

5. 래더(LD) 프로그래밍 기초

5.1 접점(열림, 닫힘), 코일, 출력 구성

(PLC 논리 회로의 기본 구성 요소)

✓ 개요

PLC는 릴레이 기반의 전통 제어 논리를 디지털 방식으로 표현한다.

이 구조의 핵심은 **입력 접점**, **논리 흐름**, **출력 코일**이라는 세 가지 기본 구성 요소다.

✦ PLC의 래더 다이어그램(LD)에서 회로를 구성하는 기본 단위는 아래 세 가지이다:

- 접점 (Contact):** 조건 판단 (입력/내부 상태)
- 코일 (Coil):** 출력 결과 (모터, 램프 등)
- 논리 흐름:** 좌측 전원 레일 → 우측 접지 레일로 흐름이 "통과"하면 코일이 동작

✓ 1. 접점 (Contact)

접점은 PLC에서 **조건을 판단**하는 요소이며, **입력 신호 또는 내부 상태 값**에 기반하여 열림/닫힘 동작을 표현한다.

접점 종류	기호	설명
NO (Normally Open, 통상열림접점)	[]	조건이 참(TRUE)일 때 통전 (기본 상태는 열림)
NC (Normally Closed, 통상닫힘접점)	[/] 또는 [x]	조건이 거짓(FALSE)일 때 통전 (기본 상태는 닫힘)

예시

- [X0] : X0 스위치가 ON일 때만 조건 통과
- [/ X0] : X0 스위치가 OFF일 때만 조건 통과

✓ 2. 코일 (Coil)

코일은 접점의 논리 조건을 만족했을 때 **출력을 설정(ON)**하거나, **내부 플래그를 활성화**하는 역할을 한다.

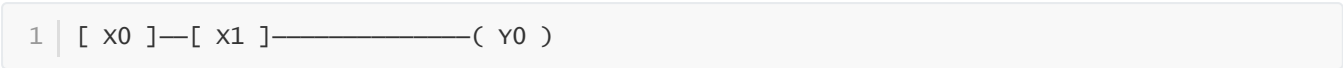
코일 종류	기호	설명
출력 코일	()	조건이 TRUE일 때 출력 ON
Set 코일	S 또는 (S)	조건이 TRUE일 때 출력을 유지(Set)
Reset 코일	R 또는 (R)	조건이 TRUE일 때 출력을 해제(Reset)

예시

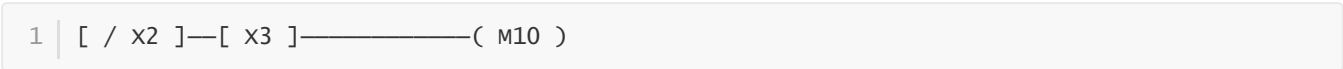
- (Y0) : 조건이 참이면 Y0 출력 켜짐
- (S M100) : M100을 Set (켜짐 유지)
- (R M100) : M100을 Reset (꺼짐)

3. 논리 흐름 구조 (좌 → 우 방향의 논리 흐름)

래더 다이어그램은 전기 회로 흐름처럼 좌에서 우로 조건이 연결되며, 모든 조건이 TRUE일 때 코일이 활성화된다.



위 구조는 “X0이 ON이고, X1도 ON이면 Y0 출력”을 의미

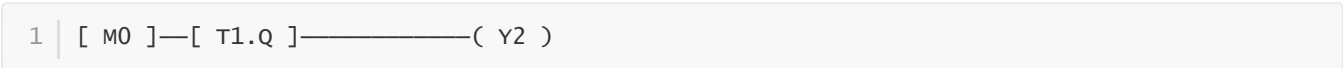


위 구조는 “X2가 OFF이고, X3가 ON이면 내부 메모리 M10 ON”

4. 내부 릴레이 (메모리 접점/코일)

항목	설명
M 영역	내부 릴레이 (사용자 논리 플래그로 활용)
T 영역	타이머 상태 접점 (예: T1.Q)
C 영역	카운터 상태 접점 (예: C1.Q)
D 영역	데이터 레지스터 (수치 저장용, D0, D1 등)

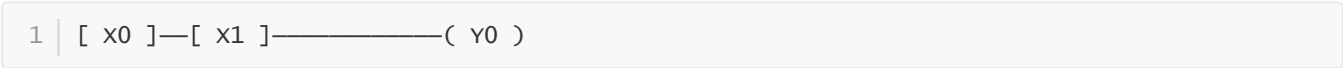
예시:



→ 내부 메모리 M0가 ON이고, 타이머 T1이 완료되면 Y2 출력

5. 래더 구성 예시

🔴 예제 1: 스위치 2개가 모두 눌렸을 때 모터 ON



🔴 예제 2: 스위치가 꺼졌을 때 램프 ON

```
1 | [ / x2 ]—————( Y1 )
```

🔴 예제 3: 스위치 ON 시 램프 Set, 다른 스위치로 Reset

```
1 | [ x3 ]—————(S M10)
2 | [ x4 ]—————(R M10)
3 | [ M10 ]—————( Y2 )
```

✅ 6. 디지털 출력 장치 예

장치	설명
램프 (LAMP)	상태 표시용
모터 (MOTOR)	구동기 작동
솔레노이드 밸브	공압/유압 제어
부저, 경광등	경고 장치
릴레이, SSR	외부 고전압 부하 제어 인터페이스

✅ 정리

- 접점(Contact)은 조건, 코일(Coils)은 결과 출력 또는 상태 저장을 의미한다.
- 래더 다이어그램에서 접점들을 논리적으로 연결하여 전기 회로처럼 표현하고, 조건이 모두 만족될 때 우측 코일이 작동한다.
- 내부 릴레이, 타이머, 카운터, 데이터 메모리 등과 결합하여 복잡한 논리 흐름도 간단한 회로로 표현할 수 있다.

5.2 AND, OR, NOT 회로 논리

(PLC 논리 회로의 기본 연산 구조)

✅ 개요

PLC 프로그래밍의 기본은 논리 연산(Logical Operation)이다.
대표적인 논리 연산에는 AND(논리곱), OR(논리합), NOT(논리부정)이 있으며,
래더 다이어그램, FBD, ST 등 모든 언어에서 입력 조건의 조합을 이 연산들로 표현한다.

🔴 실제 릴레이 제어 회로를 기반으로 하며, 입력이 참(True)일 때만 출력이 동작하는 구조를 갖는다.

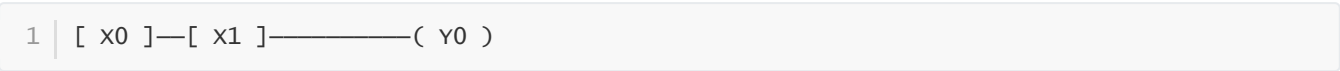
✓ 1. AND 연산 (논리곱)

개념

두 개 이상의 조건이 모두 참일 때만 출력이 참이 되는 논리 연산이다.

입력 A	입력 B	A AND B (출력)
0 (OFF)	0	0
0	1	0
1	0	0
1	1	1 (ON)

래더 예



X0이 ON이고 X1이 ON이면 Y0 출력 ON (AND 조건)

ST 예

1 | Y0 := X0 AND X1;

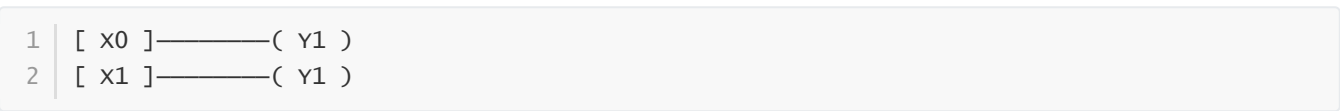
✓ 2. OR 연산 (논리합)

개념

둘 중 하나라도 참이면 출력이 참이 되는 논리 연산이다.

입력 A	입력 B	A OR B (출력)
0	0	0
0	1	1
1	0	1
1	1	1

래더 예



X0 또는 X1이 ON이면 Y1 출력 ON (OR 조건)

ST 예

```
1 | Y1 := X0 OR X1;
```

3. NOT 연산 (논리부정)

개념

입력이 참일 때 **출력이 거짓**, 입력이 거짓일 때 **출력이 참**이 되는 연산.

입력 A	NOT A (출력)
0	1
1	0

래더 예

```
1 | [ / X2 ]—————( Y2 )
```

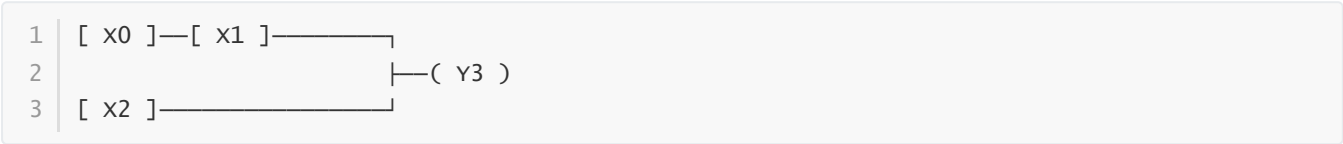
X2가 OFF일 때 Y2 ON (NOT 조건)

ST 예

```
1 | Y2 := NOT X2;
```

4. 복합 논리 조합 예제

예제 1: (X0 AND X1) OR X2



```
1 | Y3 := (X0 AND X1) OR X2;
```

예제 2: NOT (X3 AND X4)

```
1 | [ X3 ]—[ X4 ]——| NOT |——( Y4 )
```

```
1 | Y4 := NOT (X3 AND X4);
```

✓ 5. FBD 논리 구조 시각 예

• AND 블록

```
1 | X0 ┐
2 |   └─ AND ─> Y
3 | X1 ┐
```

• OR 블록

```
1 | X0 ┐
2 |   └─ OR ─> Y
3 | X1 ┐
```

• NOT 블록

```
1 | X2 ─> NOT ─> Y
```

✓ 6. 실제 적용 예

분야	AND 조건 예	OR 조건 예	NOT 조건 예
제조 설비	모든 센서가 OK일 때 작동	비상 스위치 OR 타이머로 정지	오버히트 상태가 아닐 때 가동
엘리베이터	문 닫힘 AND 호출 버튼	위층 호출 OR 아래층 호출	장애 조건이 없을 때 작동
HVAC	온도 AND 습도 조건 만족 시 냉방	사용자 버튼 OR 자동 제어	필터 점검이 아닐 때만 가동

✓ 정리

- AND, OR, NOT은 PLC 회로의 가장 기본적이면서 핵심적인 논리 연산
- 래더에서는 접점의 위치와 병렬/직렬 구조,
ST에서는 논리 수식으로 표현
- 복잡한 조건을 조합하면 다양한 실세계 제어 로직을 구성할 수 있다

5.3 자기유지 회로(셀프홀딩 회로)

✓ 개요

자기유지 회로란, 한 번 켜지면 입력이 꺼져도 상태가 유지되는 회로를 말한다.

PLC에서는 자기유지(Self-holding)를 통해 토글 기능, 시동/정지, 릴레이 유지, 모터 구동 등의 제어를 구현한다.

✦ 흔히 시작 스위치로 작동을 시작하고, 멈춤 스위치로 끄는 구조를 말한다.

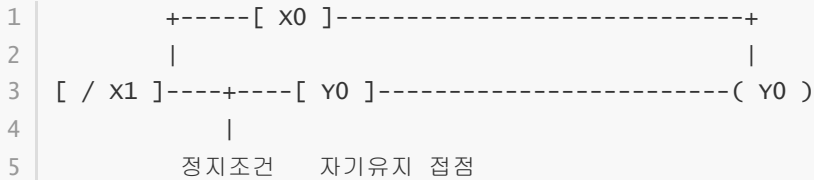
PLC 래더 다이어그램에서는 자기 자신(M코일 등)을 접점으로 재사용하여 유지 상태를 만든다.

✓ 기본 동작 원리

1. 시작 스위치가 눌림 → 출력 ON
2. 출력이 자기 자신을 접점으로 사용 → 상태 유지됨
3. 정지 스위치가 눌리면 → 흐름이 끊어지면서 OFF

✓ 회로도 예제 (래더 다이어그램)

🔴 시동(X0) / 정지(X1) → 모터(Y0)



- [/ X1] : 정지 스위치 (Normally Closed → 평소에는 통전됨)
- [X0] : 시작 스위치 (Momentary ON)
- [Y0] : 자기유지 접점
- (Y0) : 출력 코일 (예: 모터, 릴레이 등)

✓ 동작 설명

1. 평소에는 X1이 닫혀 있어서 흐름이 열려 있음
2. X0을 누르면 → Y0 ON
3. Y0이 켜지면 → [Y0] 접점도 참이 되며 상태 유지됨
4. X0을 떼도 [Y0] 가 계속 참이므로 → Y0 ON 유지
5. X1을 누르면 [/ X1] 이 끊어지므로 → Y0 OFF

✓ Structured Text(ST) 예제

```
1 IF NOT X1 AND (X0 OR Y0) THEN
2     Y0 := TRUE;
3 ELSE
4     Y0 := FALSE;
5 END_IF;
```

- X0 OR Y0 → 입력이든 자기유지든 하나가 ON이면 유지
- NOT X1 → 정지 스위치가 눌리지 않았을 때만 동작

✓ 자기유지의 실용 예

적용 분야	설명
모터 구동	시동 스위치 한 번 누르면 계속 회전, 정지 시 OFF
컨베이어 작동	작동 버튼으로 계속 작동, 정지 버튼으로 멈춤
경광등 유지	일시적 조건 발생 시 알람 유지
에어컨/난방기 제어	버튼으로 ON/OFF 토글 형태 구현 가능
비상 섣다운(인터록)	유지 중에도 비상 정지에 반응 가능

✓ 주의할 점

주의사항	설명
정지 스위치는 반드시 NC 사용	시스템 전원 OFF 시에도 항상 안전하게 OFF 가능해야 함
리셋 조건은 우선순위를 높게	안전 회로나 인터록 조건이 우선적으로 흐름을 끊도록 구성
자기유지 접점이 여러 군데 걸리면 위험	한 회로에 여러 개의 자기유지 점프선이 있을 경우 디버깅 어려움
상태 저장 시 내부 릴레이 사용 권장	출력(Y) 코일을 직접 접점으로 반복 사용 시 물리적 출력 제어 위험

✓ 자기유지 회로와 인터록 회로 차이

항목	자기유지 회로	인터록 회로
목적	상태 유지	상호 배제/보호
트리거 방식	자기 상태를 참조하여 유지	다른 상태와 연동하여 차단
사용 예	시동/정지, 램프 유지	A모터와 B모터 동시작동 방지

✓ 정리

- 자기유지 회로는 일시적 입력을 지속 출력으로 유지하는 대표적 PLC 회로
- 래더에서는 코일 출력(Y 또는 내부 릴레이 M)을 접점으로 재사용하여 상태를 유지한다
- 정지 조건, 인터록, 긴급정지 등과 병용하면 안전하고 안정적인 제어 설계가 가능

5.4 인터록(상호배제) 회로

(상호배제 논리, 동시 작동 방지 회로)

✓ 개요

인터록 회로(Interlock Circuit)란,

서로 동시에 작동하면 안 되는 두 개 이상의 장치나 조건을 상호 배제시키기 위해 설계되는 PLC 논리 회로이다.

✦ 일반적으로 모터 A와 모터 B가 동시에 작동하면 안 되는 경우,
혹은 상하위 동작 순서를 지켜야 하는 경우에 사용된다.

✓ 1. 인터록의 핵심 원리

- 조건 A가 ON이면 조건 B는 강제로 OFF
- 두 조건이 충돌하지 않도록 제어
- 자기유지 회로와 함께 사용하여 작동 상태 유지
- 서로의 작동을 감시하며 “한쪽만 동작”하도록 배제

✓ 2. 인터록 회로 기본 구조 예시 (래더 다이어그램)

✦ 예제: 모터 A(X0로 시동, Y0), 모터 B(X1로 시동, Y1)

→ Y0와 Y1은 동시에 켜지면 안 됨

```
1  [ X0 ]—[ / Y1 ]—[ Y0 ]—————( Y0 )   ← 모터 A 제어
2  [ X1 ]—[ / Y0 ]—[ Y1 ]—————( Y1 )   ← 모터 B 제어
```

✓ 설명

- X0 스위치를 눌렀을 때 → Y0 ON → 자기유지
- 동시에 Y1은 [/ Y0] 조건으로 인해 작동 불가(인터록)
- 반대로 X1 눌렀을 때 → Y1 ON → Y0은 차단됨

✓ 3. Structured Text(ST) 예시

```
1  IF X0 AND NOT Y1 THEN
2      Y0 := TRUE;
3  END_IF;
4
5  IF X1 AND NOT Y0 THEN
6      Y1 := TRUE;
7  END_IF;
```

✓ 4. 응용 예: 3상 모터 정/역운전

조건

- 정회전($X0 \rightarrow Y0$), 역회전($X1 \rightarrow Y1$)
- 동시에 둘 다 켜지면 회로 파손 우려

1	[X0]—[/ Y1]—[Y0]—————(Y0)
2	[X1]—[/ Y0]—[Y1]—————(Y1)

동일한 인터록 구조로 정/역 작동이 배제된다.

✓ 5. 산업 현장 인터록 활용 예

대상	인터록 목적	설명
모터 A / B	동시 작동 금지	부하 과다, 기계 손상 방지
로봇 암 상/하	순차 동작 필요	충돌 방지
실린더 전진/후진	동시 입력 차단	공압 압력 손실 방지
컨베이어 투입/배출	적재/과부하 방지	재료 적재 타이밍 보호
전원선 전환 (UPS ↔ AC)	전원 혼선 방지	비상시 자동 전환

✓ 6. 인터록 회로 설계 팁

팁	설명
출력 Y를 점점으로 차단 조건에 사용	자기유지와 함께 연결하여 상호 배제
우선순위 설계 필요	어떤 동작이 먼저 작동할지 정의해야 함
비상정지(E-Stop)과 병용	안전회로와 통합하여 위험 방지
시퀀스 순서와 함께 사용	SFC 등 순차 제어 논리와 병행 설계

✓ 자기유지 + 인터록 통합 회로 예시

1	[X0]—[/ Y1]—[Y0]—[Y0]——(Y0)
2	[X1]—[/ Y0]—[Y1]—[Y1]——(Y1)

자기유지를 포함한 인터록 회로로, 버튼 한 번으로 유지되며 동시에 작동은 방지됨

✓ 정리

- 인터록 회로는 상호 배제 로직 구현을 위한 필수 회로로, 기계 간 충돌, 오작동, 전기적 손상을 방지하는 역할을 한다.
- 자기유지 회로와 결합하면 강력한 시동/정지/보호 구조를 설계할 수 있다.
- 현장에서는 특히 정·역 회전, 실린더 동작, 안전 순서 제어에서 빈번하게 활용된다.

5.5 램프 제어, 모터 제어 기본

(PLC의 가장 기초적인 출력 제어 예제)

✓ 개요

PLC를 처음 배우는 이들이 가장 먼저 익히는 실습은 램프와 모터 제어이다.

이는 디지털 출력 제어의 기본 개념과, 입력과 출력의 흐름을 이해하는 데 가장 효과적이다.

✦ 이 장에서는 PLC를 통해 ON/OFF 스위치 입력으로 출력 장치인 램프, 모터를 제어하는 방법을 다룬다. 래더 다이어그램(LD), Structured Text(ST) 등의 언어로 예제를 함께 제시한다.

✓ 1. 램프 제어 기본 (ON/OFF)

✦ 목표

스위치(X0)를 누르면 → 램프(Y0) 켜짐

스위치를 떼면 → 램프 꺼짐

✦ 래더 예제

```
1 | [ X0 ] ————— ( Y0 )
```

- X0: 푸시버튼(스위치)
- Y0: 램프

스위치가 눌러 있는 동안만 램프가 켜짐

✦ ST 언어 예제

```
1 | Y0 := X0;
```

✓ 2. 모터 제어 기본 (ON/OFF)

✦ 목표

시작 스위치(X0) 누르면 → 모터(Y1) 켜짐

정지 스위치(X1) 누르면 → 모터(Y1) 꺼짐

여기서는 자기유지 회로를 활용

🔧 래더 예제

```
1  [ / X1 ] —+—[ X0 ]———( Y1 )
2              |
3              +—[ Y1 ]
```

- X0: 시작 스위치 (NO, Momentary)
- X1: 정지 스위치 (NC)
- Y1: 모터 출력

시작 스위치로 ON 후, 정지 스위치로 OFF (자기유지 포함)

🔧 ST 언어 예제

```
1  IF NOT X1 AND (X0 OR Y1) THEN
2      Y1 := TRUE;
3  ELSE
4      Y1 := FALSE;
5  END_IF;
```

✅ 3. 토글 제어 (누를 때마다 ON/OFF)

한 개의 버튼으로 켜고 끄는 구조 (상태 저장 필요)

🔧 ST 예시 (M0: 상태 저장)

```
1  IF X2 AND NOT X2_old THEN
2      M0 := NOT M0;
3  END_IF;
4  X2_old := X2;
5
6  Y2 := M0;
```

- X2: 토글용 버튼
- M0: 내부 상태
- Y2: 램프 또는 모터

✅ 4. 램프/모터 제어 시 주의할 점

항목	설명
출력 전류/전압 확인	릴레이 or SSR 필요 여부 판단
램프는 직접 제어 가능	단, 전류가 크면 릴레이 중계 필요
모터는 릴레이 or 인버터 필요	PLC에서 직접 구동 불가
안전회로 병행 설계 필수	비상정지(E-Stop), 인터록 적용

항목	설명
자기유지/인터록과 병합	현장 안전성과 신뢰도 향상

✓ 실제 적용 예

예	입력	출력
램프 점등	스위치 X0	Y0
모터 가동	시동 X0, 정지 X1	Y1
냉각팬 제어	온도센서 T > 40도	Y2 팬 ON
경고음 발생	리미트센서 X5	Y3 부저

✓ 정리

- 램프/모터 제어는 PLC 실습의 기초 중 기초
- 단순한 ON/OFF 구조부터, 자기유지, 인터록, 토글 등으로 확장 가능
- 래더와 ST 양쪽을 이해하고 병행 실습할 것

5.6 래더 시뮬레이터 사용법

(PLC 래더 프로그램 실습을 위한 소프트웨어 환경 구성 및 사용법)

✓ 개요

래더 시뮬레이터(Ladder Simulator)는 PLC 실제 장비 없이도 래더 다이어그램을 작성하고 실행 결과를 확인할 수 있는 소프트웨어이다.

초보자가 PLC 논리를 이해하고 실습하는 데 매우 유용하며, 실기 시험이나 프로토타입 설계에서도 활용된다.

✦ 대표적인 시뮬레이터:

- LDmicro (무료, 간단한 사용)
- LogixPro PLC Simulator (Allen-Bradley 스타일, 유료)
- CX-Simulator (OMRON)
- GX Simulator (Mitsubishi)
- Siemens TIA Portal + PLCSIM (공식 시뮬레이터)
- PLC Ladder Simulator 2 (모바일용)

✓ 1. 시뮬레이터 구성 요소

구성 요소	설명
래더 에디터	접점, 코일 등을 마우스로 그리거나 코드로 작성
시뮬레이션 제어판	X0, X1 등의 입력 버튼을 ON/OFF
출력 확인 창	Y0, Y1 등의 상태 확인
타이머/카운터 설정	시간 설정, 카운트 조건 테스트
디버깅 기능	스캔 사이클, 흐름, 상태 추적

✓ 2. LDmicro 예시 (PC용 무료 시뮬레이터)

📌 설치 방법

1. <https://cq.cx/dl> → "LDmicro.exe" 다운로드
2. 설치 없이 실행 가능 (무설치 포터블)

📌 사용 방법

- File → New → Choose "Ladder Diagram"
- I/O 설정: X0, X1 → 입력 / Y0, Y1 → 출력
- 마우스로 접점, 코일, 타이머 등을 배치
- Simulate → Start Simulation

예시 회로

1	[X0] ————— (Y0)
---	---------------------

→ X0 ON 시 Y0 출력 확인

✓ 3. LogixPro PLC Simulator (Allen-Bradley 스타일)

항목	설명
시스템 요구	Windows 전용
산업 현장과 유사한 인터페이스	프로세스/컨베이어/엘리베이터 예제 포함
라이트버전 무료 체험 가능	전체 버전은 유료 (\$35~)
래더 → 시뮬레이션 → 동작 결과 확인 가능	실제 시험 대비 학습에 효과적

✓ 4. 모바일 앱: PLC Ladder Simulator 2

항목	설명
플랫폼	Android / iOS
주요 기능	입력 버튼, 래더 작성, 실시간 시뮬레이션
지원 언어	영어 (직관적 GUI)
제한사항	고급 기능(Timer/Counter)은 유료일 수 있음
장점	장소 구매 없이 연습 가능

✓ 5. 실습 예제: 자기유지 회로

```
1 | [ / x1 ]—[ x0 ]—[ y0 ]—————( y0 )
```

- 시뮬레이션 중:
- X0 버튼 클릭 → Y0 ON
 - Y0 자기유지 확인
 - X1 클릭 → Y0 OFF

✓ 6. 시뮬레이터 사용 팁

팁	설명
입출력 상태 초기화 주의	매 회 실행 전 초기화 권장
회로 복잡도는 단계별 실습	접점-코일 → 자기유지 → 인터록 → 타이머 순
스캔 사이클 흐름 보기	실시간 흐름 추적으로 디버깅 감각 기르기
실제 PLC 모델과 호환 확인	나중에 실제 장비 사용 시 인터페이스 차이 최소화
자동 저장 또는 코드 내보내기	회로 백업 필수

✓ 정리

- 래더 시뮬레이터는 PLC 실습의 입문 도구로 매우 효과적
- 초보자는 LDmicro, LogixPro 등 사용하기 쉬운 도구로 시작하고,
- 현업/시험 대비는 **Mitsubishi GX Simulator, Siemens PLCSIM** 등으로 확장
- 반복적인 실습과 디버깅 훈련을 통해 논리 설계 감각을 키울 수 있음