

# codingOn x posco

K-Digital Training 스마트 팩토리 단기 8기

## 5. 사용자 정의 평션

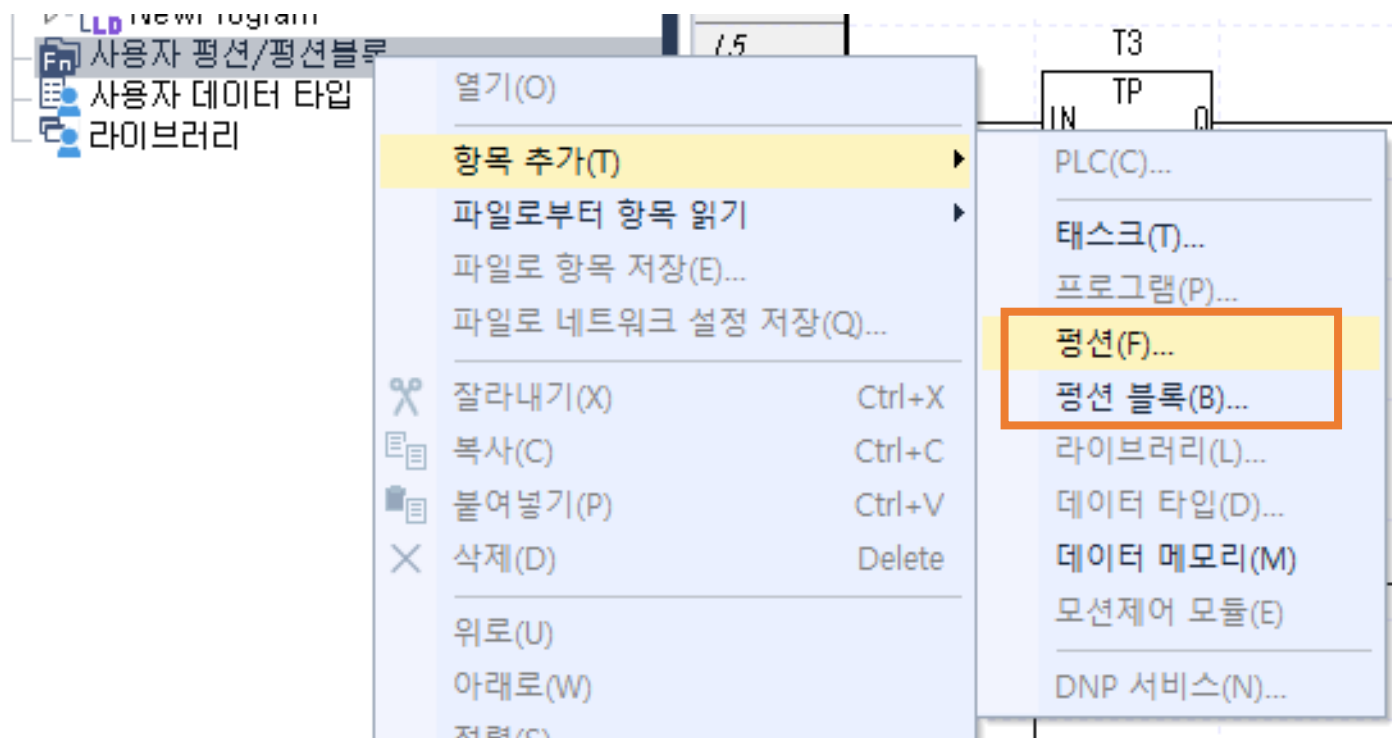
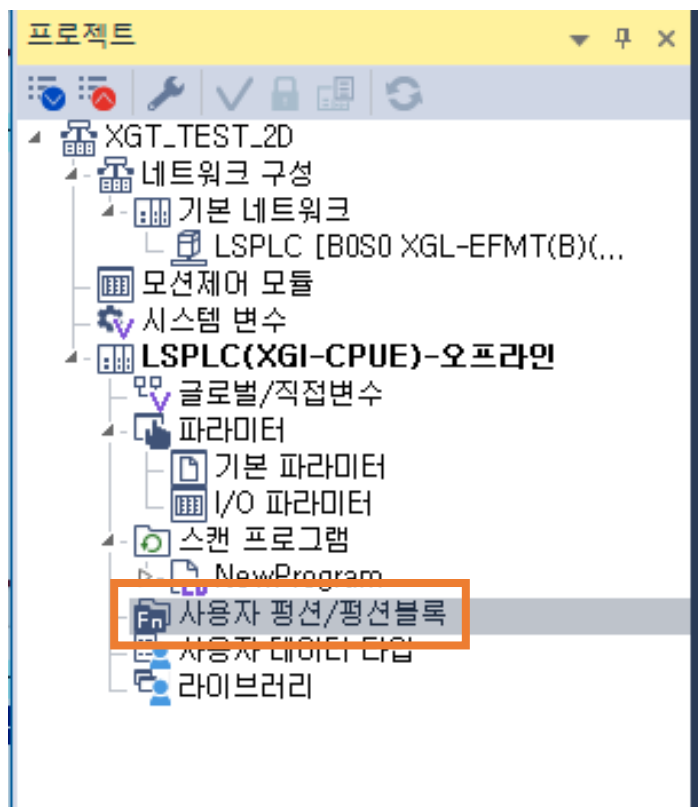


# 사용자 정의 평션

- 기존의 평션/평션블록 외에 사용자가 직접 만드는 평션/평션 블록
  - 사용자가 직접 평션 또는 평션블록을 디자인 하여 동일한 로직을 재활용
  - 사용자 정의로 로직을 숨김으로써 노하우 보호
  - 프로그램 메모리를 효과적으로 활용할 수 있음

# 사용자 정의 평선

- 사용자가 직접 내용을 정의할 수 있는 평선



# 사용자 정의 평션

- 이름, 데이터 타입, 리턴 타입 정의

사용자 평션/평션블록

사용자 평션/평션블록    비밀번호    고급 설정

이름(N):  
시간검출

언어(L):  
LD

☒ EN/ENO 사용

리턴 데이터 타입(R): ARRAY

너비 컬럼 수(W): 1   

설명문(C):

제작하려는 기능별로 자유롭게 선택 가능

배열 선택

배열 차수(I)

☒ ARRAY[0., 5 ] OF

☐ ARRAY[0., 1 ,0., 1 ] OF

☐ ARRAY[0., 1 ,0., 1 ,0., 1 ] OF

배열 타입(T):  
WORD

- 스캔 프로그램
- NewProgram
- 사용자 평션/평션블록
  - 시간검출
- 사용자 데이터 타입
- 라이브러리

# 사용자 정의 평션

- 평션에서 사용할 입력, 출력 변수 정의

UserFunction[프로그램] x ADDSUB[로컬변수] x

	변수 종류	변수	타입	사용 유무	설명문
1	VAR_INPUT	입력A	INT	<input checked="" type="checkbox"/>	
2	VAR_INPUT	입력B	INT	<input checked="" type="checkbox"/>	
3	VAR_INPUT	입력C	INT	<input checked="" type="checkbox"/>	
4	VAR_RETURN	ADDSUB	INT	<input checked="" type="checkbox"/>	
5	VAR	임시저장	INT	<input checked="" type="checkbox"/>	

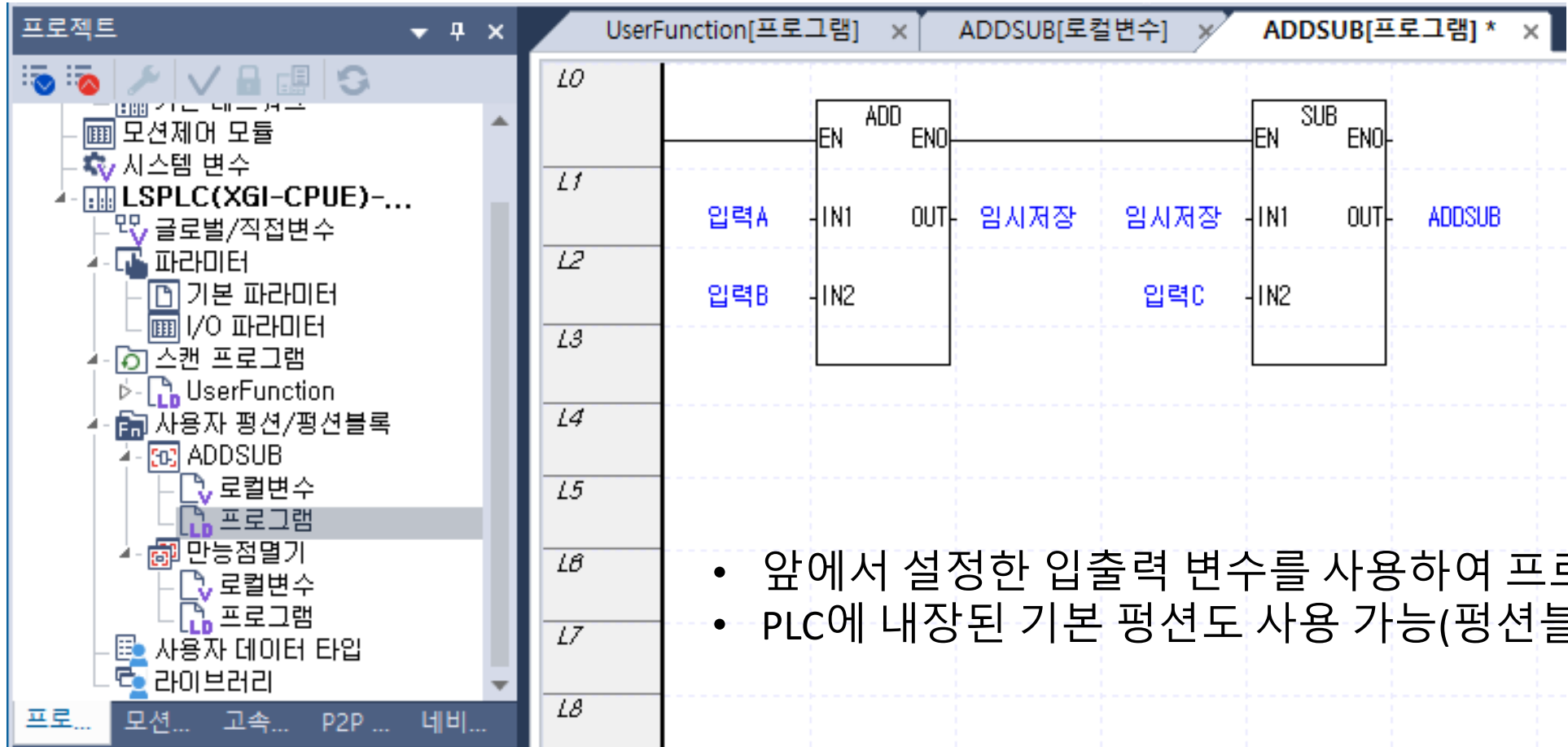
  

```
graph LR
    subgraph ADDSUB [ADDSUB]
        direction TB
        I1[INT - 입력A]
        I2[INT - 입력B]
        I3[INT - 입력C]
    end
    ADDSUB --> EN[BOOL - EN]
    ADDSUB --> END[BOOL - END]
```

The screenshot displays the PLC Studio software interface. On the left, the 'Function Block' palette is open, showing various blocks including 'ADD SUB'. The main workspace shows a ladder logic diagram with a 'Network 1' containing a 'Network 1' block. The 'Monitor' window at the bottom shows the 'Program Check' results, indicating that the program is valid and ready for execution.

# 사용자 정의 평션

- **평선 내용은 래더 프로그램으로 작성**

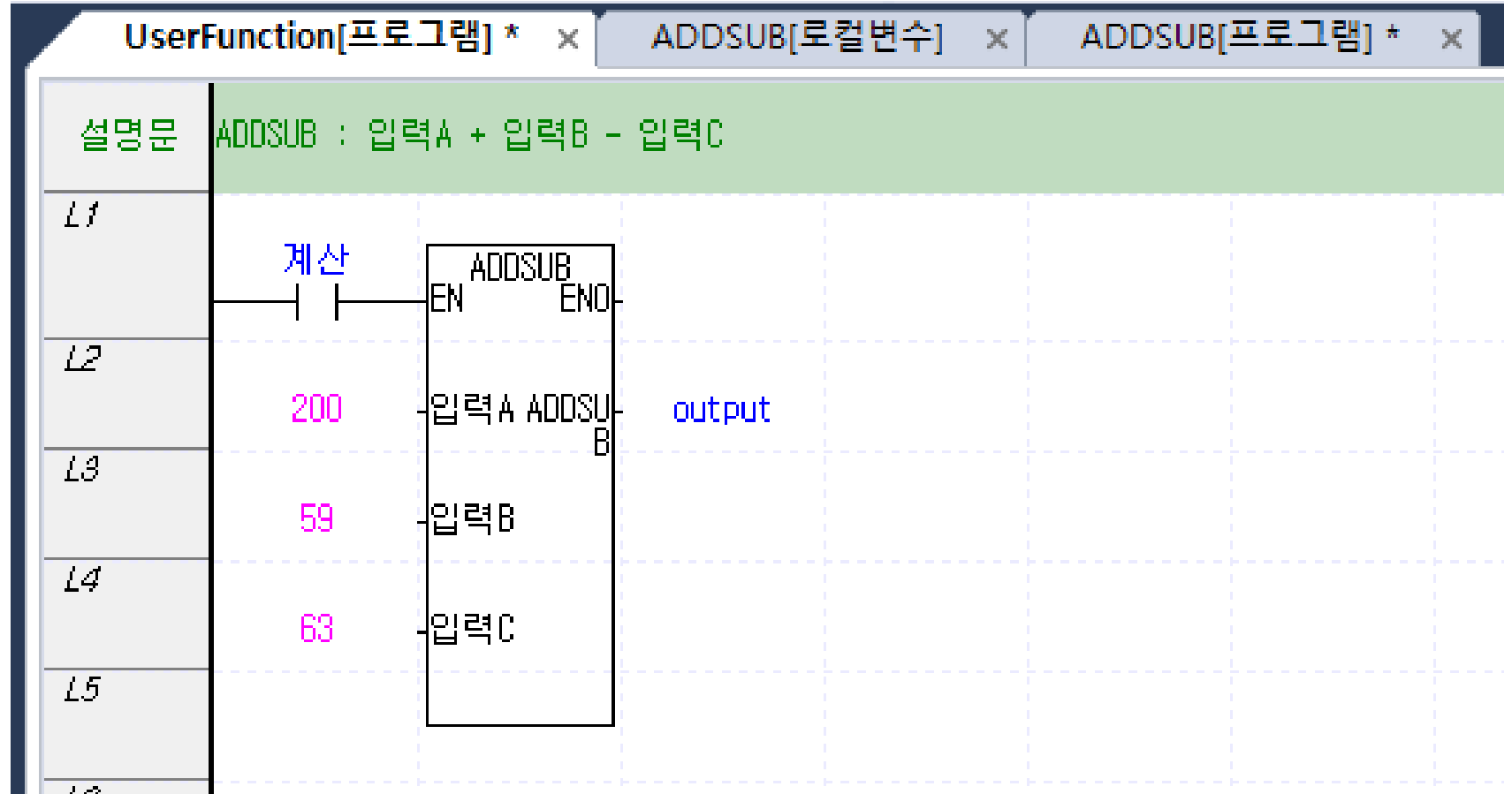
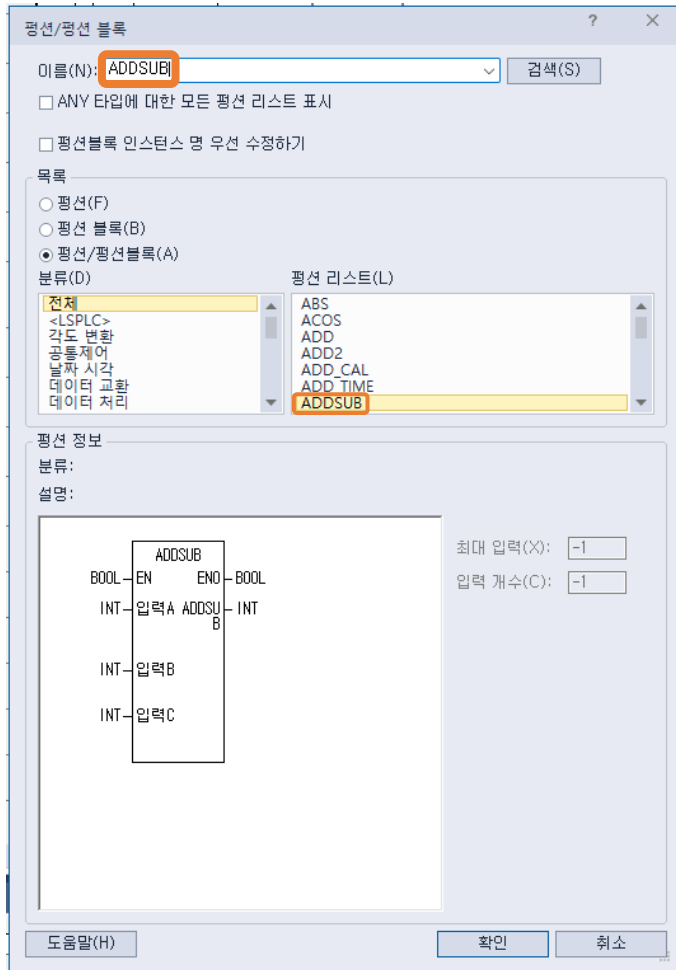


- 앞에서 설정한 입출력 변수를 사용하여 프로그램 작성 가능
- PLC에 내장된 기본 평선도 사용 가능(평선블록은x)



# 사용자 정의 평선

- 프로그램에서 불러와서 사용 가능

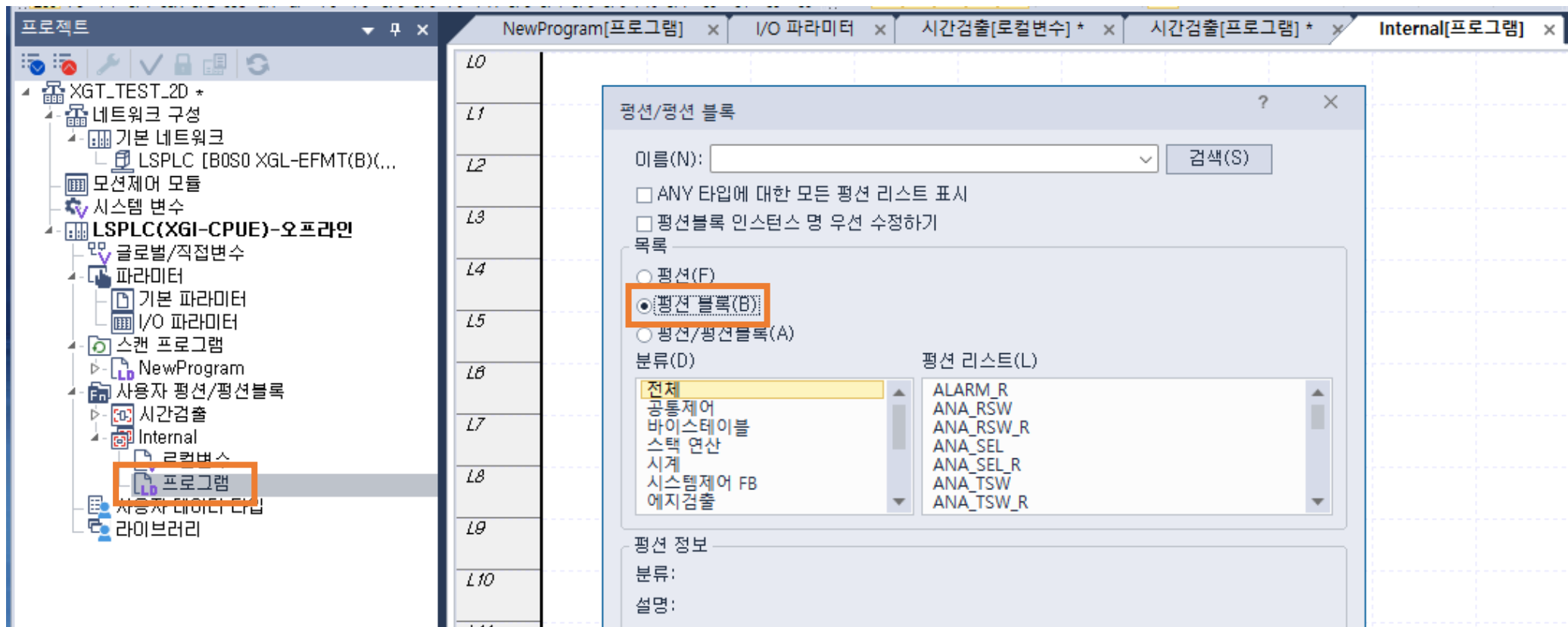


# 실습1. 피타고라스 정리

- PITA 라는 이름의 사용자 정의 평션 만들기
- A변, B변이라는 입력에 값을 넣으면, 피타고라스 정리에 의해 Result라는 결과 값을 출력
  - ※ 참고 : 피타고라스 정리  
- 밑변 a, 높이 b인 직각삼각형의 빗변 c를 구할 때
$$a^2 + b^2 = c^2$$
- 데이터 타입은 모두 Real로 진행
- SQRT(루트) 평션 사용
- 결과 예시 : A변 = 3, B변 = 4 입력 → Result = 5 출력

# 사용자 정의 평선 블록

- 사용자 정의 평선과 동일하나 프로그램 작성시 **내장된 평선 블록**도 사용 가능

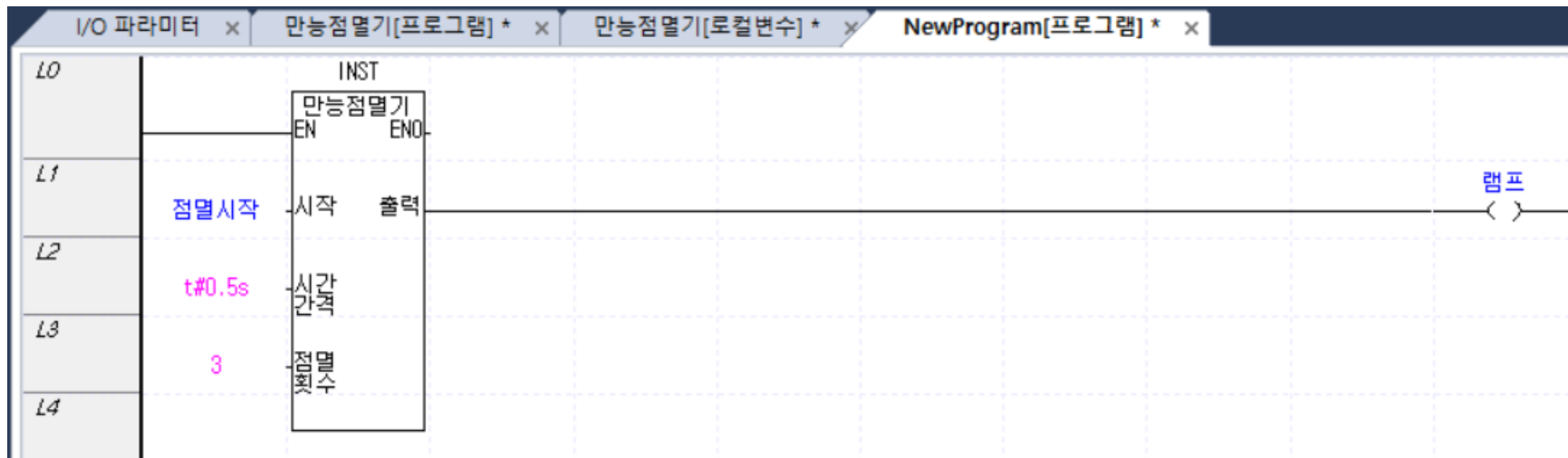


## 실습2. 사칙연산

- “사칙연산”이라는 이름의 평션 블록을 생성
- 입력A, 입력B를 정의하고
- 두입력에 대해 사칙연산을 각각 적용한 결과를 출력하는 평션 블록을 정의
- 입/출력 타입은 INT로

# 실습3. 만능점멸기

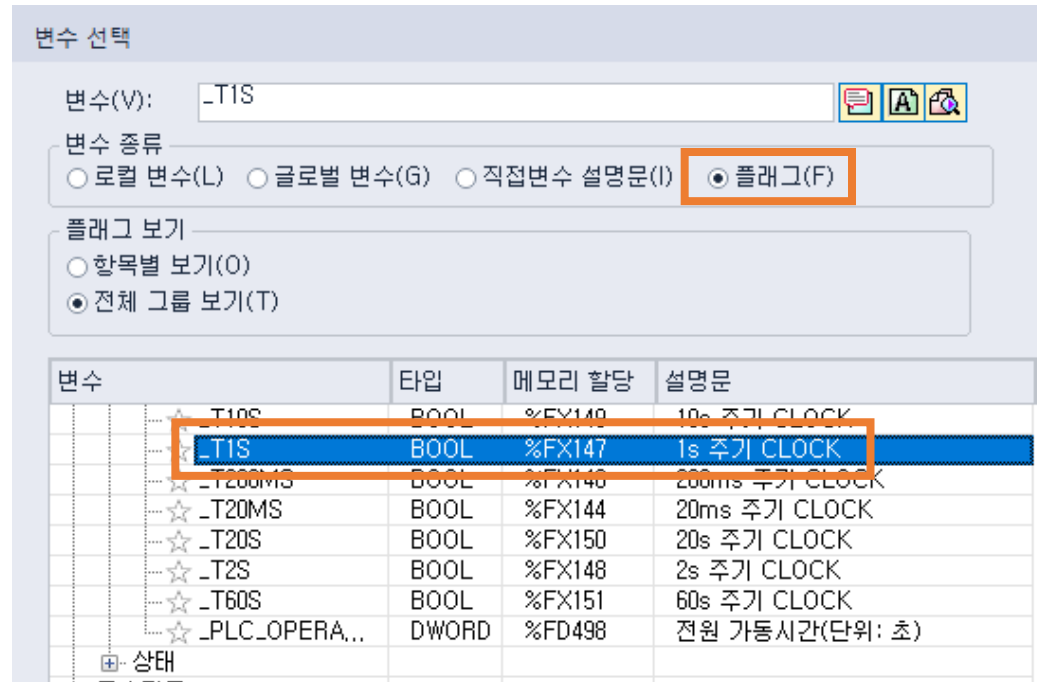
- "만능점멸기"라는 이름의 평선블록을 생성
- 아래와 같이 인풋을 정의하고
  - 시작, 시간간격, 점멸횟수
- 정의된 점멸 횟수만큼 1이 출력되도록 구현
- 예를 들어 평선블록에 램프를 연결하면 점멸 횟수만큼 램프가 깜빡여야 함



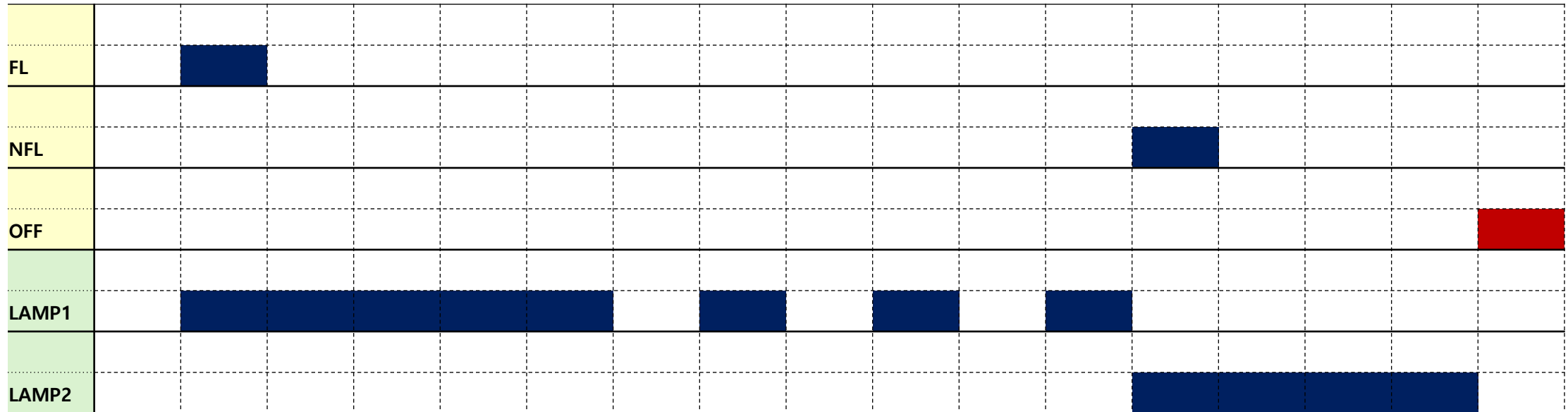
# 실습4. 깜박이 평선 블록 제작

## ■ 3개 입력, 2개 출력을 가지는 평선 블록

- 입력 FL : LAMP1이 5초 동안 켜지고, 이후 1초 단위로 깜박임
- 입력 NFL : LAMP1이 꺼지고, LAMP2가 켜짐
- 입력 OFF : 모두 꺼짐
- 출력 LAMP1
- 출력 LAMP2
- **\_T1S 플래그를 사용할 것**



# 실습4. 깜박이 평선 블록 제작



- "FL" 신호 입력 → LA1 5초간 켜지고, 5초 이후에 1초 주기로 깜빡 거림
- "NFL" 신호 입력 → 켜졌던 LA1이 꺼지고, LA2가 On된다.
- "OFF" 신호 입력 → 모든 출력값(LA1, LA2)이 OFF
- "EN" 과 "ENO"는 입/출력 모션과 연결되는 변수 (기본제공)

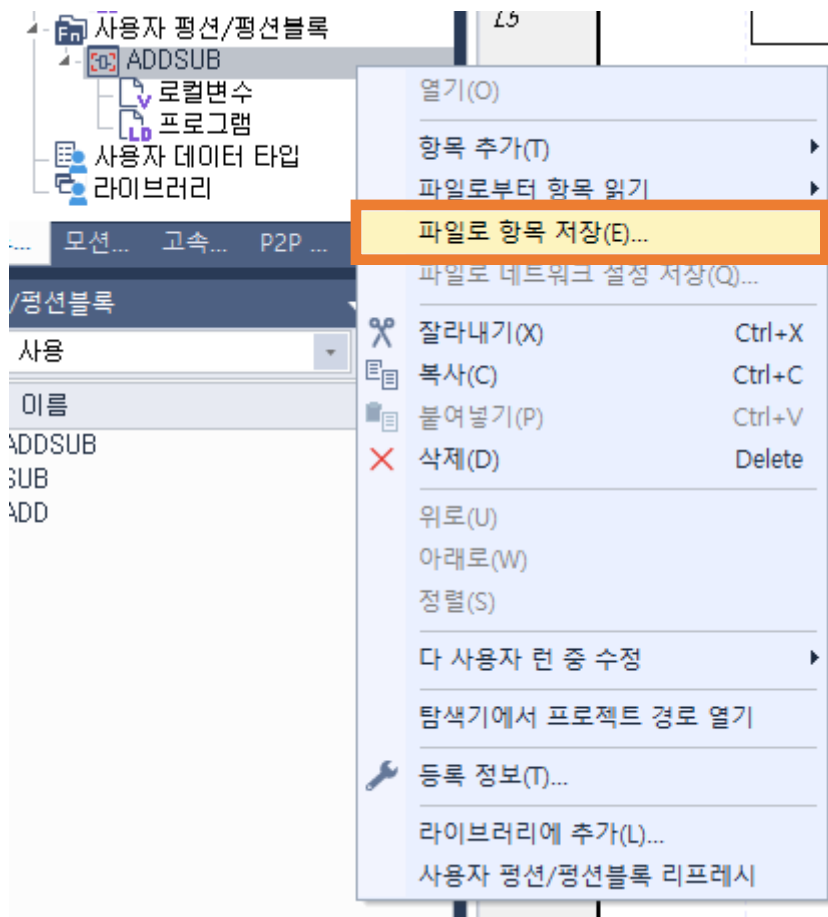
# 다른 프로젝트에서 불러오기

---



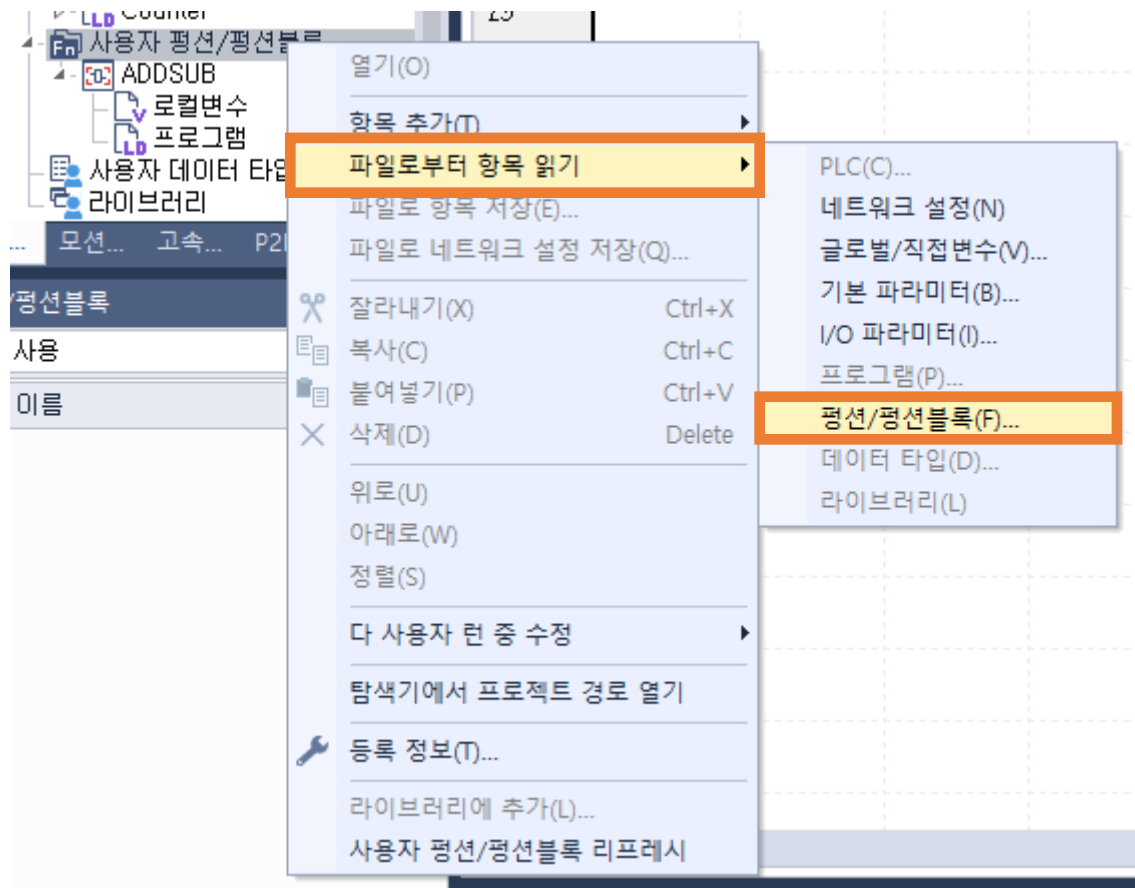
# 다른 프로젝트에서 불러오기

## ■ 파일로 함수 저장



# 다른 프로젝트에서 불러오기

- 다른 프로젝트에서 불러오기



# 라이브러리 만들기

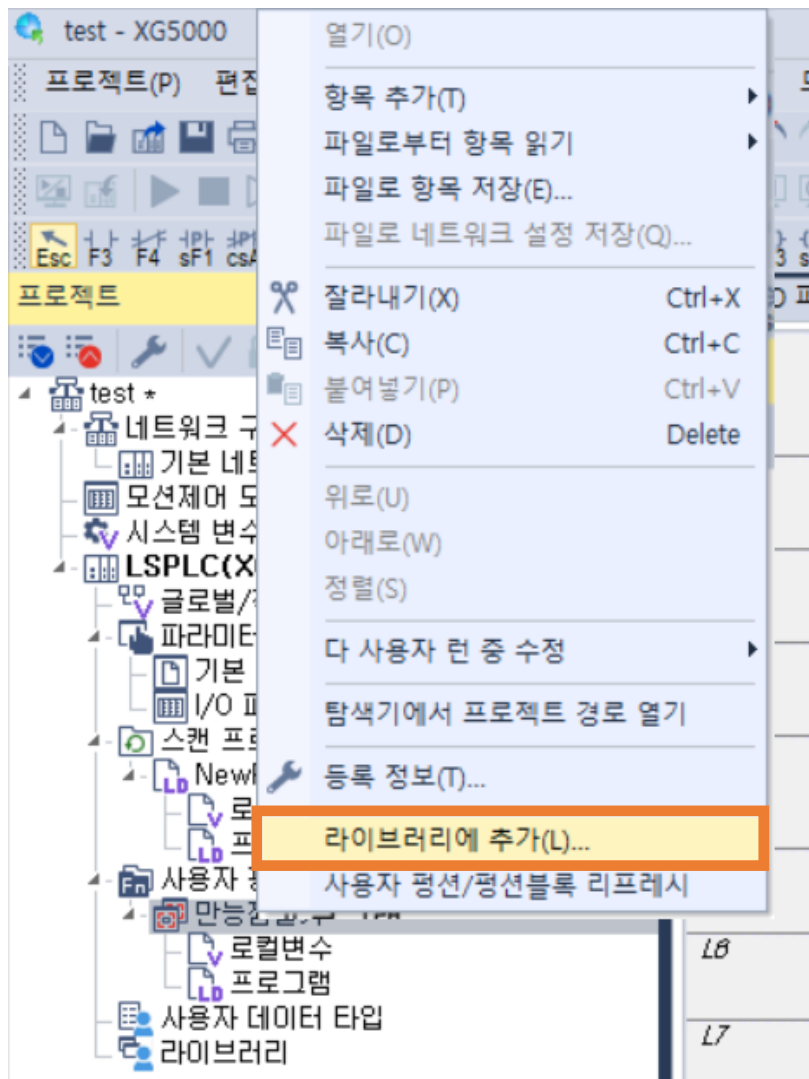
---

# 라이브러리 만들기

## ■ 라이브러리

- 자주 사용하는 기능이나 코드를 모듈화하여 저장하고 재사용할 수 있도록 하는 기능
- 개발 효율성을 높이고, 코드 일관성을 유지하며, 반복 작업을 최소화할 수 있도록 돕는 기능
- 내부 구현을 볼 수 없기 때문에 보안 유지가 가능함

# 라이브러리 만들기



# 라이브러리 만들기



# 라이브러리 만들기

라이브러리

라이브러리 이름(N):

경로(P):  ...

PLC 종류(T):

작성자(W):

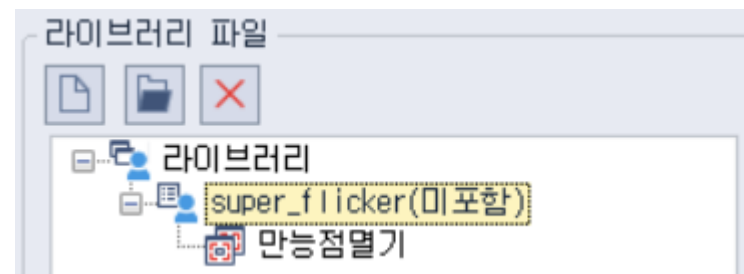
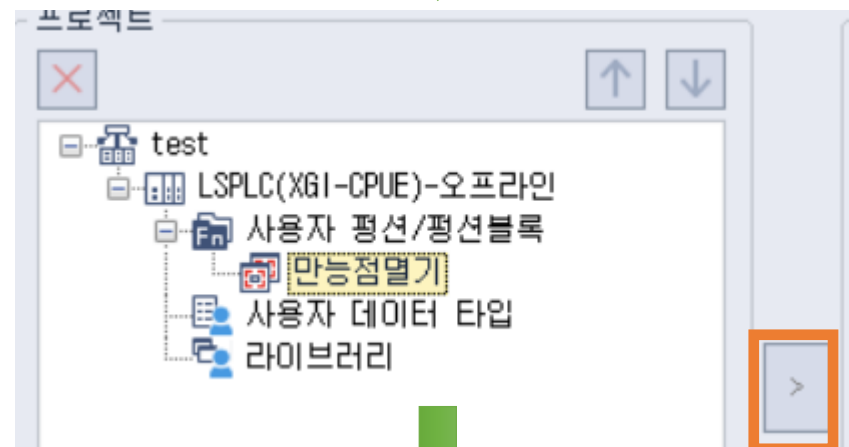
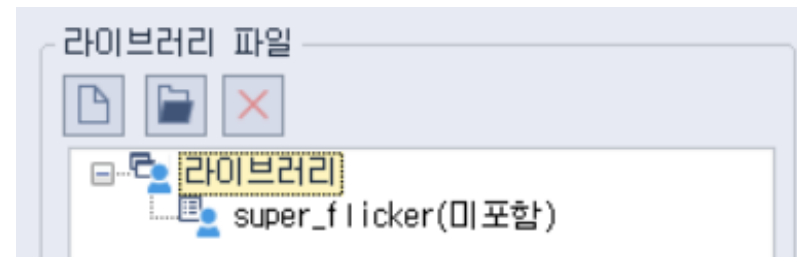
설명(C):

소스 보호

☐ 라이브러리에 소스 포함(I)

\* 해제시 실행코드만 생성됩니다.  
소스 공유가 필요없는 경우 해제하시기 바랍니다.

확인 취소



# 라이브러리 만들기

라이브러리

라이브러리 이름(N):

경로(P):  ...

PLC 종류(T):

작성자(W):

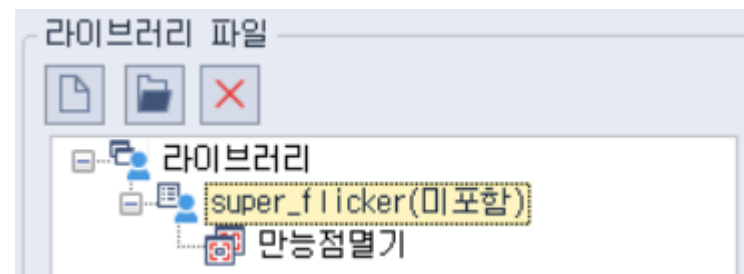
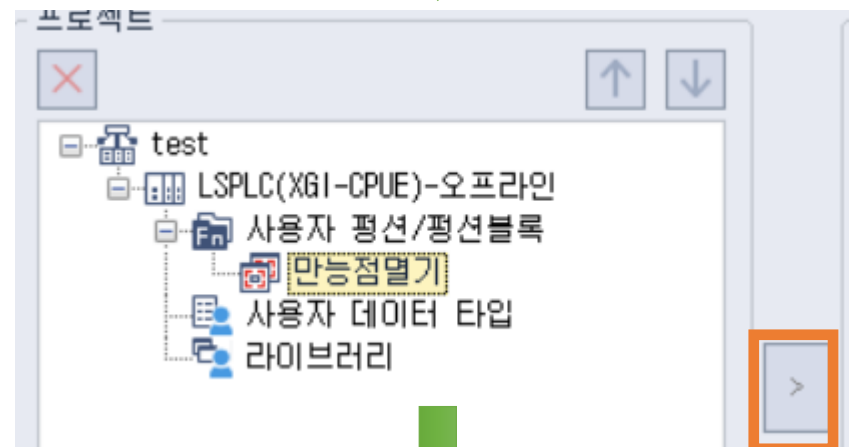
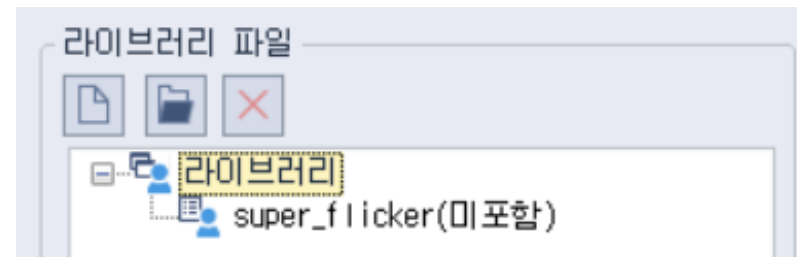
설명(C):

소스 보호

☐ 라이브러리에 소스 포함(I)

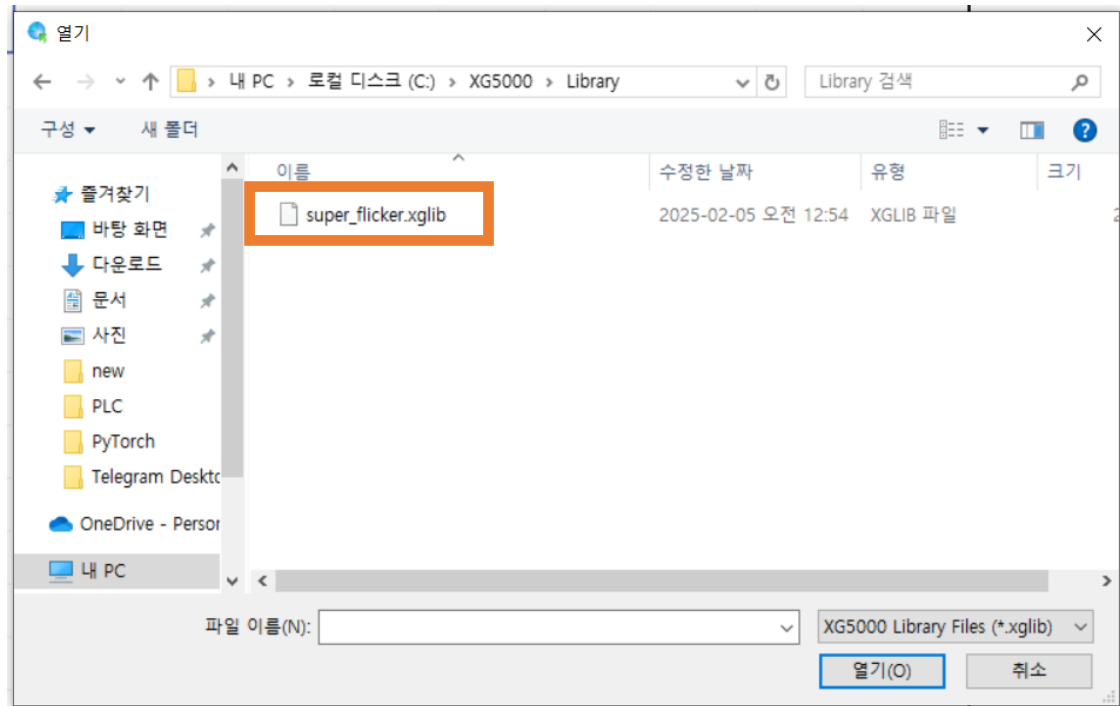
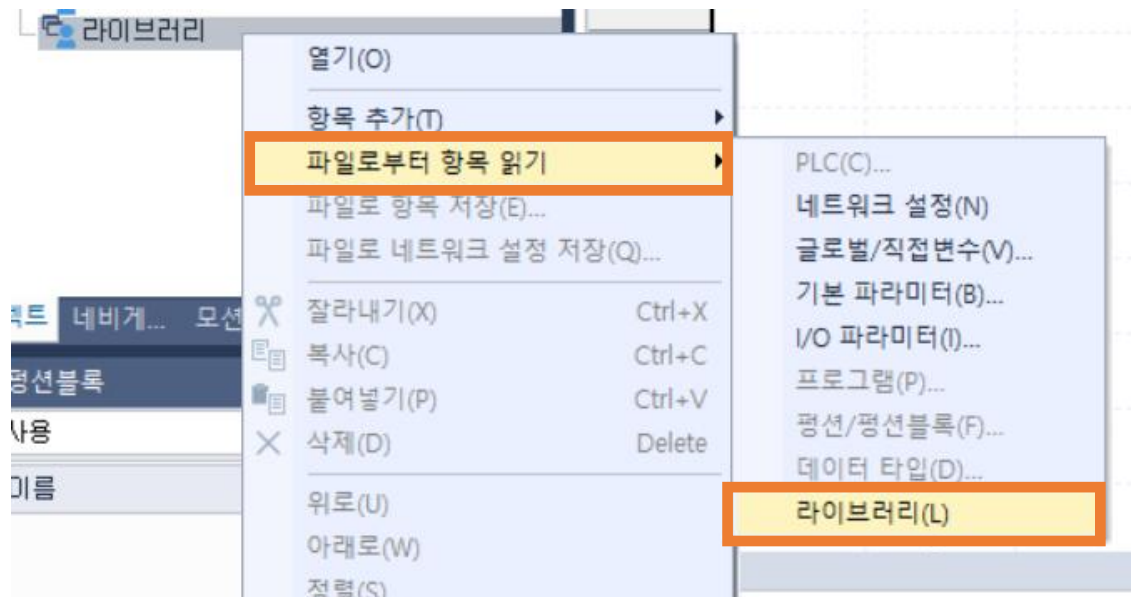
\* 해제시 실행코드만 생성됩니다.  
소스 공유가 필요없는 경우 해제하시기 바랍니다.

확인 취소

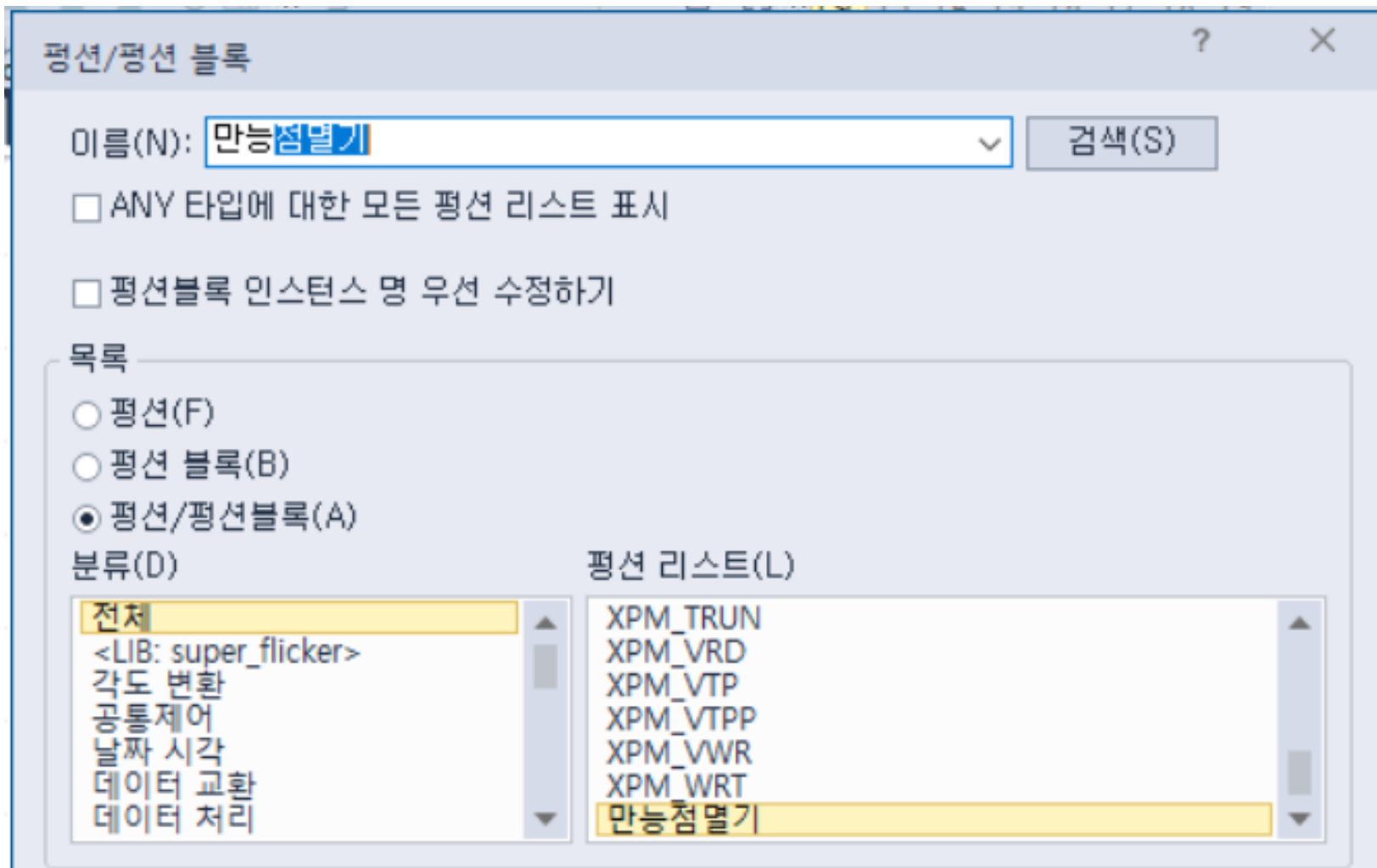




# 라이브러리 불러오기



# 라이브러리 불러오기



감사합니다

---