

CODINGO x **posco**

K-Digital Training 스마트 팩토리 3기

평선블록2: 카운터

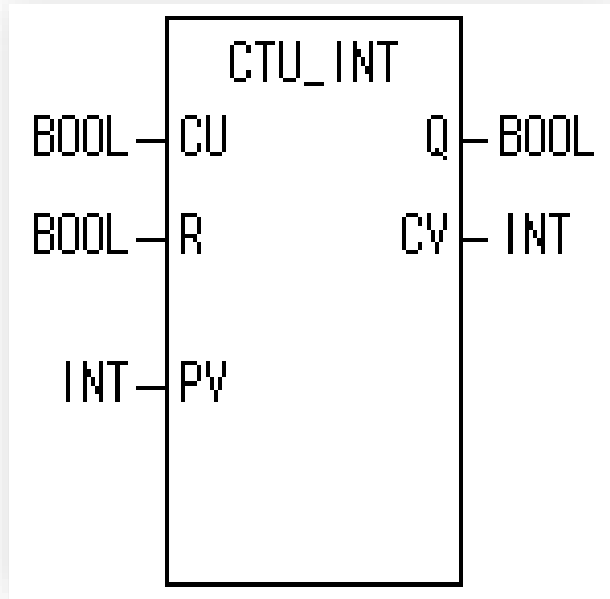
CTU_***/CTD_***/CTUD_***/CTR/..

*** : INT, DINT,LINT, UINT ,UDINT, ULINT

횟수를 측정하는데 사용됨

* 카운터 최대값: 자료형별로 상이

CTU_*** (count up)

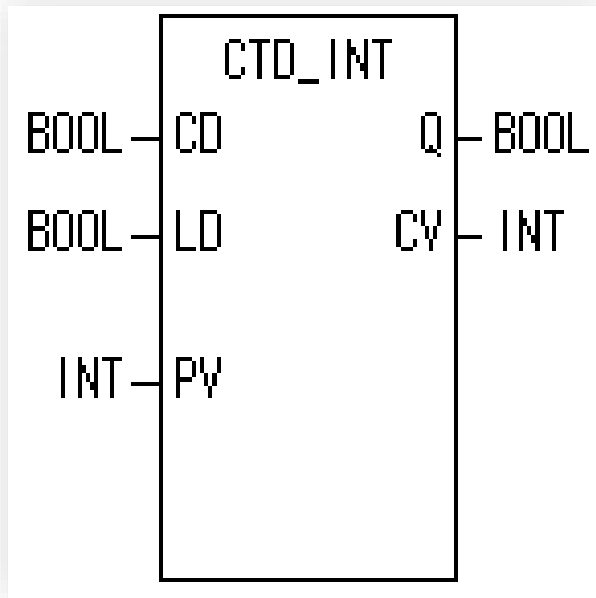


입력값이 1일 때,
입력조건의 정수값 이상으로
카운트 되면 출력으로 1전달

- 입력
 - CU: 카운터 기동 조건
 - R: 리셋 조건
 - PV : 설정한 횟수 이상(이하)일 때, Q에 1 전달
- 출력
 - Q : 카운터 접점 출력
 - CV: 카운트 횟수

- **CU** 0 >> 1이 되는 횟수를 **CV**로 출력
- **PV**에 설정한 수 이상일 때, **Q**로 1값 전달
- **R** : CV값 리셋, 따라서 Q값도 1 >> 0 으로 변경됨

CTD_*** (count down)

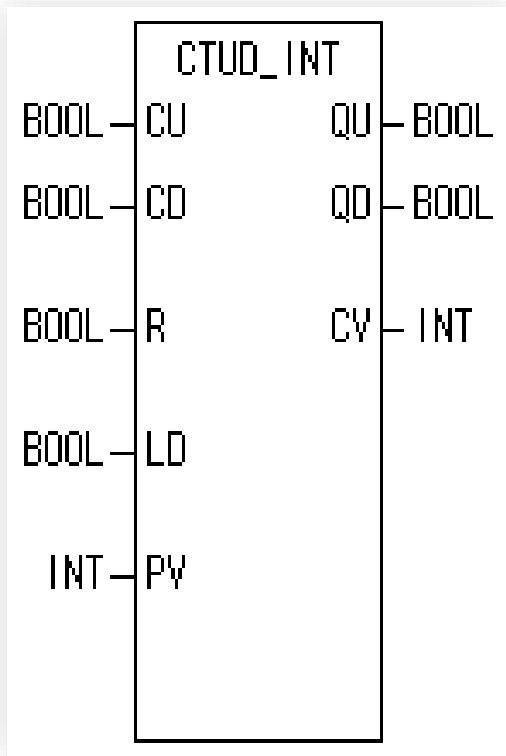


- 입력
 - CD: 카운터 기동 조건
 - LD: CV에 PV값 로드
 - PV : 시작 숫자
- 출력
 - Q : CV가 0이하일 때 1
 - CV: 카운트 횟수

감산 카운터.
입력값이 1일 때, CV가 0이라면
카운트 되면 출력으로 1전달

- **CD** 0>>1이 되는 횟수에 따라서 **CV** 1 감소
- **CV**가 0이하일 때, **Q**로 1값 전달
- **LD** : 1일 때 CV값에 PV값을 넣는다.(CV=PV)

CTUD_*** (count up/down)



• 입력

- CU: UP 카운터 기동 조건
- CD: DOWN 카운터 기동 조건
- R : up 리셋
- LD : $PV = CV$ 로
- PV :기준 숫자

• 출력

- QU: up카운트 에 대해서 출력
- QD: down카운트 조건에 대한 출력
- CV: up/down 되는 숫자 출력

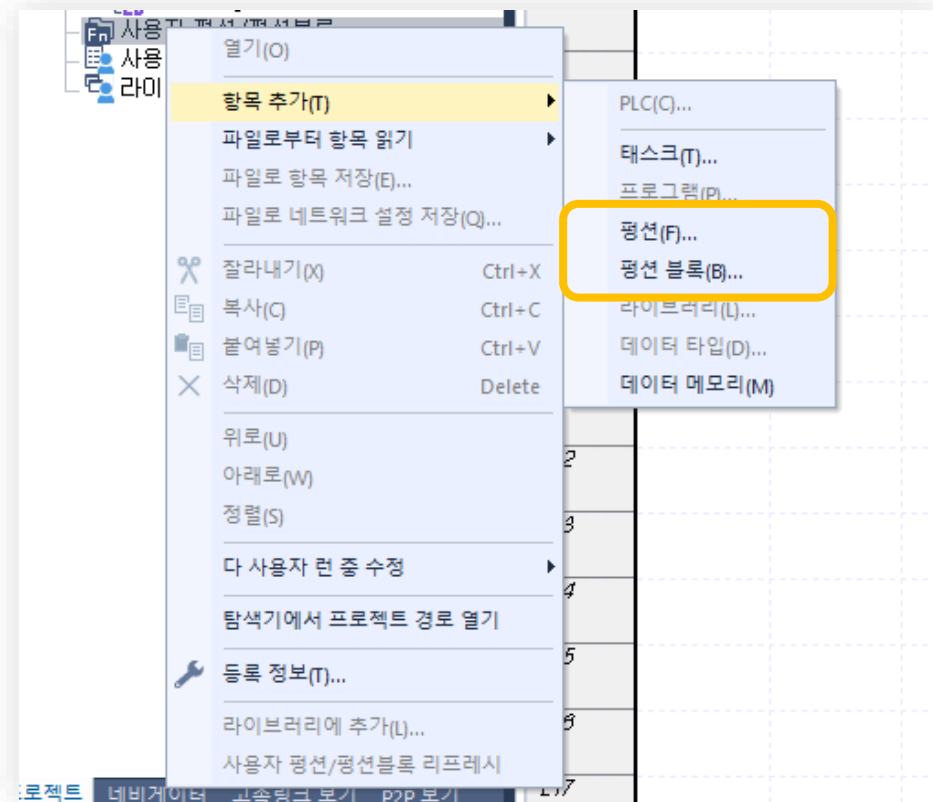
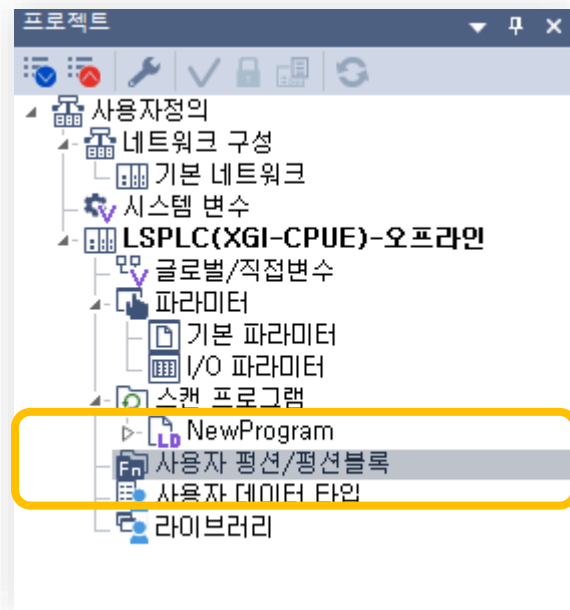
CTU + CTD

사용자 정의 평션/평션 블록

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



사용자 정의 : 이름과 데이터타입, 리턴타입

사용자 평선/평선블록

사용자 평선/평선블록 비밀번호 고급 설정

이름(N):
시간검출

언어(L):
LD

☒ EN/ENO 사용

리턴 데이터 타입(R): ARRAY

너비 컬럼 수(W): 1 너비 최적화

설명문(C):

확인 취소

배열 선택

배열 차수(I)
☒ ARRAY[0.. 5] OF
☐ ARRAY[0.. 1].. [1] OF
☐ ARRAY[0.. 1].. [1].. [1] OF

배열 타입(T)
WORD

확인 취소

만들어줄 평선의 기능별로 리턴 데이터 타입이
다르겠죠~?

내부 사용 변수 선언을 하면 최종 평선 모양이 나와요.

The screenshot displays the XG5000 software interface. On the left, the '프로젝트' (Project) tree shows the hierarchy: '사용자정의 *' (User-defined) > '네트워크 구성' (Network configuration) > '시스템 변수' (System variables) > 'LSPLC(XGI-CPUE)-오프라인' (LSPLC(XGI-CPUE)-Offline) > '글로벌/직접변수' (Global/Direct variable) > '파라미터' (Parameter) > '스캔 프로그램' (Scan program) > 'NewProgram' > '로컬변수' (Local variable) > '시간검출' (Time detection) > '로컬변수' (Local variable). The main window shows a table of variable declarations for '시간검출[로컬변수]' (Time detection [Local variable]).

	변수 종류	변수	타입	사용 유무	설명문
1	VAR_RETURN	시간검출	ARRAY[0..5]	<input type="checkbox"/>	
2	VAR_INPUT	_RTC_TIME_0	BYTE	<input type="checkbox"/>	_RTC_TIME[0]
3	VAR_INPUT	_RTC_TIME_1	BYTE	<input type="checkbox"/>	
4	VAR_INPUT	_RTC_TIME_2	BYTE	<input type="checkbox"/>	
5	VAR_INPUT	_RTC_TIME_3	BYTE	<input type="checkbox"/>	
6	VAR_INPUT	_RTC_TIME_4	BYTE	<input type="checkbox"/>	
7	VAR_INPUT	_RTC_TIME_5	BYTE	<input type="checkbox"/>	

An orange arrow points from the '_RTC_TIME_0' variable in the table to a ladder logic diagram on the right. The diagram shows a series of 'RTC_TIME' variables connected in a sequence, with 'EN' and 'ENO' labels. The final output is labeled 'ARRAY[6] OF WORD'.

내부에서 사용하는 변수의 이름은 자유롭게 지어주셔도 됩니다!
(XG5000 내부에서 미리 사용하고 있는 이름이 아니라면)

사용자정의의 - XG5000

프로젝트(P) 편집(E) 찾기/바꾸기(F) 보기(V) 온라인(O) 모니터(M) 디버그(D) 도구(T) 창(W) 도움말(H)

시간검출[프로그램] x

정의할 평선에 대해서
내용을 작성하는 화면이 나오면
해당 화면에 내용을 작성하면 돼
요!

모니터 1

	값	타입	디바이스/변수	설명문
1				

프로그램 검사

오류 0개 경고 0개 메시지 0개

결과 프로그램... 찾기 1 찾기 2 통신 메모리 ... 사용된 ... 이종 코...

프로젝트 네비게이터 고속링크 보기 p2p 보기

평선/평선블록

최근 사용 편집

LSPLC

런

행 17, 열 2

견침

100%

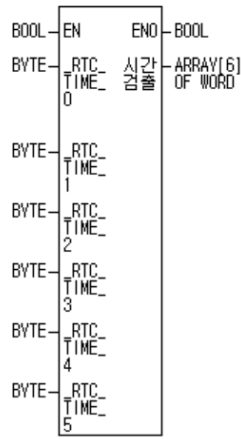
현재 날짜 검출

- 트리거 조건 발생 시 현재 시간(년,월,일,시,분,초)을 워드타입으로 지정한 디바이스에 저장하는 펄션 작성
 - 트리거 조건: 함수를 동작하게 하는 조건입니다.
- _RTC_TIME[0~5] : 현재 년,월,일,시,분,초 를 나타내는 플래그 변수 이용

시간 검출 로컬 변수와 평선 모양

	변수 종류	변수	타입	사용 유무
1	VAR_INPUT	_RTC_TIME_0	BYTE	<input checked="" type="checkbox"/>
2	VAR_INPUT	_RTC_TIME_1	BYTE	<input checked="" type="checkbox"/>
3	VAR_INPUT	_RTC_TIME_2	BYTE	<input checked="" type="checkbox"/>
4	VAR_INPUT	_RTC_TIME_3	BYTE	<input checked="" type="checkbox"/>
5	VAR_INPUT	_RTC_TIME_4	BYTE	<input checked="" type="checkbox"/>
6	VAR_INPUT	_RTC_TIME_5	BYTE	<input checked="" type="checkbox"/>
7	VAR_RETURN	시간검출	ARRAY[0..5]	<input checked="" type="checkbox"/>
8				<input type="checkbox"/>

선언시 설정해둔 리턴 값의 변수와 타입이기 때문에 미리 들어가 있습니다.
평선의 리턴값은 "최종 출력될 데이터"



입력하고자 하는 기본 플래그 변수인 **_RTC_TIME[0~7]**은 **Byte**

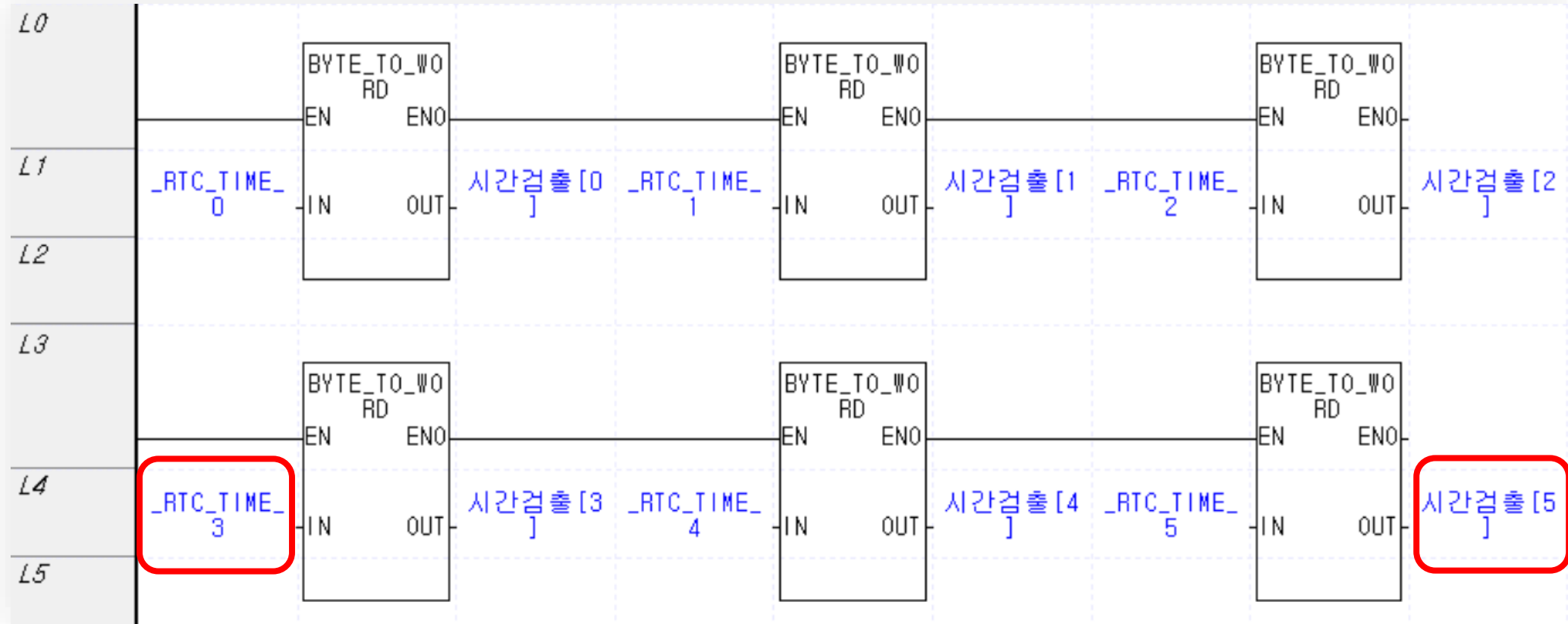
타입,

출력 데이터("시간검출"이라는 변수) 타입은 **WORD**이기 때문에

형변환이 필요해요!

(_RTC_TIME[6](요일),_RTC_TIME[7](연대) 은 사용하고 있지 않고 있습니다.)

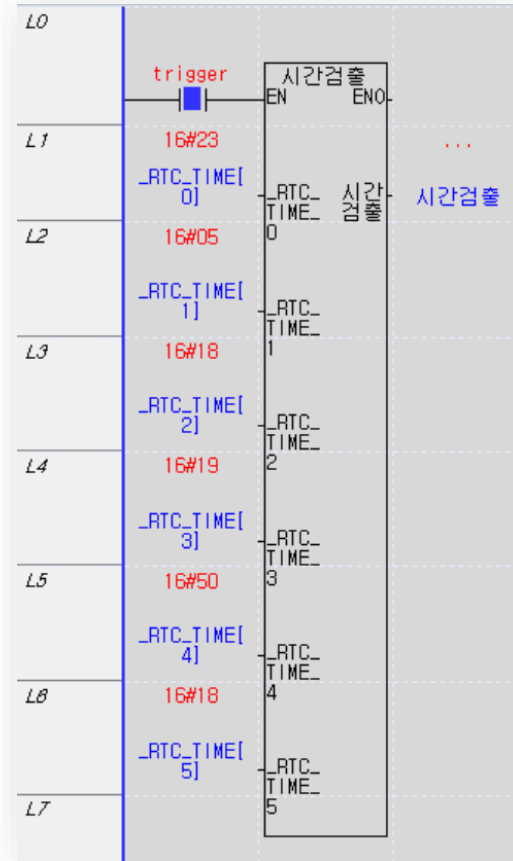
시간 검출 평선 내용



평선[로컬변수] 설정시
미리 선언해둔 VAR_INPUT,(BYTE 형)

선언할 때 미리 설정해준 리턴인
WORD 형 Array[0..5] 가
출력 데이터가 됩니다.

스캔프로그램에서 “시간검출” 평션 사용

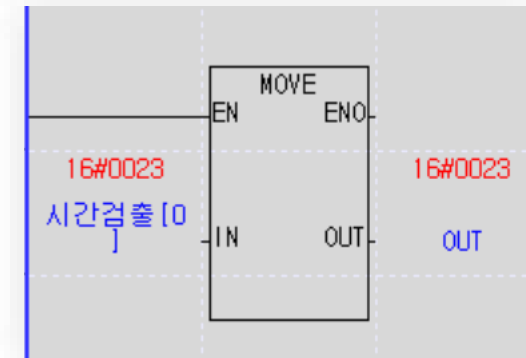


→ Byte 타입 플래그 변수 **_RTC_TIME[0~5]**이 입력되면
시간검출[5] 배열에 **Word** 형으로 형 변환되어서 출력됩니다.

모니터 4					
	PLC	프로그램	변수/디바이스	값	타입
1	LSPLC	사용자_정의	시간검출		ARRAY[0]
2			시간검출[0]	16#0023	WORD
3			시간검출[1]	16#0005	WORD
4			시간검출[2]	16#0018	WORD
5			시간검출[3]	16#0019	WORD
6			시간검출[4]	16#0057	WORD
7			시간검출[5]	16#0052	WORD

프로그램 하단의 모니터에서 배열 값들을
확인할 수 있어요.

“시간검출” 출력 선택 후
[모니터 → 현재 변수 모니터에 추가]



MOVE 평션을 이용하면 시간검출[0...5]에
어떤 배열이 들어있는지도 확인 할 수 있습니다.

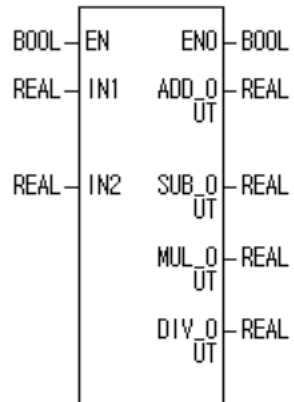
단, MOVE의 IN과 OUT은
WORD 타입으로 설정해 줘야겠네요!!

사칙연산

- 숫자 두 개를 입력 받아서 모든 사칙 연산을 한 번에 수행하는
평션 블록 작성

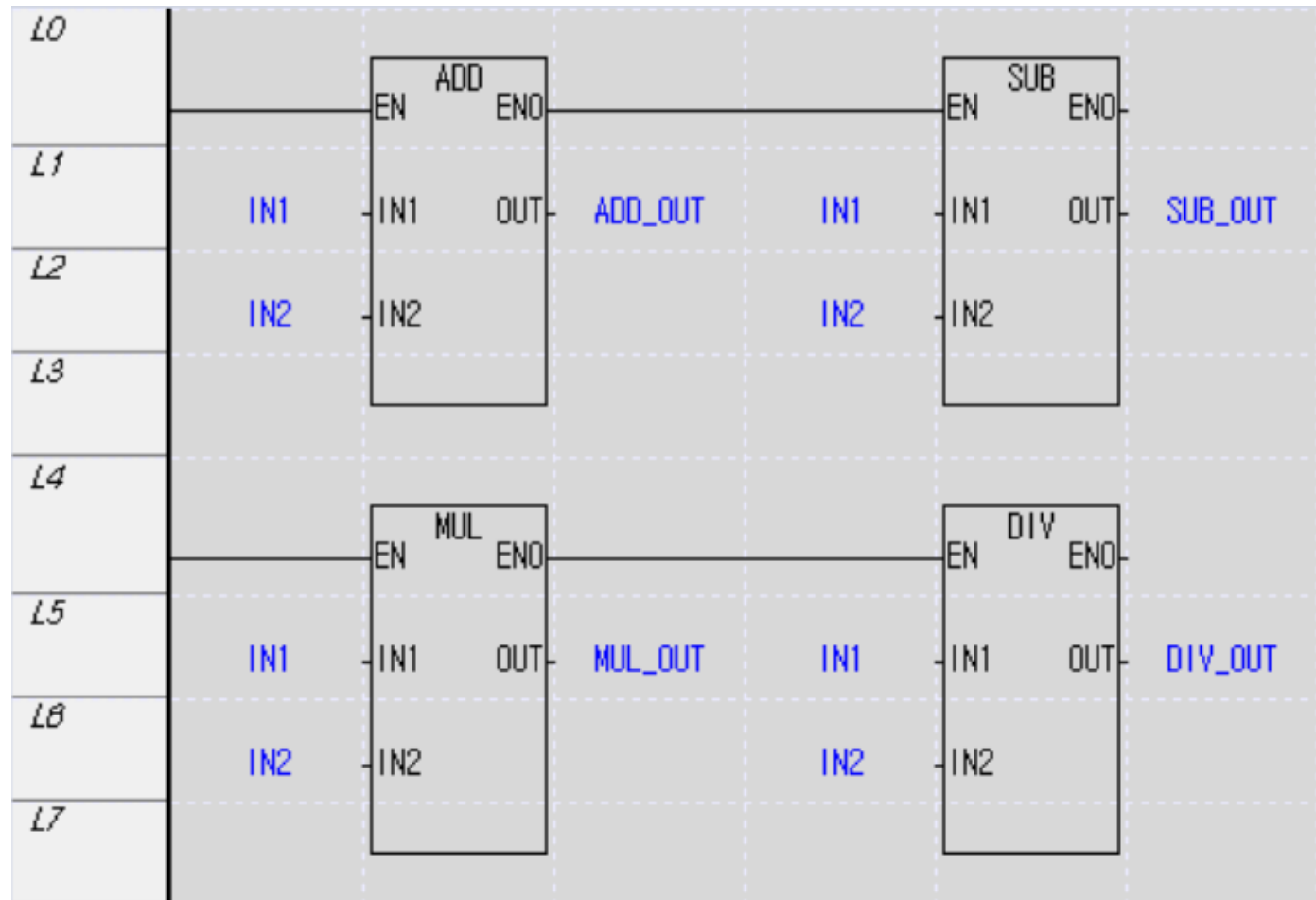
“사칙연산” 변수 설정

	변수 종류	변수	타입
1	VAR_INPUT	IN1	REAL
2	VAR_INPUT	IN2	REAL
3	VAR_OUTPUT	ADD_OUT	REAL
4	VAR_OUTPUT	SUB_OUT	REAL
5	VAR_OUTPUT	MUL_OUT	REAL
6	VAR_OUTPUT	DIV_OUT	REAL

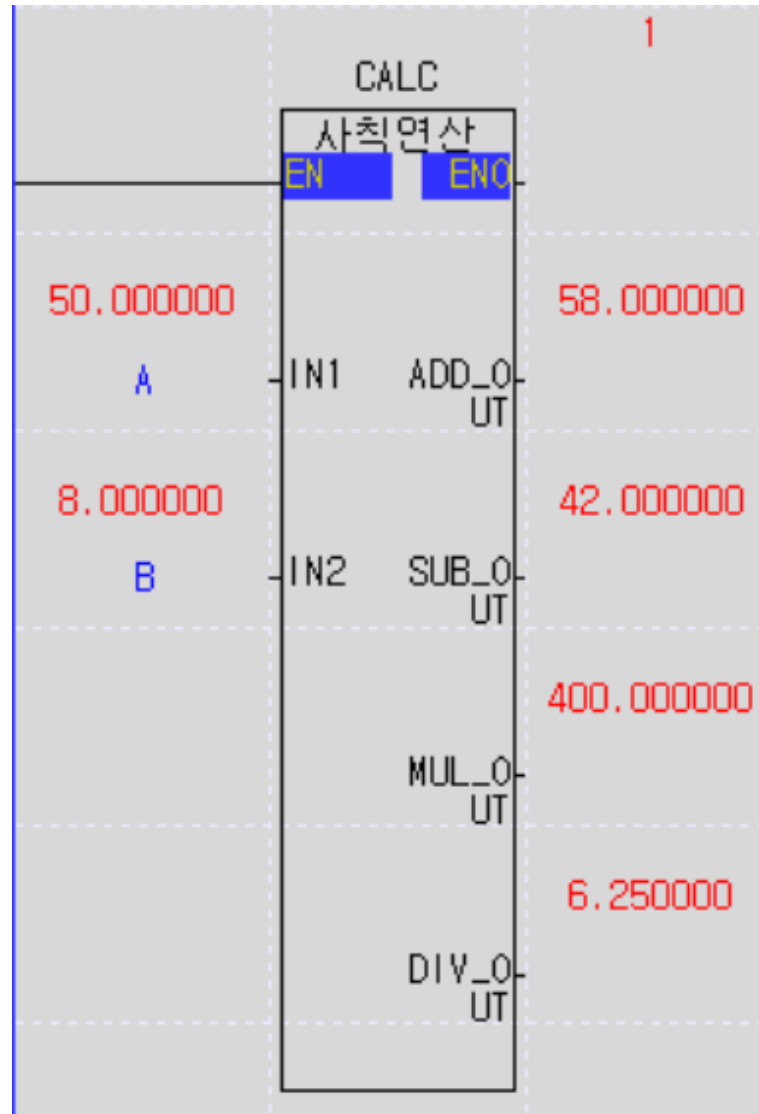


간단한 프로그램이지만
출력이 2개 이상이기 때문에 평선 블록으로 작성

“사칙연산” 내부 로직



“사칙연산” 스캔 프로그램에서 사용



실습1, 사용자 정의 "평션"

- PITA 라는 이름의 사용자 정의 **평션** 만들기
- A변, B변이라는 입력에 값을 넣으면, 피타고라스 정리에 의해 Result라는 결과 값을 출력
- 데이터 타입은 모두 Real로 진행
- 정의 완료하고 사용해보기!!
 - A변 = 3/ B변 = 4 → Result에 5가 출력이 되는지 확인해보기!
- SQRT(루트) 평션 사용

실습2, 사용자 정의 "펄스 블록"

- "FL" 신호 입력 → LA1 5초간 켜지고, 5초 이후에 1초 주기로 깜빡거림
 - "NFL" 신호 입력 → 켜졌던 LA1이 꺼지고, LA2가 On된다.
 - "OFF" 신호 입력 → 모든 출력값(LA1, LA2)이 OFF
 - "EN" 과 "ENO"는 입/출력 모선과 연결되는 변수 (기본제공)
-
- L1과 L2는 각각 램프라고 생각하면 되겠지요?
 - 꼭 위의 변수 이름은 지키지 않아도 됩니다.
 - 인스턴스명은 자유롭게 설정해 주세요.