

codingOn x posco
K-Digital Training 스마트 팩토리 단기 8기

2. 래더 프로그래밍 기초



2-1. PLC CPU

PLC의 CPU

- LS Electric의 CPU에는 여러가지 종류의 CPU가 있음
- LS Electric의 CPU Series
 - XGT / XGB / XGS / XGR
 - XGT : 중대형 컨트롤러 → 수업에서 사용할 제품군
 - XGB : 소형 컨트롤러
 - XGS : 세이프티 컨트롤러
 - XGR : 이중화 PLC

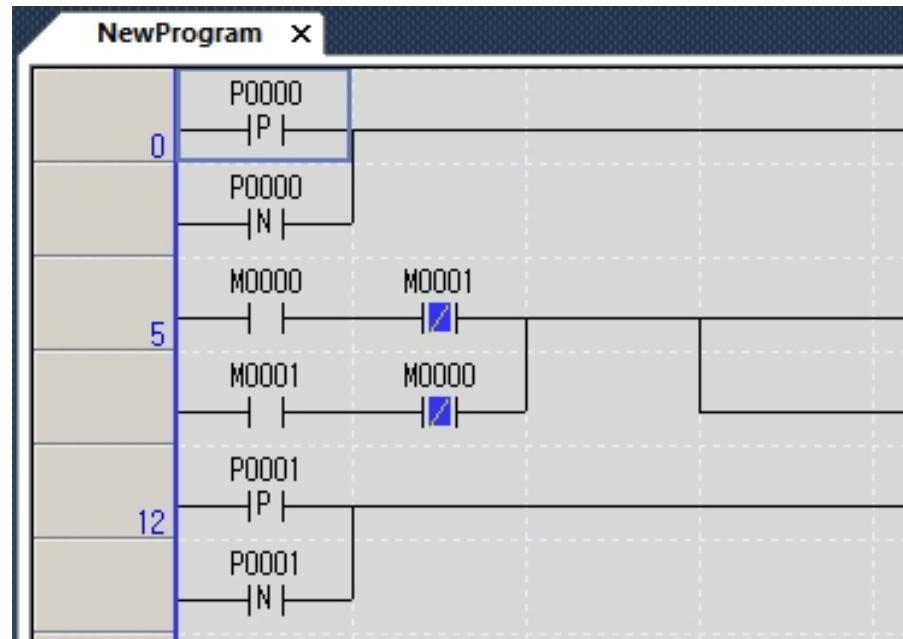
PLC의 CPU

• XGT와 XGB 비교

구분	XGT	XGB
목적	대규모 시스템, 복잡한 제어 작업	소규모 시스템, 기본 제어 작업
성능	고성능 (고속 처리 및 대용량 지원)	중간 성능 (단순 로직 및 제어에 적합)
확장성	뛰어난 확장성 (모듈형 구조)	제한적 확장성 (블록형 구조)
비용	고가 (대규모 프로젝트에 적합)	저가 (소규모 프로젝트에 적합)
사용 사례	스마트 팩토리, 공장 자동화, 프로세스 제어	소형 생산 라인, 간단한 장치 제어

PLC의 CPU

- PLC는 CPU 종류에 따라 프로그래밍 하는 방식이 다름



XGB, XGK 시리즈 CPU



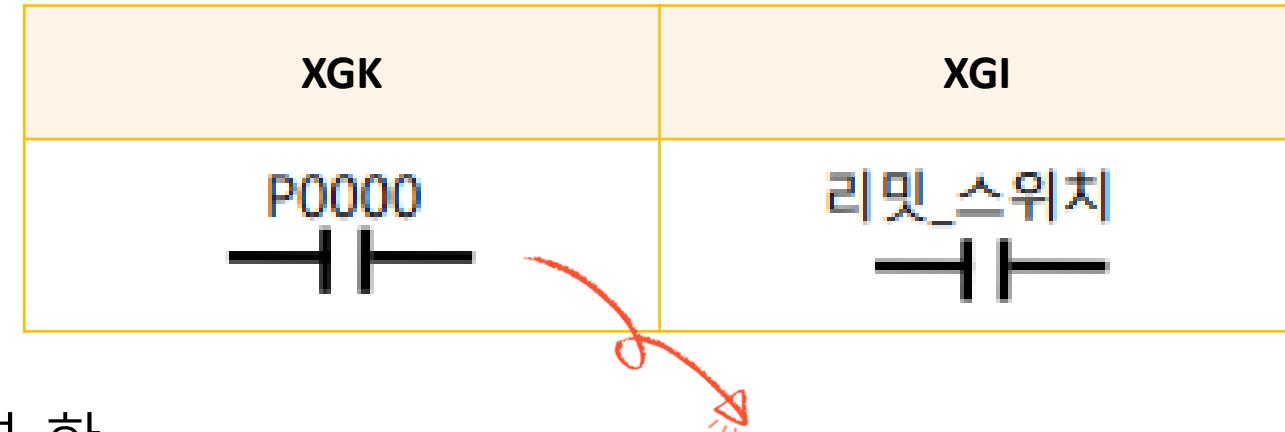
XGI 시리즈 CPU

PLC의 CPU: XGT-XGI

- XGT-XGI

- IEC 국제 표준 규격을 따르는 PLC
- 국제 규격의 통신 프로토콜
- 윈도우 환경의 프로그래밍 툴 지원
- 프로그램 작성 용이
(입출력 식별자명: 실제 접속되는 기기명-한글, 한자, 영어 가능-으로 프로그래밍 가능)
- XGI CPU 스펙 참조

<https://www.ls-electric.com/ko/product/view/P01114>



XGI 는 한글 이름으로 사용할 수 있지만
XGT 는 정해진 규칙을 따라서
식별자명을 작성해야 함

IEC 국제 표준 규격

- 정의
 - IEC 표준 규격은 전기, 전자 및 산업 자동화 기술의 국제 표준으로, 시스템의 호환성과 안정성을 보장함.
- 필요한 이유
 - 통합된 기술 표준으로 산업 환경에서의 혼란을 줄이고, 다양한 시스템 간 상호운용성을 보장하기 위해 필요함
- 효과
 - 제조사 간 호환성, 설계 효율성, 안전성을 향상시키고, 비용 절감과 확장성을 제공.

IEC 국제 표준 규격

- 특징

- 다양한 데이터 타입 지원
- 평선, 평션블록, 프로그램과 같은 구성 요소를 이용해 프로그램을 구조적으로 작성할 수 있음
- 사용자가 작성한 프로그램을 라이브러리화하여 다른 프로젝트에서 재사용 가능
- 다양한 언어 지원, 사용자는 최적의 언어를 선택해서 사용할 수 있음
 - 도형식 언어 : LD(래더), FBD
 - 문자식 언어 : IL, ST
 - SFC



2-2. 데이터 타입(자료형)



데이터 타입

구 분	예 약 어	데이터 형	크기 (비트)	범 위
수치 (ANY_NUM)	SINT	Short Integer	8	-128 ~ 127
	INT	Integer	16	-32768 ~ 32767
	DINT	Double Integer	32	-2147483648 ~ 2147483647
	LINT	Long Integer	64	$-2^{63} \sim 2^{63}-1$
	USINT	Unsigned Short Integer	8	0 ~ 255
	UINT	Unsigned Integer	16	0 ~ 65535
	UDINT	Unsigned Double Integer	32	0 ~ 4294967295
	ULINT	Unsigned Long Integer	64	$0 \sim 2^{64}-1$
	REAL	Real Numbers	32	-3.402823466e+038 ~ 1.175494351e-038 or 0 or 1.175494351e-038 ~ 3.402823466e+038
시간	LREAL	Long Reals	64	-1.7976931348623157e+308 ~ -2.2250738585072014e-308 or 0 or 2.2250738585072014e-308 ~ 1.7976931348623157e+308
	TIME	Duration	32	T#0S ~ T#49D17H2M47S295MS
날짜	DATE	Date	16	D#1984-01-01 ~ D#2163-6-6
	TIME_OF_DAY	Time Of Day	32	TOD#00:00:00 ~ TOD#23:59:59.999
	DATE_AN	Date And Time Of Day	64	DT#1984-01-01-00:00:00 ~ DT#2163-12-31-23:59:59.999
문자열	STRING	Character String	30*8	-
비트 상태 (ANY_BIT)	BOOL	Boolean	1	0, 1
	BYTE	Bit String Of Length 8	8	16#0 ~ 16#FF
	WORD	Bit String Of Length 16	16	16#0 ~ 16#FFFF
	DWORD	Bit String Of Length 32	32	16#0 ~ 16#FFFFFFFF
	LWORD	Bit String Of Length 64	64	16#0 ~ 16#FFFFFFFFFFFFFF

수치(ANY_NUM)

자료형	크기	표현 범위	주요 용도
SINT (Short)	8비트	-128 ~ 127	작은 상태 값, 간단한 계산
INT	16비트	-32,768 ~ 32,767	센서 값, 기본 크기의 정수 값
DINT (Double)	32비트	-2,147,483,648 ~ 2,147,483,647	긴 시간 값, 카운터 값
LINT (Long)	64비트	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	고정밀 데이터, 대형 카운터 값
REAL	32비트	$\pm 1.18 \times 10^{-38}$ ~ $\pm 3.4 \times 10^{38}$	물리량, 소수점 계산

- 각 키워드에 U가 앞에 붙으면 Unsigned 의 의미가 됨(부호가 없는 양의 데이터)
 - SINT(8비트)가 -128 ~ 127 까지 나타낼 수 있다면, USINT는 똑같은 비트를 가지고 있지만 0~ 255 까지 양으로 더 많은 범위를 나타낼 수 있음

시간/날짜/문자열

자료형	크기	표현 범위	주요 용도
TIME	32비트	T#0ms ~ T#24d_20h_31m_23s_647ms	타이머 설정, 경과 시간 계산
DATE	16비트	D#1990-01-01 ~ D#2089-12-31	특정 날짜 지정, 기록 관리
TIME_OF_DAY (TOD)	32비트	TOD#00:00:00 ~ TOD#23:59:59.999	하루 중 특정 시간 설정
DATE_AND_TIME (DT)	64비트	DT#1990-01-01-00:00:00 ~ DT#2089-12-31-23:59:59.999	날짜와 시간을 동시에 관리
STRING	30*8비트	-	사용자 메시지, 장비 코드/이름, 데이터 기록

비트 상태

자료형	크기	표현 범위 (양수 기준)	사용 목적
BOOL	1비트	0 또는 1	ON/OFF 상태, 플래그 비트
BYTE	8비트	0 ~ 255	디지털 입력/출력 그룹 관리
WORD	16비트	0 ~ 65,535	센서 값, 타이머 값
DWORD	32비트	0 ~ 4,294,967,295	큰 카운터 값, 시간 계산
LWORD	64비트	0 ~ 18,446,744,073,709,551,615	매우 큰 값, 복잡한 데이터 처리

데이터 표현

데이터 타입	초기값
SINT, INT, DINT, LINT	0
USINT, UINT, UDINT, ULINT	0
BOOL, BYTE, WORD, DWORD, LWORD	0
REAL, LREAL	0.0
TIME	T#0s
DATE	D#1984-01-01
TIME_OF_DAY	TOD#00:00:00
DATE_AND_TIME	DT#1984-01-01-00:00:00
STRING	" (empty string)

2-3. 메모리



메모리

구분	메모리	범위
자동 변수 영역	A	1,024KB
입력 변수	I	16KB
출력 변수	Q	16KB
직접 변수	M	512KB
	R	64KB * 16블록
	W	1,024KB
플래그 변수	F	8KB
	K	16KB
	L	22KB
	N	42KB
	U	8KB

메모리

메모리 영역	데이터 유지 여부	범위
M 영역	유지되지 않음	512KB
R 영역	유지됨	64KB * 16블록
W 영역	유지됨	1,024KB

2-4. PLC 변수



변수의 종류

항목	설명
심볼릭 변수	사용자가 이름을 직접 지어주는 변수
직접 변수	메모리 주소를 직접 작성하여 사용하는 변수
플래그	시스템에 의해 미리 선언되어 있는 변수 (시간, 통신 등)

변수의 종류

- **직접 변수**

- PLC의 입출력 또는 기억 장소에 대해서 직접적으로 표현하는 것 (기본 제공)
- 이미 지정된 식별자와 주소가 있기 때문에 변수 선언이 필요 없음
- 메모리 영역에 저장

- **심볼릭 변수**

- 사용자가 이름을 직접 지어주는 변수
- 변수 선언 필요
- 자동 변수 영역에 저장



심볼릭(Symbolic) 변수

- 메모리 주소 대신 사용자가 정의한 이름(심볼)을 사용하여 변수를 참조.
- 구성 요소
 - 변수 이름: 역할을 나타내는 사용자 정의 이름
 - 데이터 타입: BOOL, INT, REAL 등 데이터 유형.
 - 설명(Comment): 변수의 용도나 의미를 주석으로 기술.
 - 메모리 주소(Optional): 필요 시 실제 메모리 주소를 할당해서 사용



심볼릭(Symbolic) 변수

변수종류	내용
VAR	읽고 쓸 수 있는 일반적인 변수
VAR_RETAIN	정전 유지 변수
VAR_CONSTANT	읽기만 할 수 있는 변수
VAR_EXTERNAL	VAR_GLOBAL로 선언된 변수를 사용하기 위한 선언

심볼릭 변수 표현 규칙

1. 문자나 밑줄 문자(_)로 시작
2. 시작 문자 이후로는 문자, 숫자, 밑줄 문자('_) 조합으로
3. 빈 칸(Space) 포함 X
4. 문자는 한자, 영문, 한글 모두 제한 X
5. 영어일 경우 대·소문자 구별하지 않고 같은 문자면 같은 변수로 인식

직접 변수 표현 방식

- 메모리 주소를 직접 참조하여 사용하는 변수.
- 퍼센트 기호(%)로 시작하고 위치 접두어, 크기 접두어와 숫자들로 구성됨

※ 사용 예시

종류	사용 예
입력 변수	%IX0.0.0, %IB0.0.1, %IW0.0.1, %ID0.0.0
출력 변수	%QX0.1.0, %QB0.1.1, %QW0.1.1, %QD0.1.0
내부 메모리	%MX100, %MB50, %MW100, %MD100
	%MB50.3, %MW100.10, %MD100.31

변수 표현 - 입출력 메모리 할당

① 위치 접두어

변수나 메모리 주소가 어떤 역할을 하는지 나타내기 위해

사용되는 기호

% I X 0.0.0

(1)

(2)

(3)

(4)

(5)

접두어	의미
I	입력 위치 (Input Location)
Q	출력 위치 (Output Location)

%IX0.0.0 입력 위치에 대한 변수 표현

%QX0.0.0 출력 위치에 대한 변수 표
현

변수 표현 - 입출력 메모리 할당

② 크기 접두어

데이터를 처리할 때 변수의 크기를 명확히 하기 위해 사용

% I X 0 . 0 . 0

① ② ③ ④ ⑤

접두어	의미
X(or None)	1 비트의 크기
B	1 바이트(8 비트)의 크기
W	1 워드(16 비트)의 크기
D	1 더블 워드(32 비트)의 크기
L	1 롱 워드(64 비트)

변수 표현 - 입출력 메모리 할당

③ & ④ 베이스 번호와 슬롯 번호

- 베이스 번호: [크기접두어]에 따른 N1 번째 데이터
- 슬롯 번호: N1번째 데이터 상의 N2 번째 비트
- 베이스와 슬롯 번호는 0부터 시작

%_1XN.N.0

(1) (2) (3) (4) (5)

베이스와 슬롯

- 베이스(Base)는 랙의 큰 단위이며, 각 베이스에는 여러 개의 슬롯(Slot)이 존재.
- PLC의 CPU의 확장을 위해서 사용하고 하나의 베이스에 슬롯을 추가하는 형태로 사용
- 베이스와 슬롯 번호를 조합하여 모듈의 위치를 고유하게 식별 가능
- XGI 시리즈는 단일 베이스에서 최대 12개의 슬롯을 지원하고, 확장 베이스를 이용하면 베이스도 확장 가능

변수 표현 - 입출력 메모리 할당

- ⑤ N2 슬롯에 대한 N3번째 데이터
(크기 접두어 번호)

%IXN.N.0³
① ② ③ ④ ⑤

%IX0.0.0 ?!

- 입력 접두의 비트 단위의 데이터 &
- 0번베이스, 0번 슬롯의 0번째 데이터라는 의미
- 비트 단위의 데이터의 크기 접두어 x는 생략하는 것과 의미가 같으므로,

%I0.0.0 과 동일

변수 표현 - 내부 메모리 할당

① 위치 접두어

% M B N a N ₂
 ① ② ③ ④

접두어	의미
M	내부 메모리의 M 영역
R	내부 메모리의 R 영역
W	내부 메모리의 W 영역

- 내부 메모리를 할당한다면 사용 영역에 따라 위치 접두어로 M, R, W 가능

변수 표현 - 내부 메모리 할당

② 크기 접두어 (입출력과 동일)

% M
B N 1 N 2
 ① ② ③ ④

접두어	의미
X(or None)	1 비트의 크기
B	1 바이트(8 비트)의 크기
W	1 워드(16 비트)의 크기
D	1 더블 워드(32 비트)의 크기
L	1 롱 워드(64 비트)

변수 표현 - 내부 메모리 할당

③ & ④ 내부 메모리 할당

- 입·출력 메모리의 할당과 기본적인 방법은 동일

but, 베이스 번호와 슬롯 번호를 지정하지 X

③ : N1은 크기 접두어에 대한 번호

④ : 크기 접두어가 X(비트 단위)가 아닐 때, 비트번호

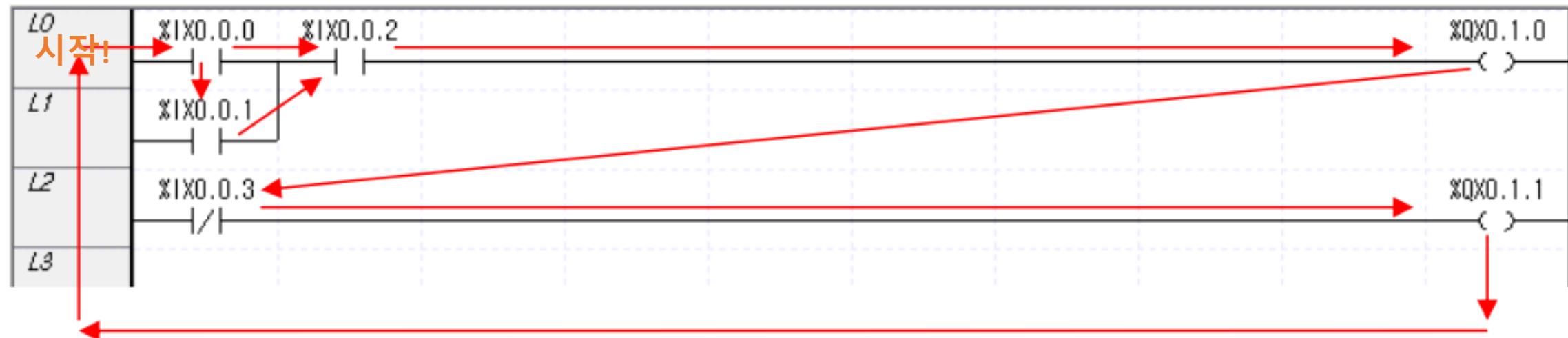
% M B N 1 N 2
 ① ② ③ ④

2-5. 래더 다이어그램

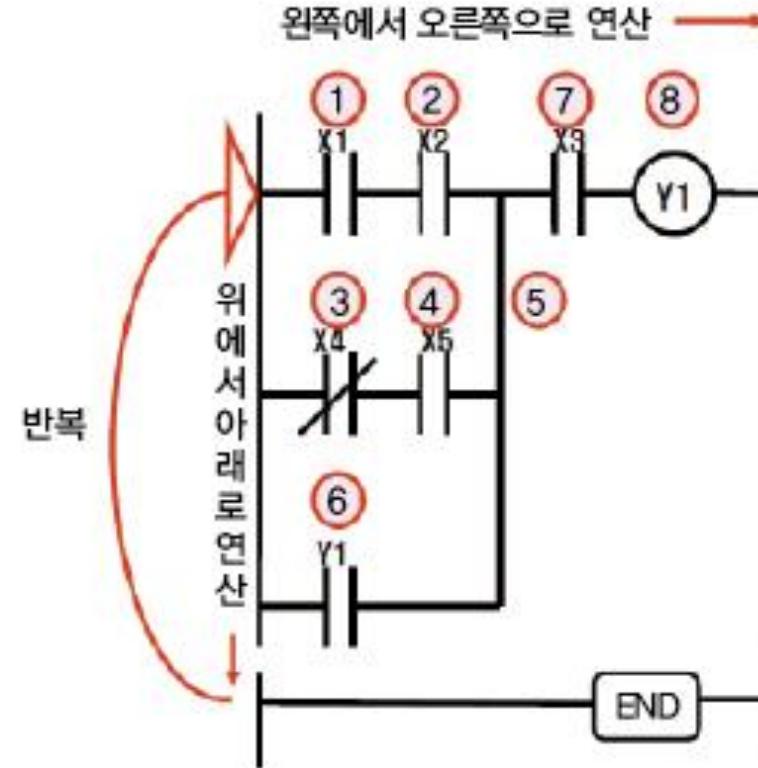


래더 프로그래밍

- 사다리 형태로, 릴레이 로직과 유사한 도형 기반의 언어
- PLC 프로그래밍에 현재 가장 널리 사용되고 있음



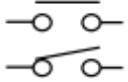
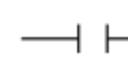
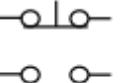
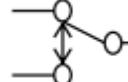
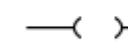
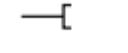
래더도 처리 순서



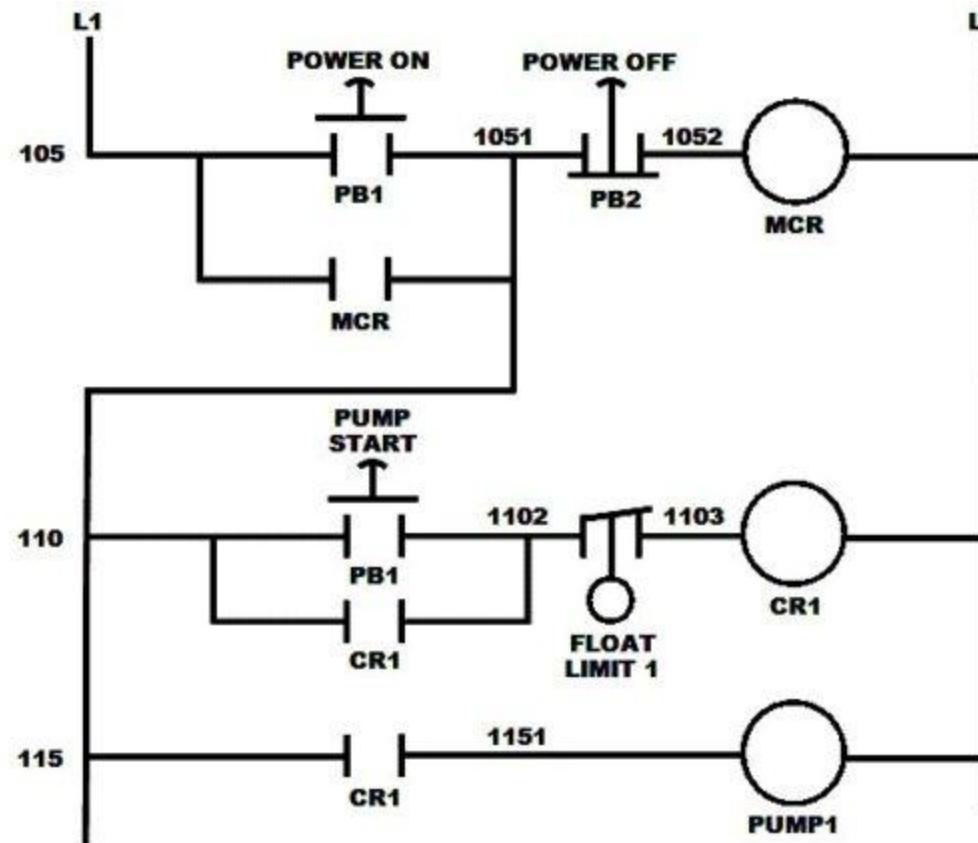
- 래더도 작성의 예와 처리 순서

사용 기호 (명령어)

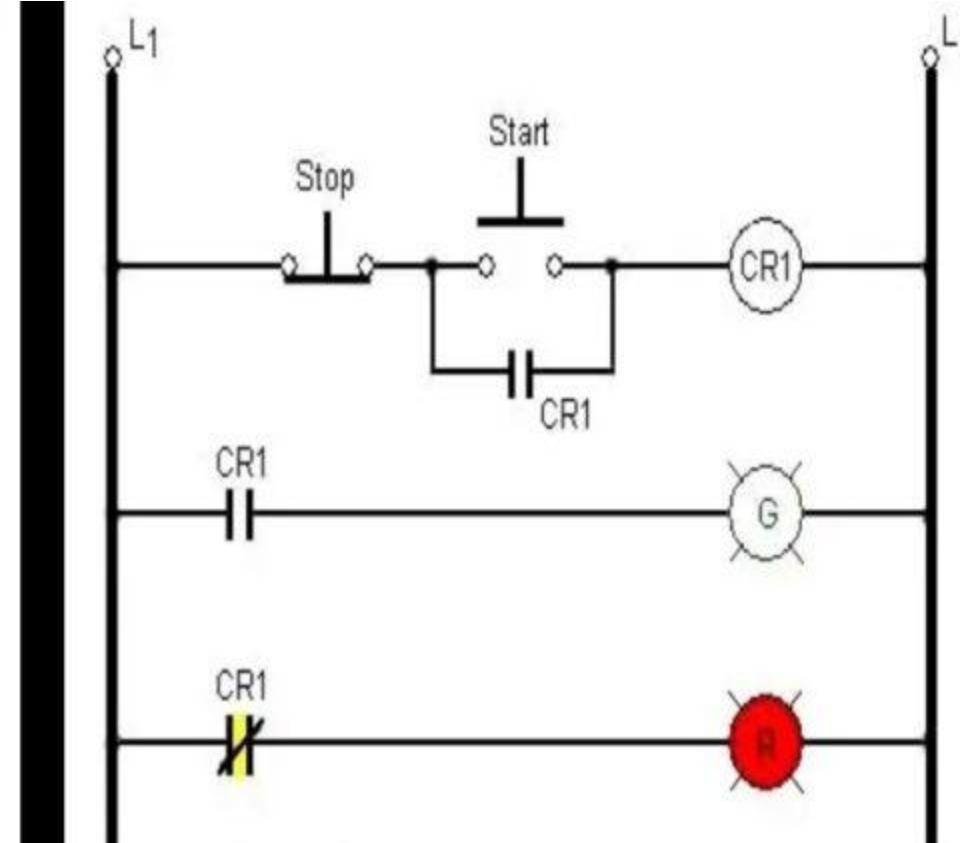
- 스위치 형태의 입력과, 출력 코일
- 릴레이 로직의 기호와 흡사함

구분	릴레이 로직	PLC 로직	내용
A접점			평상시 개방(Open)되어 있는 접점 N.O. (Normally Open) PLC: 외부입력, 내부출력 ON/OFF상태를 입력
B접점			평상시 폐쇄(Closed)되어 있는 접점 N.C. (Normally Closed) PLC: 외부입력, 내부출력 ON/OFF상태의 반전된 상태를 입력
C접점		없음	a, b접점 혼합형으로 PLC에서는 로직의 조합으로 표현
출력 코일			이전까지의 연산 결과 접점 출력
응용 명령	없음		PLC응용 명령을 수행

사용 기호 (명령어)



LADDER LOGIC



RELAY LOGIC

기초 용어

용어	설명
점(Point)	<ul style="list-style-type: none">PLC의 입출력 용량을 나타내는 단위.입력(스위치, 센서 등)과 출력(램프, 모터 등)의 수를 기준으로 표시됨. ➤ 예: 입력 8점, 출력 16점.
스텝(Step)	<ul style="list-style-type: none">PLC 명령어의 최소 단위.프로그램 내에서 실행되는 가장 작은 연산 또는 작업 단위.
스캔 타임(Scan Time)	<ul style="list-style-type: none">사용자가 작성한 프로그램을 1회 수행하는 데 걸리는 시간.PLC의 처리 속도를 나타내는 중요한 성능 지표.
WDT (Watch Dog Timer)	<ul style="list-style-type: none">이상 상황 발생 시 출력 작업이 중단되면 설정된 시간만큼 대기 후 에러를 발생시키는 감시용 타이머.시스템 안전성을 보장함.
파라미터(Parameter)	<ul style="list-style-type: none">프로그램과 함께 PLC에 저장되는 운전 데이터.설정 값, 한계값, 운전 모드 등

감사합니다