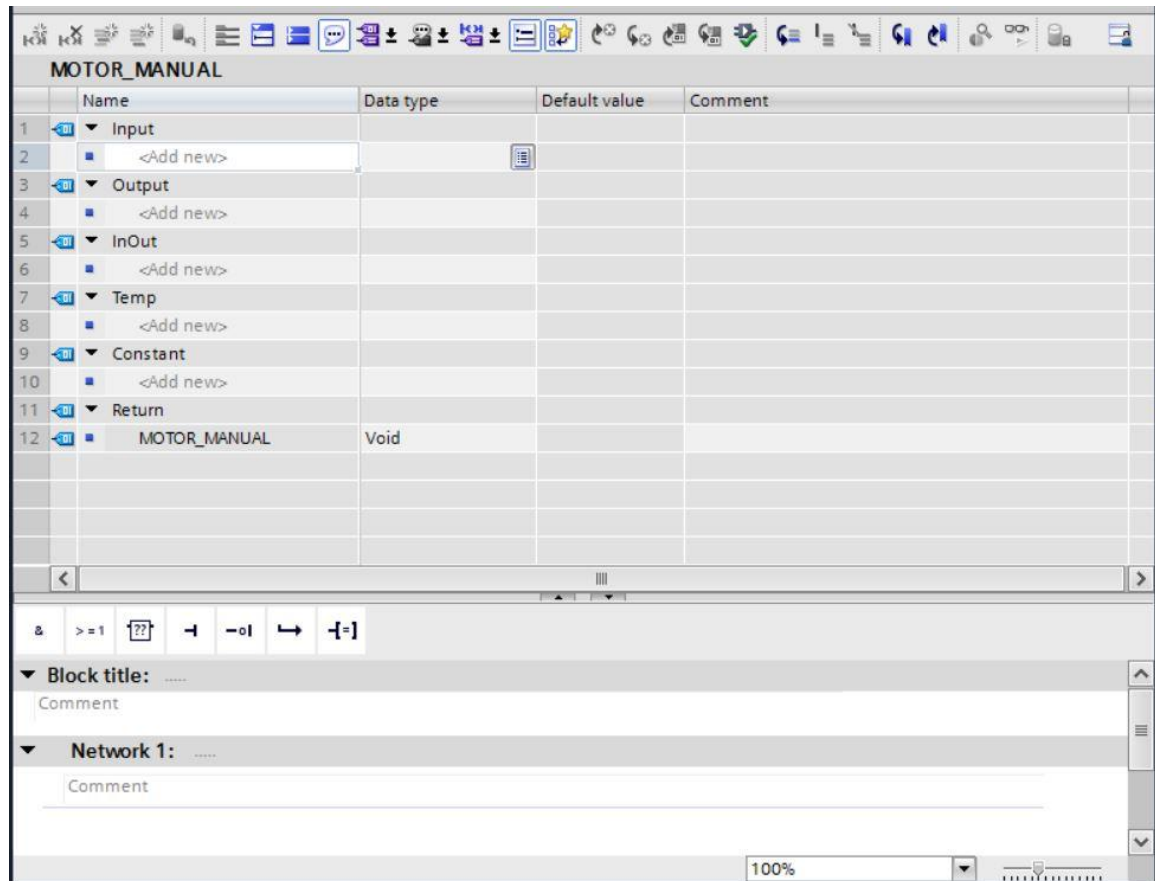
> **Additional information**

☒ Add new and open

7.6 평선 FC1 “MOTOR_MANUAL”에 대한 인터페이스 정의

“Add new and open”를 선택하면 추가한 블록을 생성할 수 있도록 창에서 프로젝트 뷰가 열립니다.

- ⑧ 프로그래밍 뷰의 상단 부분에 평선에 대한 인터페이스 목록(Interface description)이 표시됩니다.



- ⑧ 컨베이어 모터를 제어하려면 바이너리 출력 신호가 필요합니다. 이러한 이유로 먼저 “불(Bool)” 유형의 로컬 출력 태그 `#Conveyor_motor_manual_mode` 를 생성합니다. 이 파라미터의 코멘트로 “수동 모드의 컨베이어 모터 제어”를 입력합니다.

(⑧ Output: `Conveyor_motor_manual_mode` ⑧ Bool ⑧ 수동 모드의 컨베이어 모터 제어)

	Name	Data type	Default value	Comment
1	▼ Input			
2	■ <Add new>			
3	▼ Output			
4	■ Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode
5	■ <Add new>			
6	▼ InOut			
7	■ <Add new>			
8	▼ Temp			
9	■ <Add new>			
10	▼ Constant			
11	■ <Add new>			
12	▼ Return			
13	■ MOTOR_MANUAL	Void		

- ⑧ “입력”으로 가서 입력 인터페이스로서 파라미터 `#Manual_mode_active` 를 추가하고 엔터 키를 누르거나 입력 필드를 빠져나와서 입력을 확정합니다. 데이터 유형 “불(Bool)”이 자동으로 지정됩니다. 이를 그대로 유지합니다. 그 다음으로, 관련 코멘트 “수동 모드 활성화”를 입력합니다.

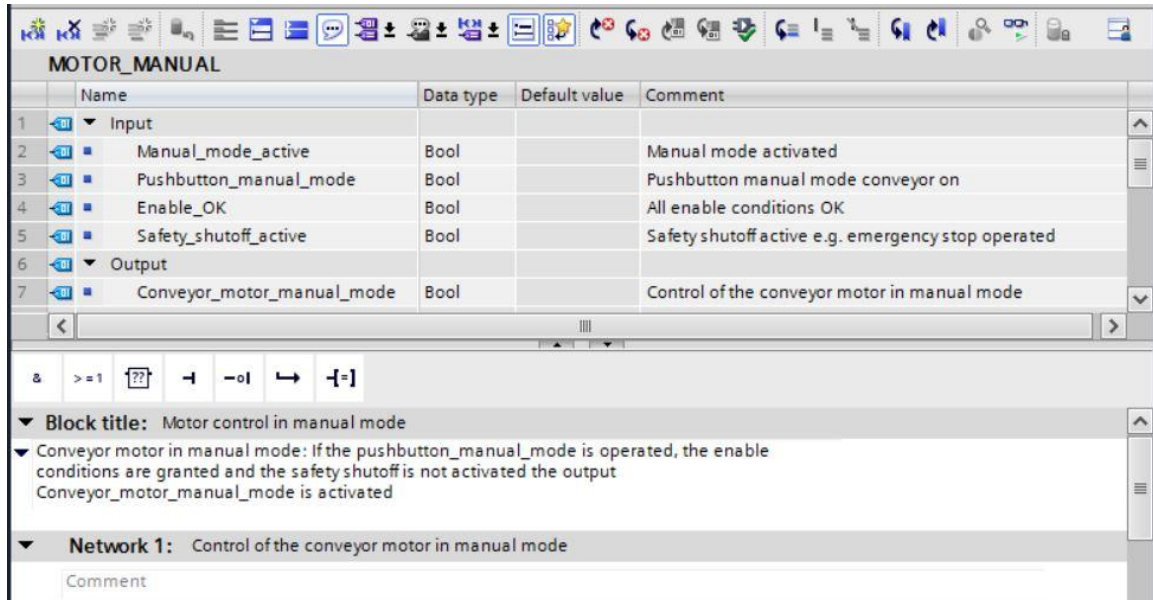
(⑧ `Manual_mode_active` ⑧ 엔터 ⑧ Bool ⑧ 수동 모드 활성화)

- ⑧ 계속해서 추가 바이너리 입력 파라미터로서 `#Pushbutton_manual_mode`, `#Enable_OK` 및 `#Safety_shutoff_active` 를 추가하고 이들의 데이터 유형을 확인합니다. 설명 코멘트를 추가합니다.

	Name	Data type	Defau...	Comment
1	▼ Input			
2	■ Manual_mode_active	Bool		Manual mode activated
3	■ Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on
4	■ Enable_OK	Bool		All enable conditions OK
5	■ Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operated
6	■ <Add new>			
7	▼ Output			
8	■ Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode
9	■ <Add new>			
10	▼ InOut			
11	■ <Add new>			
12	▼ Temp			
13	■ <Add new>			
14	▼ Constant			
15	■ <Add new>			
16	▼ Return			

- ⑧ 프로그램 문서화를 목적으로 블록 타이틀, 블록 코멘트, 네트워크 1에 대한 유용한 네트워크 타이틀을 지정합니다.

(⑧ Block title: 수동 모드의 모터 제어 ⑧ Network 1: 수동 모드의 컨베이어 모터 제어)



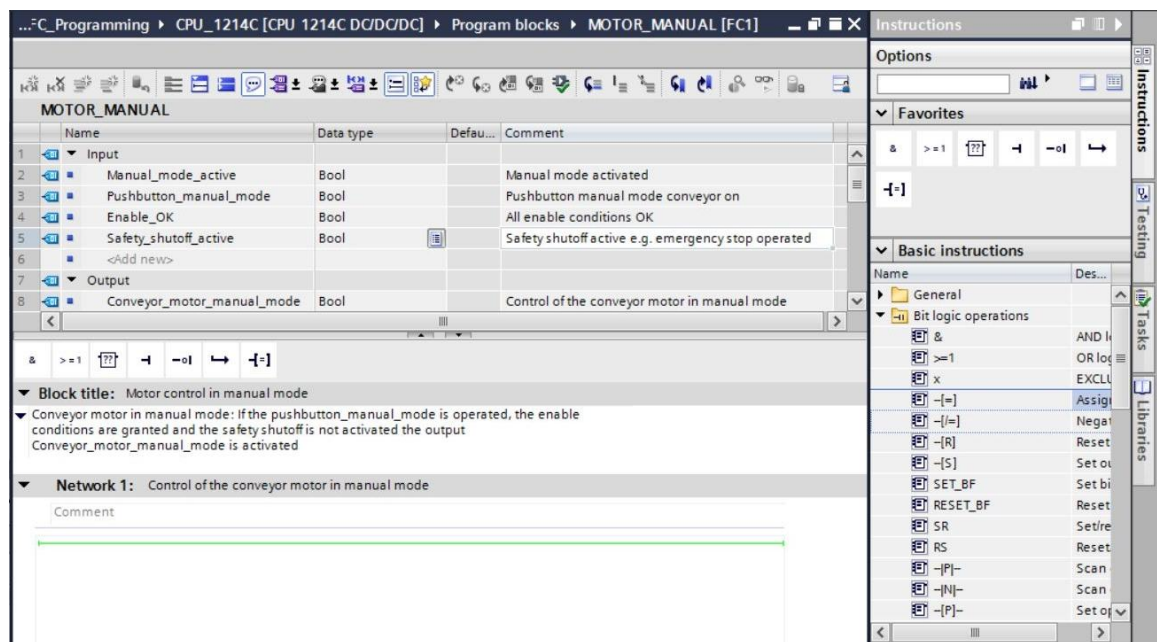
7.7 프로그램 FC1: MOTOR_MANUAL

⑧ “인터페이스 목록”으로 가면 프로그래밍 창에 여러가지 논리 연산을 가진 도구 모음이 보이고, 그 아래 네트워크 영역이 있습니다. 블록 타이틀과 첫 번째 네트워크에 대한 타이틀은 앞서 이미 지정해 두었습니다. 개별 로직 블록을 이용해 네트워크 내에서 프로그래밍이 수행됩니다. 여러 네트워크로 나누면 프로그램의 명확성을 지키는 데 도움이 됩니다. 아래에서는, 로직 블록을 삽입할 수 있는 다양한 방법에 대해 알아보겠습니다.



⑧ 프로그래밍 창의 오른쪽에 프로그램에서 사용할 수 있는 명령어 목록이 있습니다. ⑧ “Basic instructions” 아래의 ⑧ “Bit logic operations”으로 가서 평선 -[=](지정)를 찾고 끌어다 놓기를 이용해 Network 1 로 이를 이동시킵니다(녹색 선이 나타나고 마우스 포인터에 + 심볼이 표시됨).

(⑧ Instructions ⑧ Basic instructions ⑧ Bit logic operations ⑧ -[=])



- ⑧ 이제 끌려다 놓기를 이용해 출력 파라미터 `#Conveyor_motor_manual_mode` 를, 삽입한 블록 위의 `<???.?>`로 이동시킵니다. “인터페이스 목록”에서 파라미터를 선택하는 가장 좋은 방법은 파란색 심볼 을 클릭하여 “잡는” 것입니다.

(⑧ Conveyor_motor_manual_mode)

MOTOR_MANUAL

	Name	Data type	Defau...	Comment
1	Input			
2	Manual_mode_active	Bool		Manual mode activated
3	Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on
4	Enable_OK	Bool		All enable conditions OK
5	Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operated
6	<Add new>			
7	Output			
8	Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode

Block title: Motor control in manual mode

Conveyor motor in manual mode: If the pushbutton_manual_mode is operated, the enable conditions are granted and the safety shutoff is not activated the output Conveyor_motor_manual_mode is activated

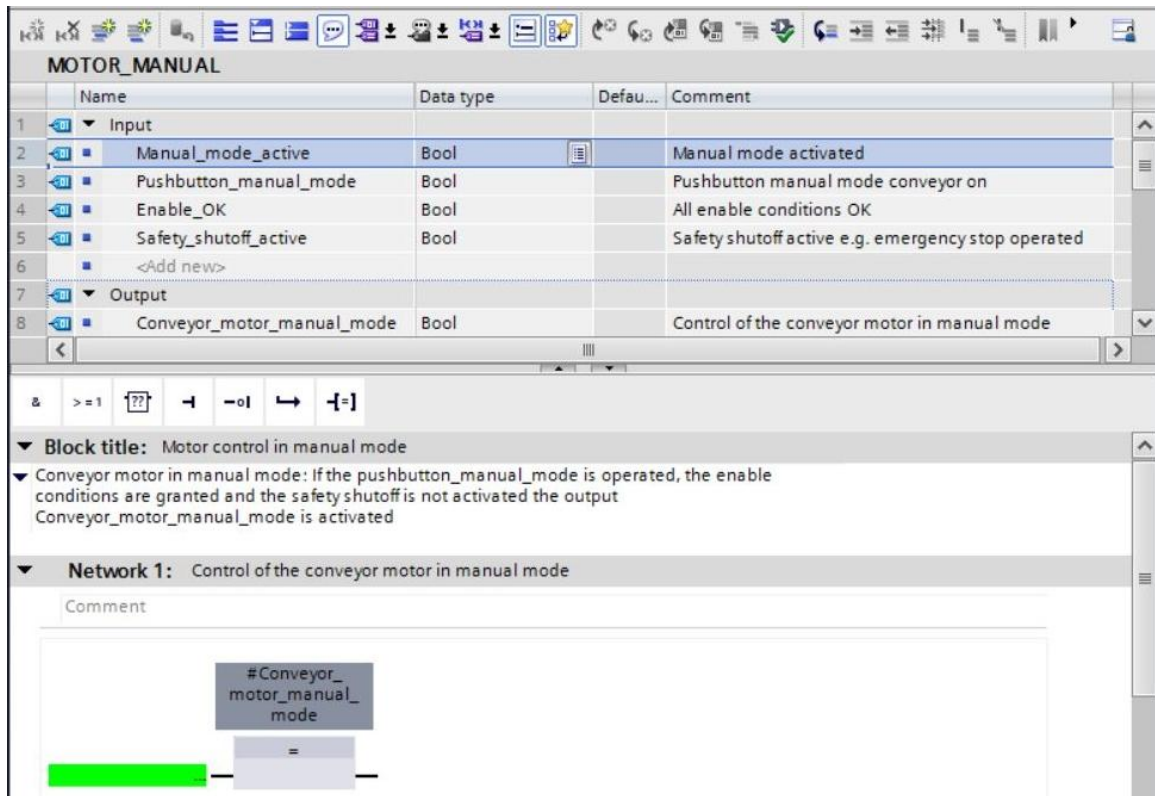
Network 1: Control of the conveyor motor in manual mode

Comment

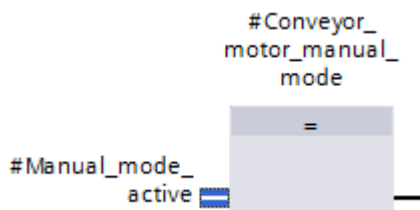
`<???.?>`

- ⑧ 이로써 이 블록에 의해 파라미터 **#Conveyor_motor_manual_mode** 가 기록되도록 결정합니다. 하지만 입력 조건이 아직 누락되어 있습니다 (실제로 이런 일이 발생함). 끌어다 놓기를 이용해 입력 파라미터 **#Manual_mode_active** 를 할당 블록의 왼쪽으로 이동시킵니다.

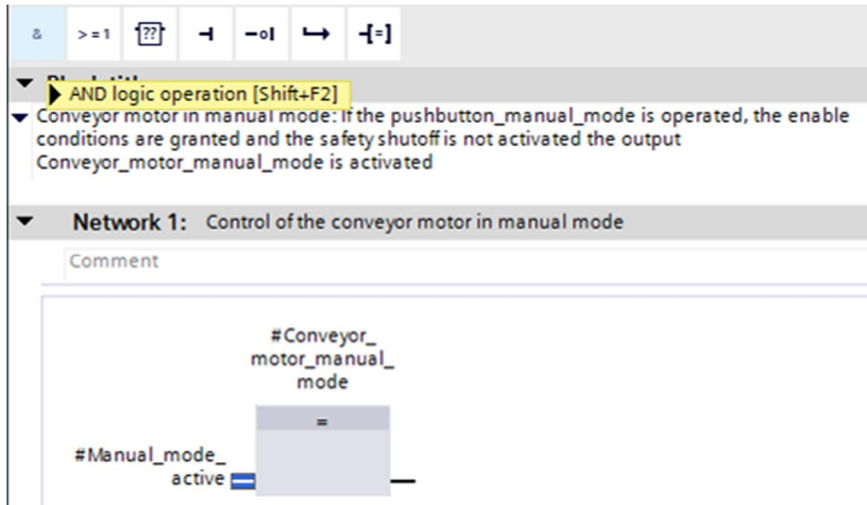
(⑧  **Manual_mode_active**)



- ⑧ 할당 블록의 입력 역시 **AND** 논리 연산을 통해 다른 파라미터들과 논리적으로 결합이 될 것입니다. 이를 위해 먼저, **#Manual_mode_active** 가 이미 연결되어 있는 블록의 입력을 클릭해서 입력 라인이 파란색 백그라운드가 되도록 합니다.

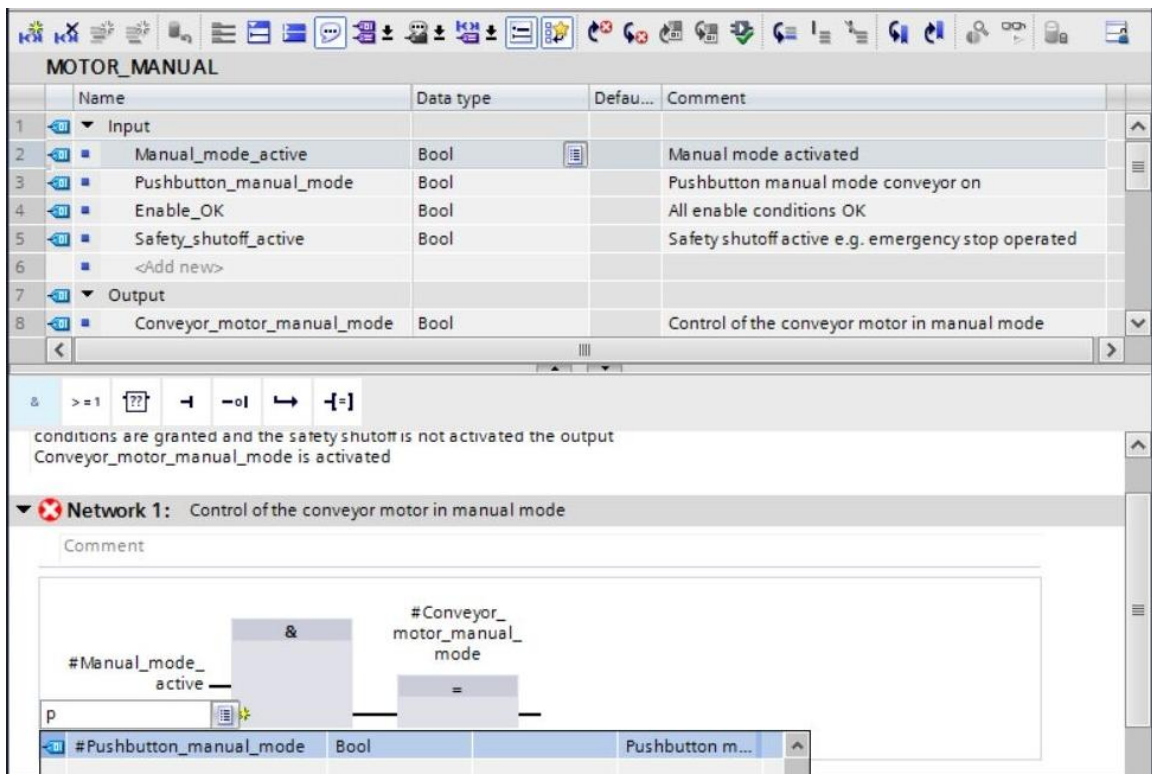


- ⑧ 로직 도구 모음의  아이콘을 클릭해서 #Manual_mode_active 태그와 할당 블록 사이에 AND 논리 연산을 삽입합니다.



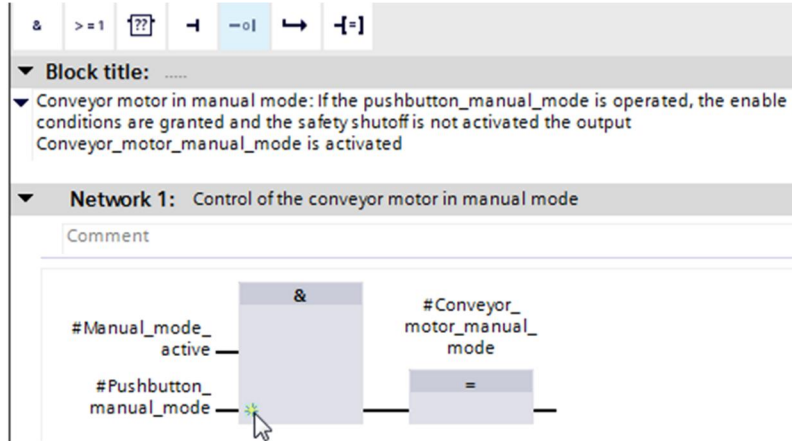
- ⑨ AND 논리 연산의 두 번째 입력 **<???.?>**을 더블클릭하고 이 때 나타난 필드에 문자 “P”를 입력하면 “P”로 시작되는 가용 태그 목록을 확인할 수 있습니다. #Pushbutton_manual_mode 태그를 클릭하고 ⑩ 엔터를 눌러 적용합니다.

(⑧ AND 블록 ⑧ **<???.?>** ⑧ P ⑧ #Pushbutton_manual_mode ⑧ 엔터)

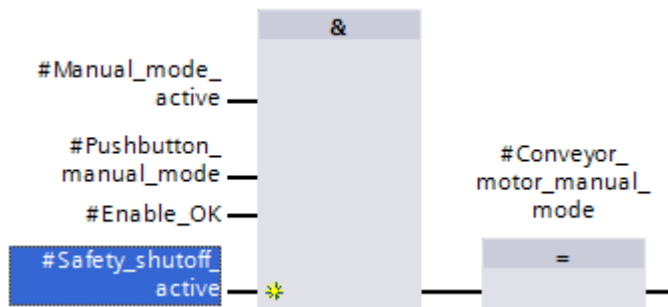


참고: 이런 방식으로 태그를 지정하면 태그 테이블에서 글로벌 태그를 혼동할 위험이 있습니다. 따라서 “인터페이스 목록”에서 끌어다 놓기를 이용해 이전에 표현된 절차를 우선적으로 사용해야 합니다.

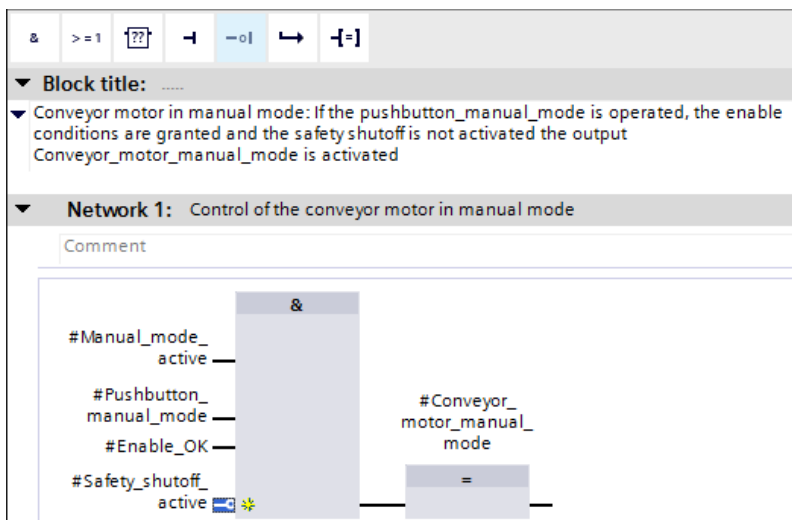
- ⑧ 활성화 조건이 충족되고 안전 전원 차단이 활성화되지 않을 때만 출력을 제어할 수 있으려면 **#Enable_OK** 및 **#Safety_shutoff_active** 입력 태그를 **AND** 논리 연산을 통해 논리적으로 결합해야 합니다. 이를 위해 **AND** 블록의 노란색 별 표시 를 더블클릭해서 2 개의 입력을 추가합니다.



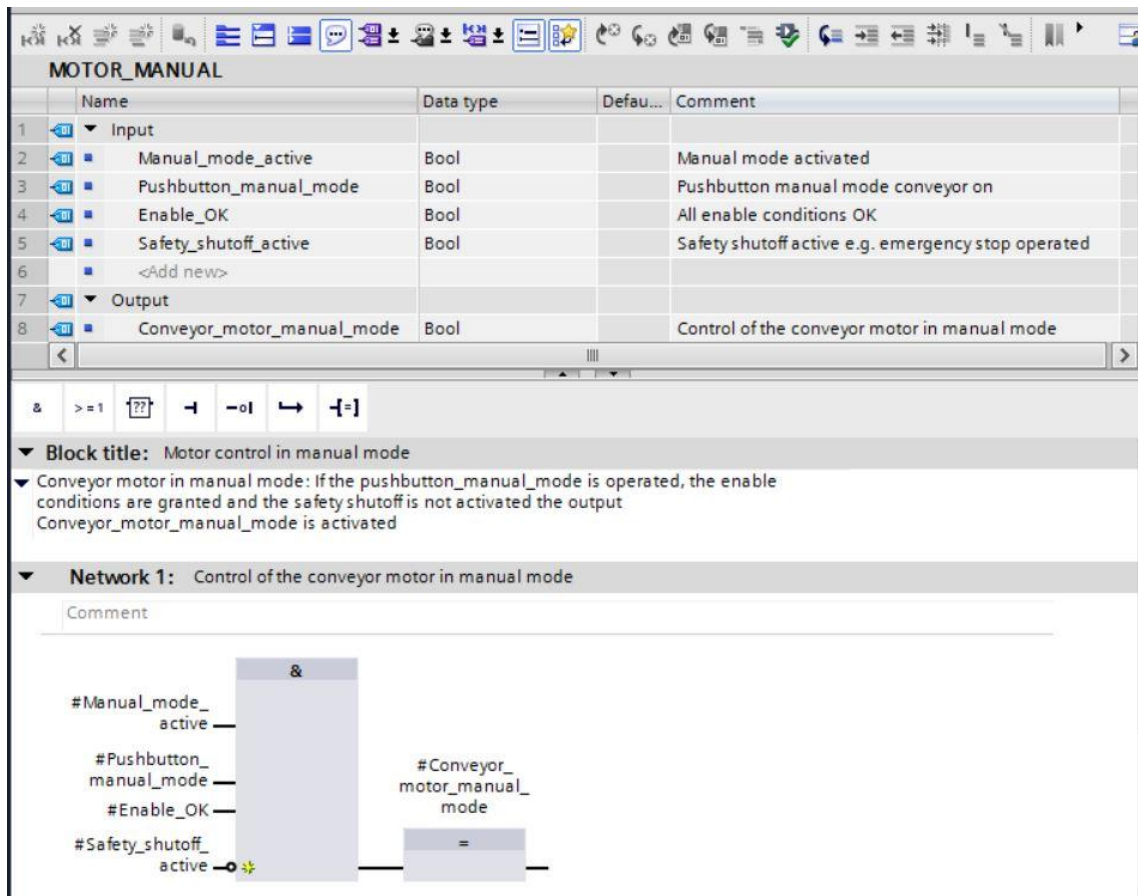
- ⑧ **AND** 블록의 새로 생성된 입력에 입력 태그 **#Enable_OK** 및 **#Safety_shutoff_active** 를 추가합니다.



- ⑧ 파라미터 **#Safety_shutoff_active** 을 선택하고  을 클릭하면 이에 연결된 입력이 부정화됩니다.

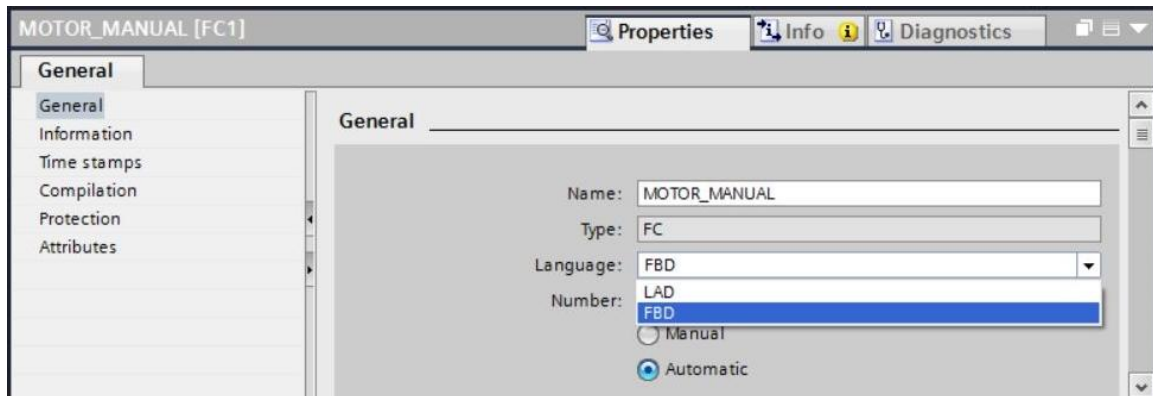


- ⑧ 주기적으로  Save project 을 클릭하는 것을 잊지 않도록 합니다. FBD 에서 완료된 평선 “MOTOR_MANUAL” [FC1]가 아래와 같이 나타납니다.

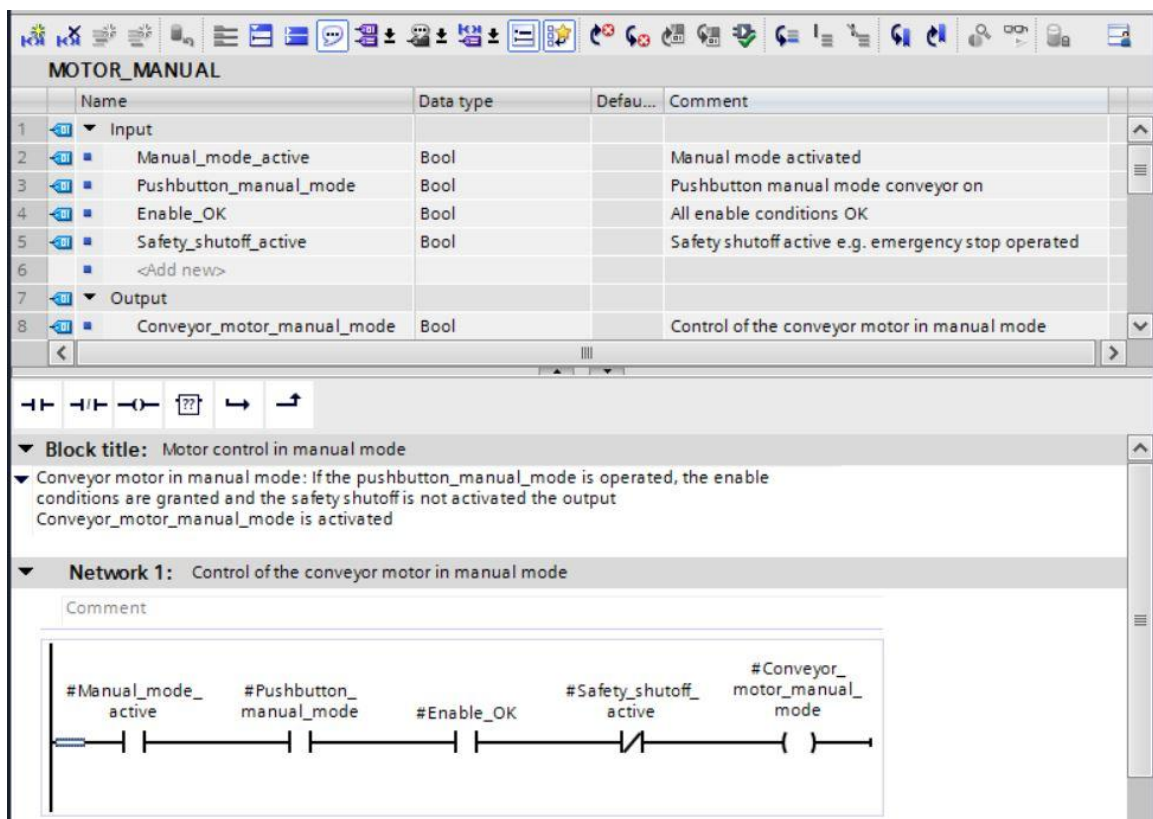


⑧ 블록 속성의 “General”으로 가서 “Language”를 래더 로직(LAD)으로 변경할 수 있습니다.

(⑧ Properties ⑧ General ⑧ Language: LAD)



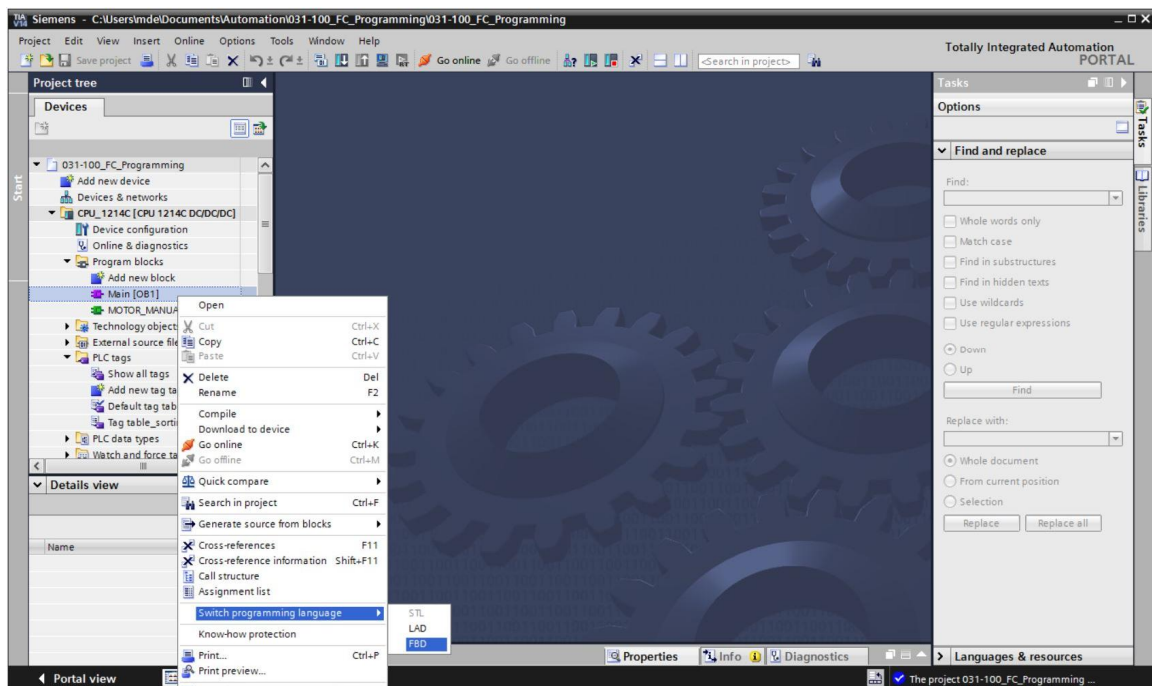
⑧ 이 프로그램은 LAD로 다음과 같이 나타납니다.



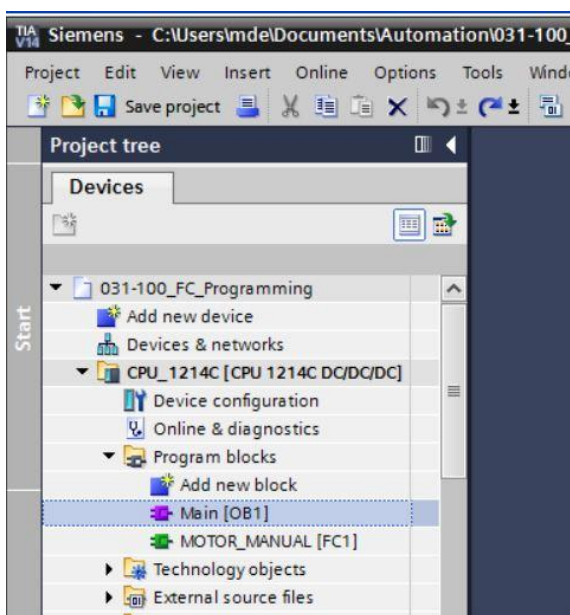
7.8 오거나이제이션 블록 OB1 프로그래밍 – 수동 모드에서 컨베이어 전진 트래킹 제어

⑧ 오거나이제이션 블록 “Main [OB1]”을 프로그래밍하기 전에 평선 블록 다이어그램(FBD)으로 프로그래밍 언어를 전환해야 합니다. 이를 위해 먼저, “Program blocks” 폴더에서 “Main [OB1]”를 클릭합니다.

(⑧ CPU_1214C [CPU 1214C DC/DC/DC ⑧ Program block ⑧ Main [OB1] ⑧ Switch programming language ⑧ FBD)

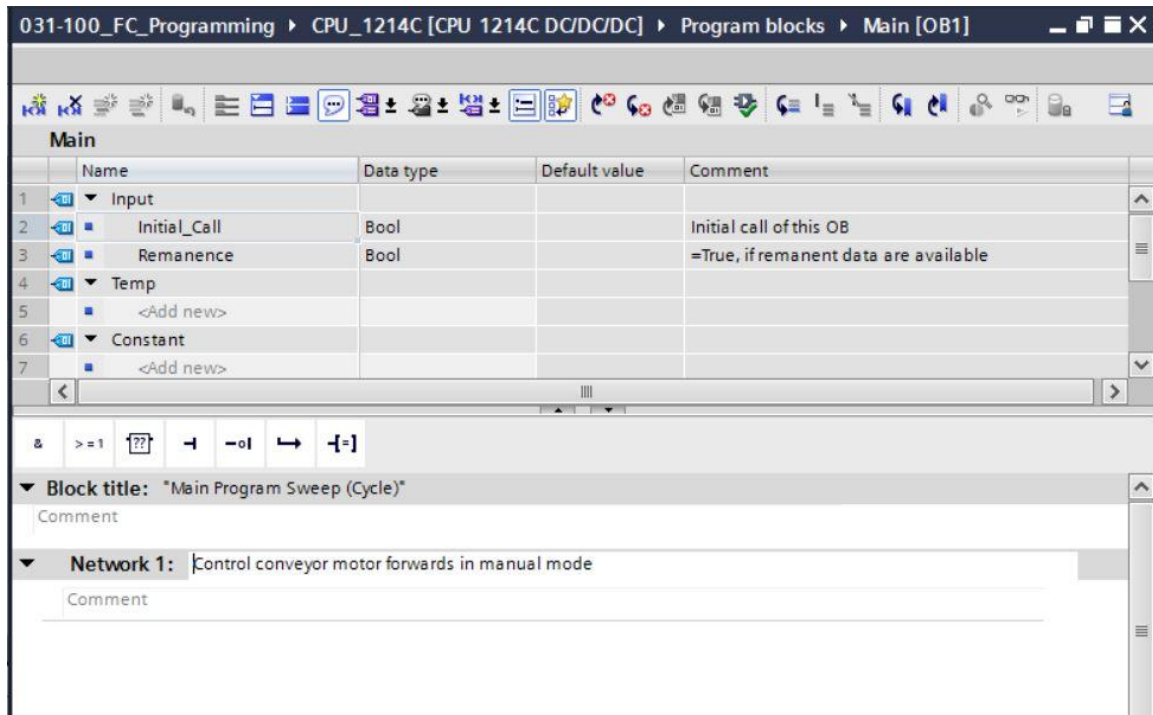


⑧ “Main [OB1]” 오거나이제이션 블록을 더블 클릭해서 엽니다.

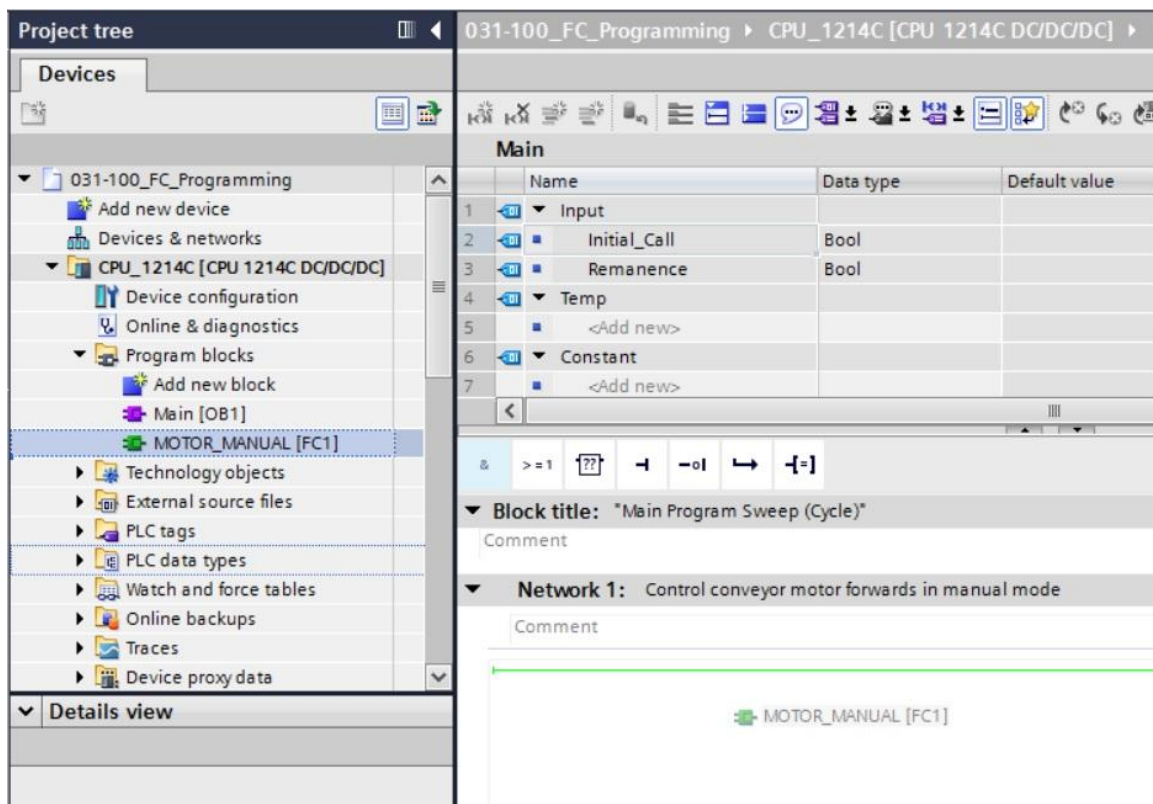


⑧ Network 1의 이름을 “수동/조그 모드에서 컨베이어 전진 트래킹 제어”라고 지정합니다.

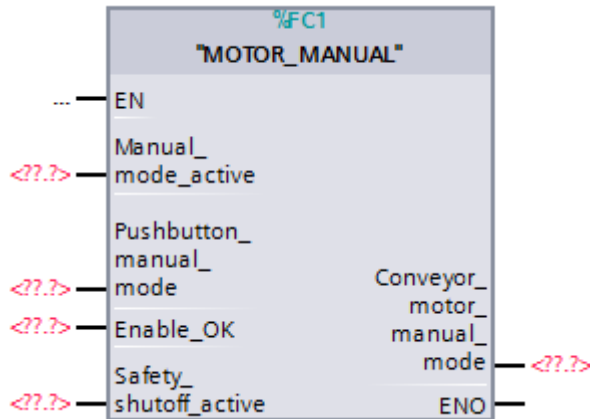
(⑧ Network 1: ⑧ 수동/조그 모드에서 컨베이어 전진 트래킹 제어)



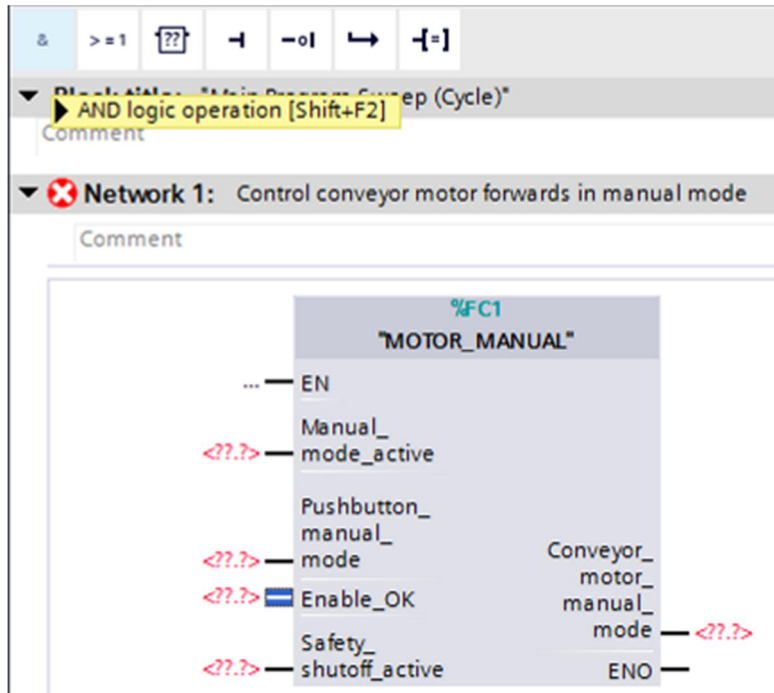
⑧ 끌어다 놓기를 이용해 “MOTOR_MANUAL [FC1]” 평선을 네트워크 1의 녹색 라인으로 이동시킵니다.



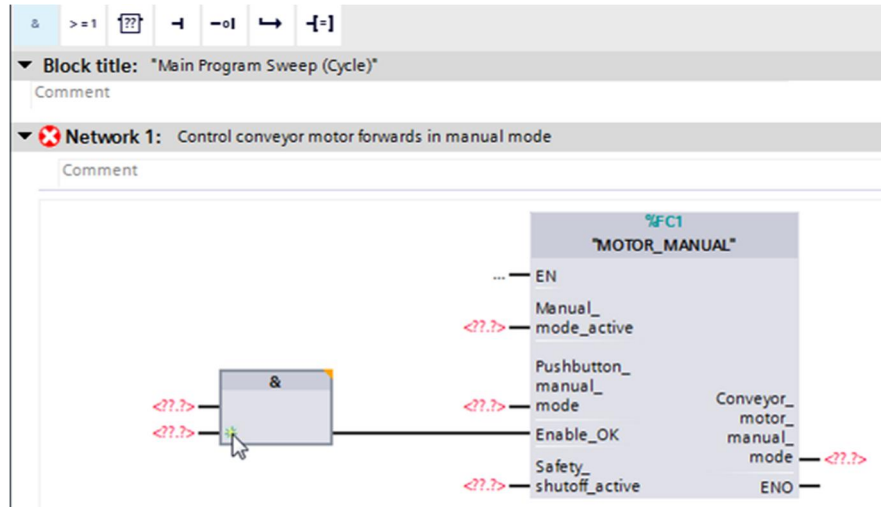
⑧ 정의한 인터페이스와 EN 및 ENO 연결이 포함된 블록이 **Network 1**에 삽입됩니다.



⑧ 입력 파라미터 “Enable_OK” 앞에 AND 를 삽입하기 위해 이 입력을 선택하고 로직 도구 모음의 아이콘  을 클릭해서 AND 를 삽입합니다. (⑨ )

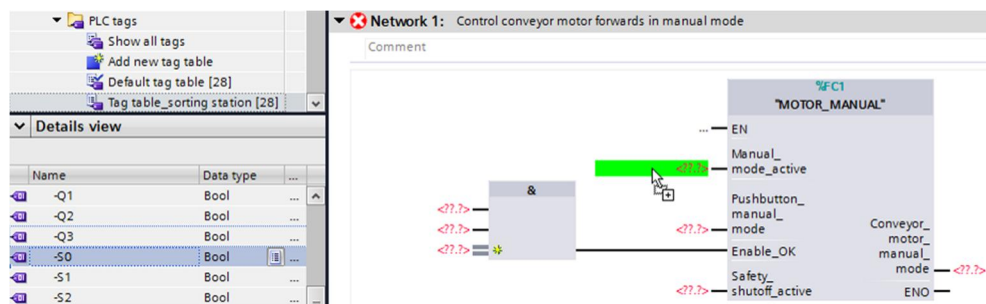


⑧ AND 블록의 노란색 별 표시 를 클릭해서 다른 입력을 추가합니다. (⑧ )

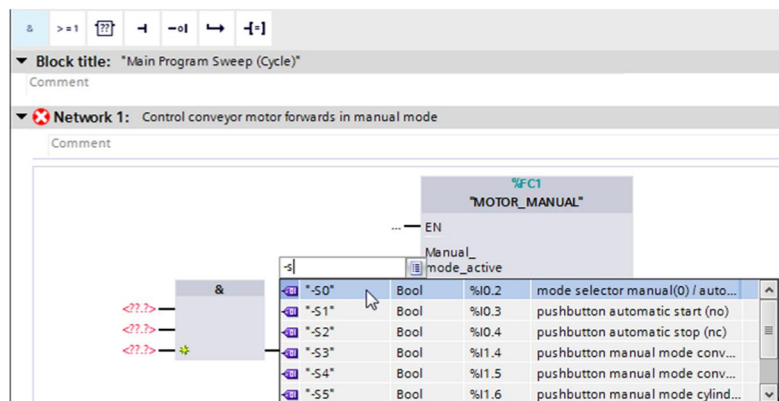


⑧ “Tag_table_sorting_station”의 글로벌 태그에서 블록으로 연결하는 방법은 두 가지가 있습니다.

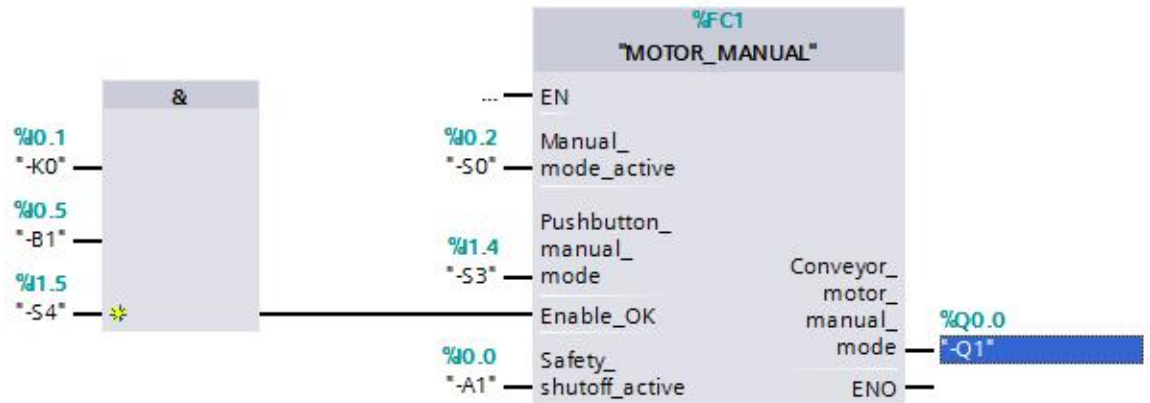
⑧ 프로젝트 트리의 “Tag_table_sorting_station”의 상세 뷰(Details View)에서 원하는 글로벌 태그를 선택하여 끌어다 놓기를 이용해 FC1의 인터페이스로 이동시키는 방법입니다.
(⑧ Tag_table_sorting_station ⑧ 상세 뷰. ⑧ -S0 ⑧ Manual_mode_active)



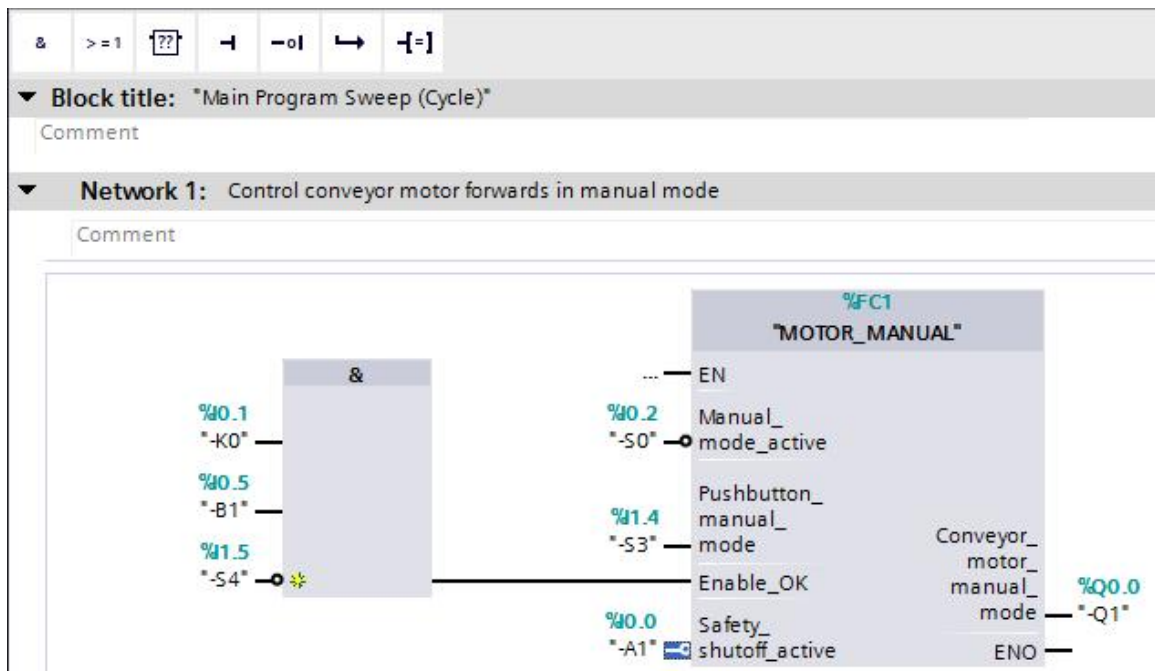
⑧ 또는 <??.?>에 대해 원하는 글로벌 태그의 시작 문자(예: “-S”)를 입력하고 화면에 나타난 목록에서 글로벌 입력 태그 “-S0”(%I0.2)를 선택합니다(⑧ Manual_mode_active ⑧ -S ⑧ -S0).



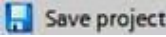
- ⑧ 다른 입력 태그 “-S3”, “-K0”, “-B1”, “-S4” 및 “-A1”을 삽입하고 출력 “Conveyor_motor_manual_mode”에 출력 태그 “-Q1”(Q0.0)를 삽입합니다.

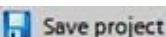


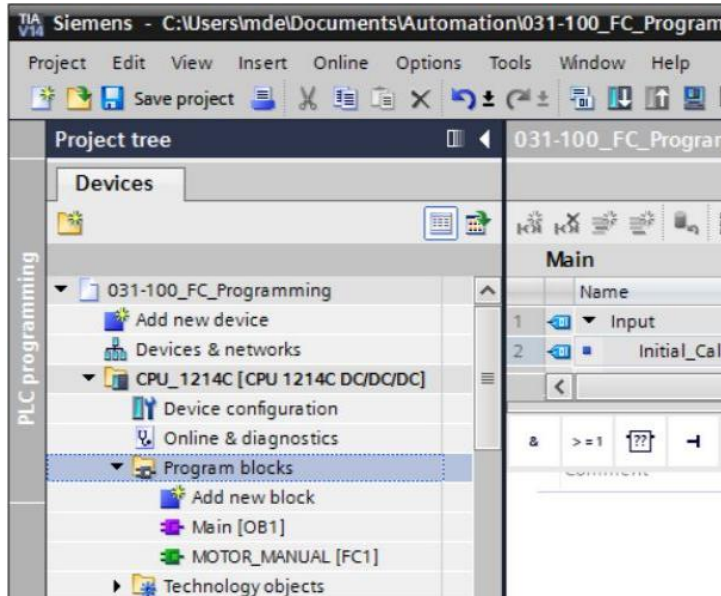
- ⑧ 입력 태그 “-S0”, “-S4” 및 “-A1”을 선택하고  을 클릭하면 이들에 대한 쿼리가 부정화됩니다. (⑧ -S0 ⑧  ⑧ -S4 ⑧  ⑧ -A1 ⑧ )



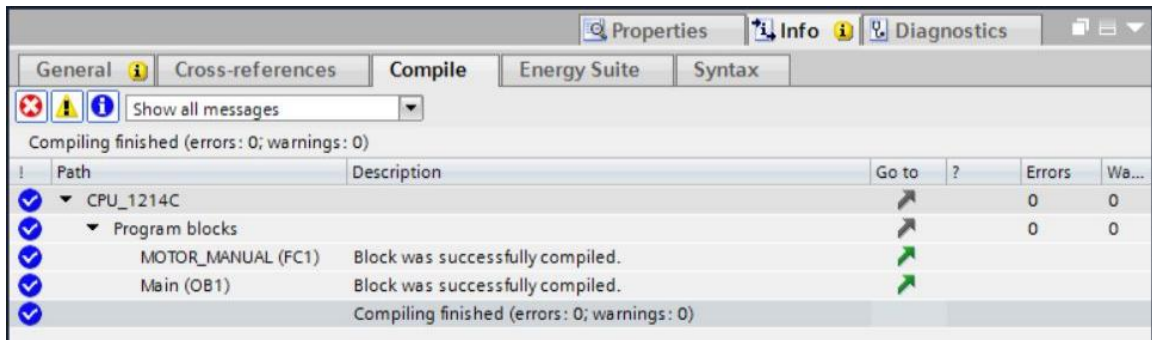
7.9 프로그램 저장 및 컴파일

⑧ 프로젝트를 저장하려면 메뉴에서  버튼을 선택합니다. 모든 블록을 컴파일하려면 “Program blocks” 폴더를 클릭하고 메뉴에서 컴파일을 위한 아이콘  을 선택합니다

(⑧  ⑧ Program blocks ⑧ ).

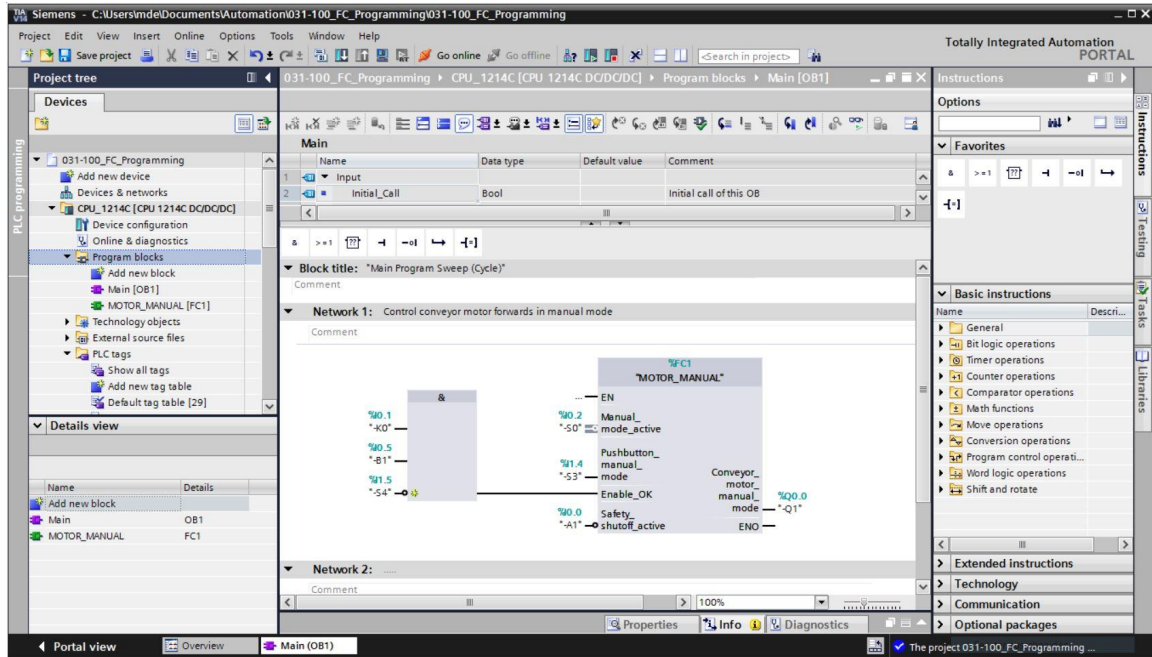


⑧ “Info” 아래의 “Compile” 영역에 어떤 블록이 성공적으로 컴파일 되었는지가 나타납니다.



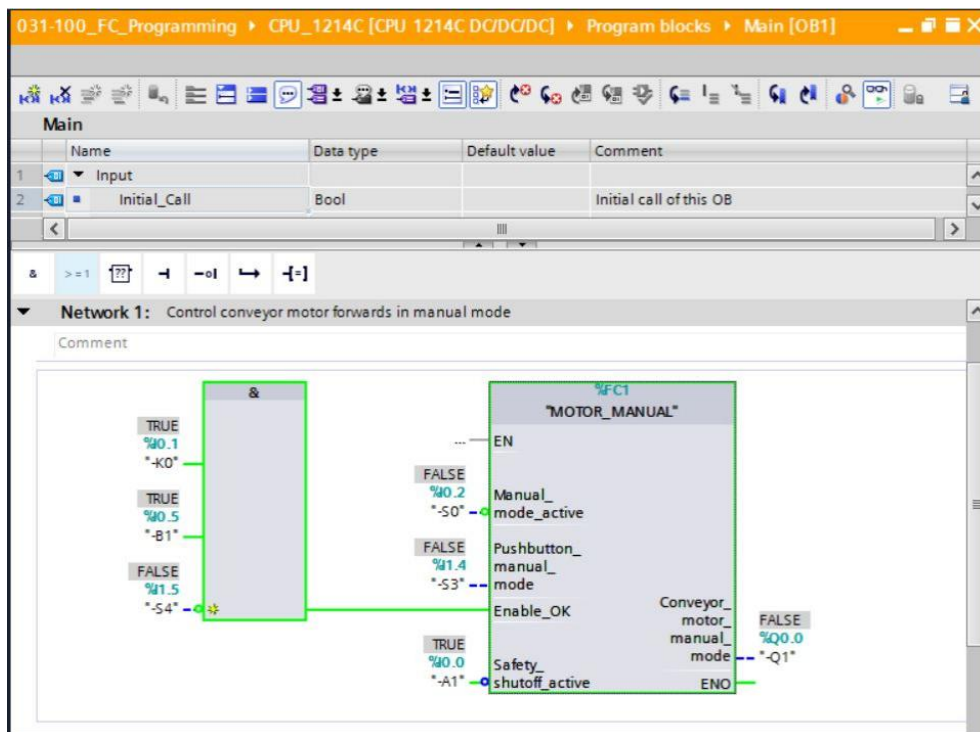
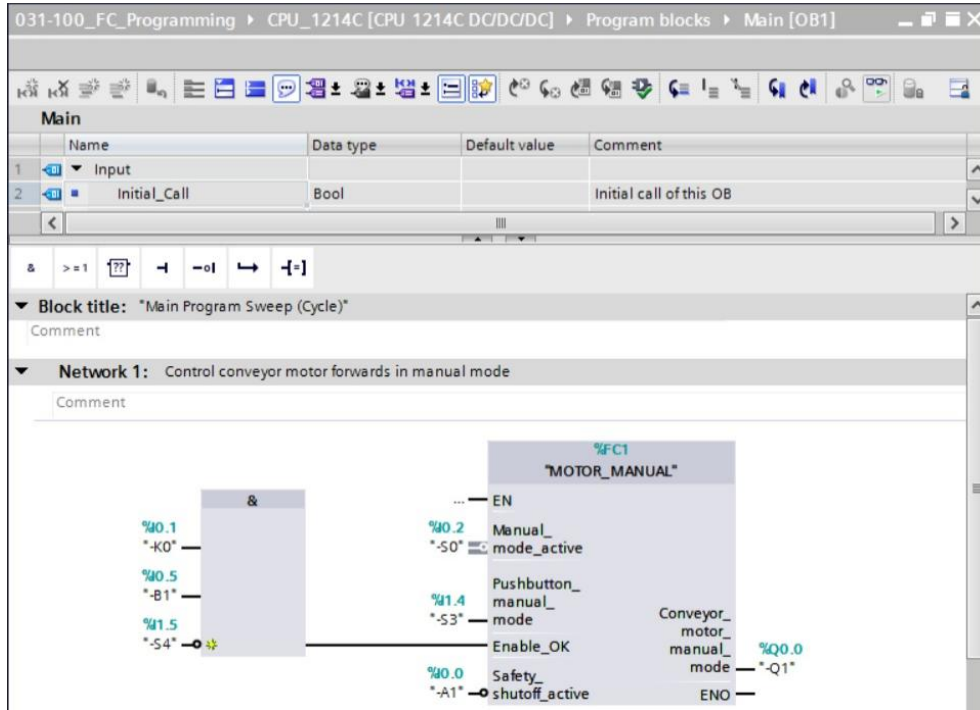
7.10 프로그램 다운로드

- ⑧ 컴파일이 성공적으로 완료되고 앞서 설명한 하드웨어 구성을 위한 모듈에서와 같이, 생성된 프로그램과 함께 전체 컨트롤러를 다운로드 할 수 있습니다 (⑧ )



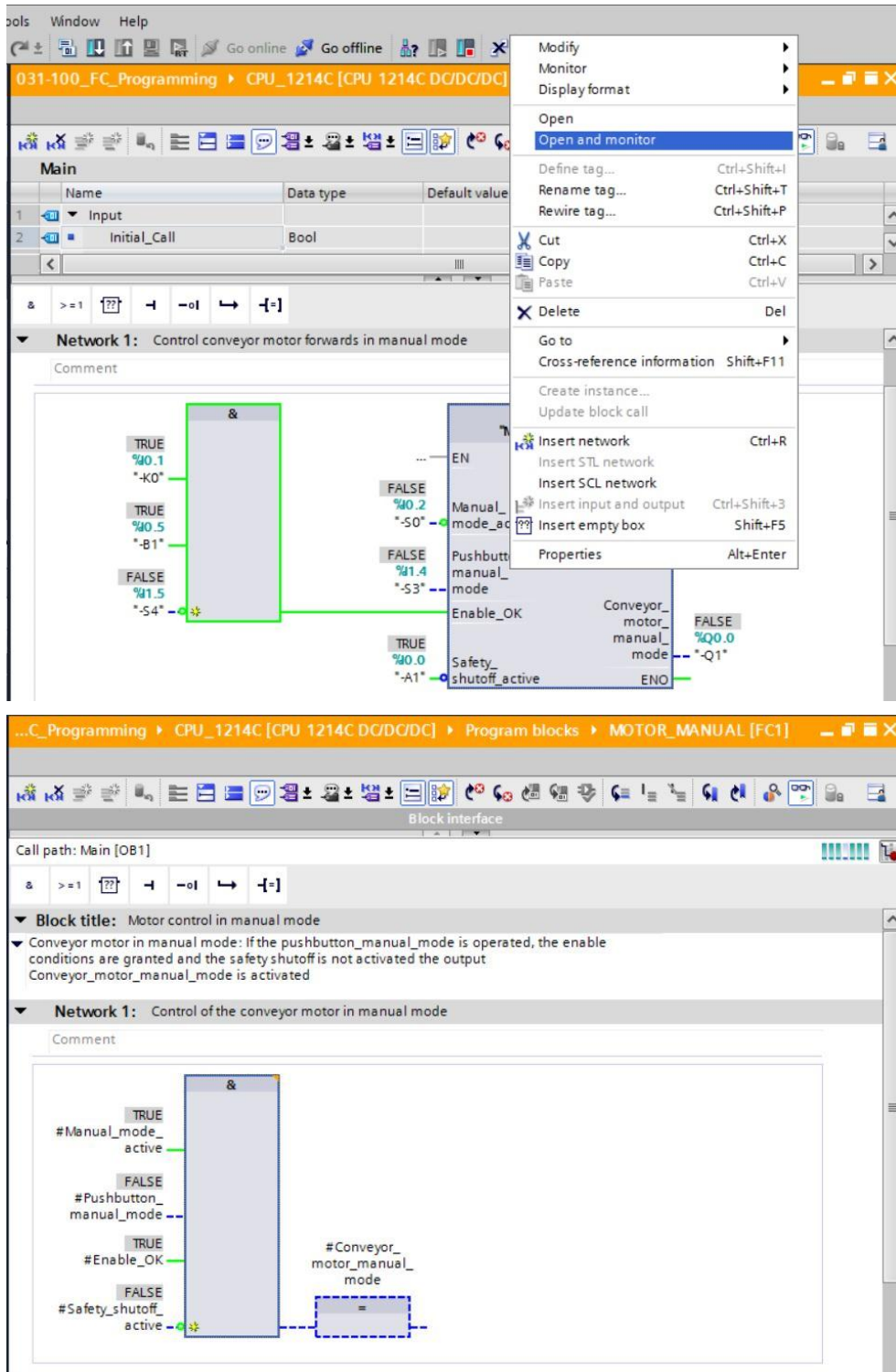
7.11 프로그램 블록 모니터링

⑧ 다운로드된 프로그램을 모니터링 하려면 원하는 블록을 열어야 합니다.  아이콘을 클릭해서 모니터링을 활성화/비활성화할 수 있습니다(⑧ Main [OB1] ⑧ ).



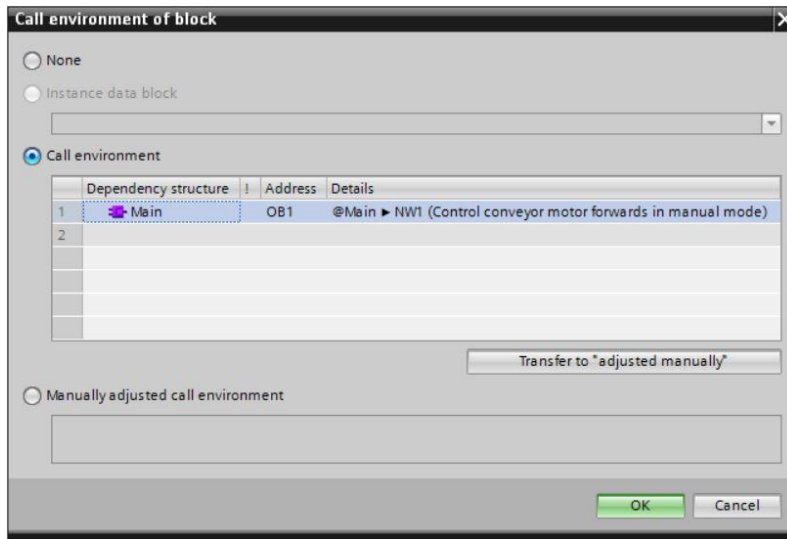
참고: 위의 모니터링 상태는 신호와 관련이 있으며 컨트롤러에 따라 다르게 보일 수 있습니다. 터미널에서의 신호 상태는 TRUE 또는 FALSE 로 표시가 됩니다.

- ⑧ 마우스 오른쪽 버튼을 클릭하여 “열기 및 모니터링”에서 **Main [OB1]** 오거나이제이션 블록에서 호출된 “MOTOR_MANUAL” [FC1] 평션을 직접 선택할 수 있습니다. (⑧ “MOTOR_MANUAL” [FC1] ⑧ Open and monitor)



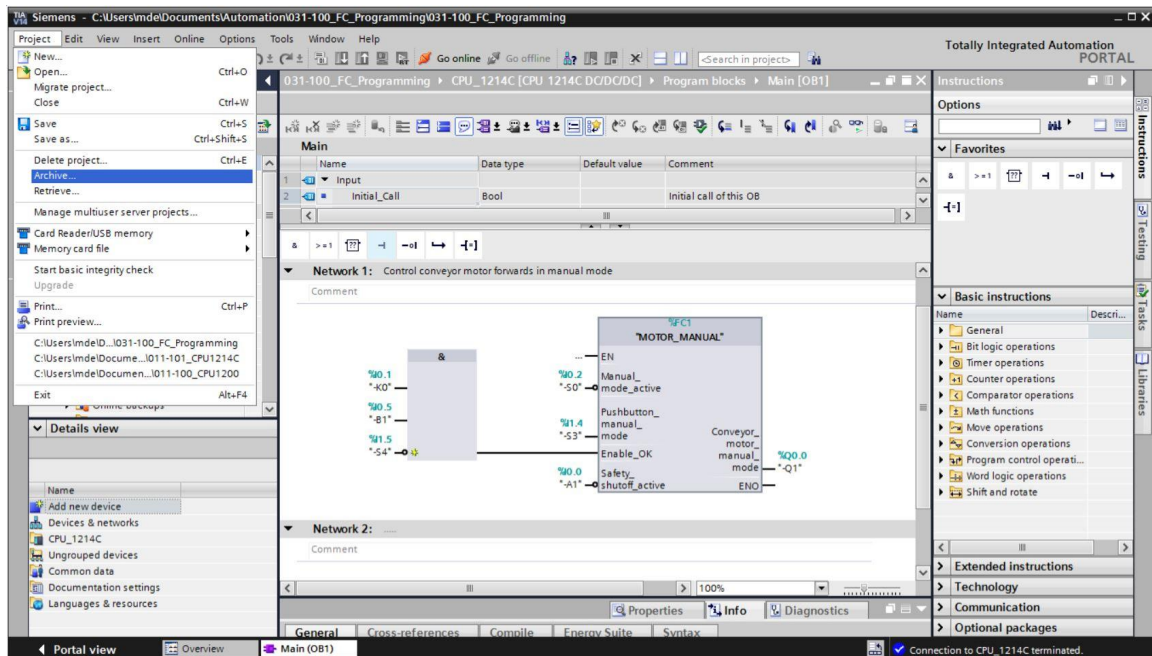
참고: 위의 모니터링 상태는 평션과 관련이 있으며 컨트롤러로부터 독립적입니다. 센서 작동 및 스테이션 상태가 여기에 TRUE 또는 FALSE로 표시가 됩니다.

- ④ “MOTOR_MANUAL” [FC1] 평선의 특정 사용 위치를 모니터링 하려는 경우에는  아이콘을 이용해 호출 환경을 선택할 수 있습니다. (④  ④ Call environment ④ OK)



7.12 프로젝트 아카이브

- ④ 마지막 단계로 전체 프로젝트를 아카이브 하려고 합니다. ④ “Project” 메뉴에서 ④ “Archive...” 항목을 선택합니다. 프로젝트를 아카이브 하고자 하는 폴더를 선택하고 “TIA Portal 프로젝트 아카이브” 파일 유형으로 이를 저장합니다. (④ Project ④ Archive... ④ TIA Portal 프로젝트 아카이브 ④ 031-100_FC Programming.... ④ Save)



7.13 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 수동 모드(-S0 = 0) 수동 모드 컨베이어 전진 활성화(-S3 = 1) 컨베이어 모터 고정 속도로 전진(-Q1 = 1)	
4	3 과 동일하지만 비상 정지 오프 활성화(-A1 = 0) ® -Q1 = 0	
5	3 과 동일하지만 자동 모드(-S0 = 1) ® -Q1 = 0	
6	3 과 동일하지만 스테이션 전원 끄기(-K0 = 0) ® -Q1 = 0	
7	3 과 동일하지만 실린더가 복귀되지 않음(-B1 = 0) ® -Q1 = 0	
8	3 과 동일하지만 수동 모드에서 컨베이어 후진 활성화 (-S4 = 1) ® -Q1 = 0	
9	프로젝트가 성공적으로 아카이브 됨	

8 연습

8.1 과제 – 연습

이 챕터에서는 다음과 같이 선별기 프로세스 설명에 대한 평선들을 계획, 프로그래밍 및 테스트 해 보겠습니다.

- 수동 모드 - 수동/조그 모드에서 컨베이어 후진 트래킹 제어

8.2 기술 다이어그램

여기에는, 과제에 대한 기술 다이어그램이 나와 있습니다.

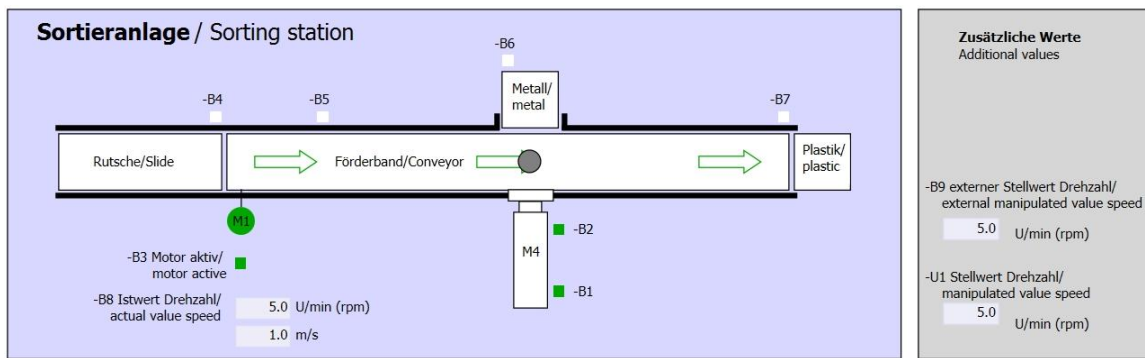


그림 10: 기술 다이어그램



그림 11: 제어 패널

8.3 참조 목록

이 과제를 위한 글로벌 오퍼랜드로서 아래와 같은 신호들이 필요합니다.

DI	유형	식별자	평션	NC/NO
I 0.0	BOOL	-A1	반환 신호 비상 정지 확인	NC
I 0.1	BOOL	-K0	메인 스위치 "ON"	NO
I 0.2	BOOL	-S0	모드 선택 수동(0)/자동(1)	수동 = 0 자동 = 1
I 0.5	BOOL	-B1	센서 실린더 M4 복귀	NO
I 1.4	BOOL	-S3	푸시버튼 수동 모드 컨베이어 M1 전진	NO
I 1.5	BOOL	-S4	푸시버튼 수동 모드 컨베이어 M1 후진	NO

DO	유형	식별자	평션	
Q 0.1	BOOL	-Q2	컨베이어 모터 M1 후진 고정 속도	

참조 목록 범례

DI	디지털 입력	DO	디지털 출력
AI	아날로그 입력	AO	아날로그 출력
I	입력	Q	출력
NC	상시 닫힘		
NO	상시 열림		

8.4 계획 수립

과제 수행에 대한 계획을 자체적으로 수립합니다.

8.5 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 수동 모드(-S0 = 0) 수동 모드 컨베이어 후진 활성화(-S4 = 1) 컨베이어 모터 후진 고정 속도(-Q2 = 1)	
4	8 과 동일하지만 비상 정지 오프 활성화(-A1 = 0) ® -Q2 = 0	
5	8 과 동일하지만 자동 모드(-S0 = 1) ® -Q2 = 0	
6	8 과 동일하지만 스테이션 전원 끄기(-K0 = 0) ® -Q2 = 0	
7	8 과 동일하지만 실린더가 복귀되지 않음(-B1 = 0) ® -Q2 = 0	
8	8 과 동일하지만 수동 모드 컨베이어 전진 활성화 (-S3 = 1) ® -Q1 = 0 and -Q2 = 0	
9	프로젝트가 성공적으로 아카이브 됨	

9 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

"추가 정보" 미리보기

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

- TIA Portal Videos
- TIA Portal Tutorial Center
- Getting Started
- Programming Guideline
- Easy Entry in SIMATIC S7-1200
- Download Trial Software/Firmware
- Technical Documentation SIMATIC Controller
- Industry Online Support App
- TIA Portal, SIMATIC S7-1200/1500 Overview
- TIA Portal Website
- SIMATIC S7-1200 Website
- SIMATIC S7-1500 Website

추가 정보

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

SCE 교육 커리큘럼
[siemens.com/sce/documents](https://www.siemens.com/sce/documents)

SCE 교육 담당자 패키지
[siemens.com/sce/tp](https://www.siemens.com/sce/tp)

SCE 담당 파트너
[siemens.com/sce/contact](https://www.siemens.com/sce/contact)

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

Industrie 4.0
[siemens.com/future-of-manufacturing](https://www.siemens.com/future-of-manufacturing)

완전히 통합된 자동화 시스템 (TIA)
[siemens.com/tia](https://www.siemens.com/tia)

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

SIMATIC 컨트롤러
[siemens.com/controller](https://www.siemens.com/controller)

SIMATIC 기술 문서
[siemens.com/simatic-docu](https://www.siemens.com/simatic-docu)

산업 온라인 지원
support.industry.siemens.com

제품 카탈로그 및 온라인 주문 시스템 산업 물
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](https://www.siemens.com/sce)