9. PLC 통신 기초 및 실무

9.1 통신 방식 개요 (Polling, Master/Slave, Broadcast)

(PLC와 외부 장치 간 데이터 통신을 위한 기본 구조 및 특징 정리)

✅ 개요

PLC 시스템은 다양한 장비(센서, HMI, 인버터, 로봇 등)와 **통신을 통해 데이터를 송수신**한다.
이때 사용하는 **통신 구조**는 신뢰성, 반응 속도, 장비 간 상호작용 방식에 큰 영향을 미친다.
주요 통신 방식에는 **Polling**, **Master/Slave**, **Broadcast** 방식이 있다.

✅ 1. Polling 방식

> 중앙 제어 장치(PLC 등)가 **일정 주기로 각 Slave에게 순서대로 질의**하는 방식

📌 특징

- **PLC가 적극적으로 데이터를 요청**
- 각 Slave는 PLC가 요청할 때만 응답
- **지속적인 상태 확인에 적합** (예: 온도 센서 값 주기 수집)

📌 구조도

```
[PLC] → [Slave1 요청] → 응답
     → [Slave2 요청] → 응답
     → [Slave3 요청] → 응답 ...
```

📌 장점

- 구현이 간단하고 안정적
- 통신 충돌 없음
- 데이터 동기화 용이

📌 단점

- 장비 수 증가 시 응답 지연
- 이벤트 기반 실시간성은 부족

✅ 2. Master/Slave 방식

> 하나의 **Master(주 장치)**가 여러 **Slave(종속 장치)**를 통제하는 구조

📌 특징

- Master는 **요청 송신**, Slave는 **응답 수신만 수행**
- Slave 간 직접 통신 불가
- 대부분의 산업용 프로토콜(MODBUS RTU, Profibus 등)이 이 구조 기반

📌 구조도

```
[Master PLC] ⇄ [Slave #1]
             ⇄ [Slave #2]
             ⇄ [Slave #3]
```

📌 장점

- 구조가 단순하고 통제가 명확
- 통신 충돌 방지

📌 단점

- Master 고장 시 전체 통신 중단
- Slave 간 협력이 어려움 (오직 Master 통해야 함)

✅ 3. Broadcast 방식

> **하나의 노드가 다수의 장비에게 동시에 데이터 전송**
> (모든 수신자가 같은 메시지를 받음)

📌 특징

- 수신자는 메시지를 듣고 자신에게 해당되는 정보만 처리
- 사용 예: 시간 동기화, 상태 전파, 펌웨어 전송

📌 구조도

```
[PLC] → Broadcast("설비 정지 신호")
       ↳ Slave1 수신
       ↳ Slave2 수신
       ↳ Slave3 수신
```

📌 장점

- 다수 장비를 동시에 제어
- 통신 효율 높음

📌 단점

- 개별 응답 불가능
- 충돌/경합 방지 필요

✅ 비교 요약표

항목	Polling	Master/Slave	Broadcast
통신 방향	Master → Slave 요청	Master 지시, Slave 응답	Master → All
구조	순차적	1:N	1 → N
충돌 가능성	없음	적음	있음 (경쟁 상황 고려 필요)
실시간성	낮음	중간	높음 (전파 빠름)
데이터 흐름	순환	명령-응답	일괄 송신
예시	센서 수집 주기 스캔	MODBUS RTU, Profibus	동기화 브로드캐스트

✅ 실무 적용 예시

분야	방식	설명
MODBUS RTU	Master/Slave	PLC가 인버터/센서에 주기적으로 데이터 요청
EtherNet/IP	Polling 기반 I/O Scan	분산 I/O 상태 수집
시간 동기화 (NTP 등)	Broadcast	여러 장치에 동시 시간 배포
IoT 장비 연결	Hybrid	Polling + Event 기반 병행 사용

✅ 정리

- **Polling**: 순차 질의 방식, 단순하지만 응답 지연 우려
- **Master/Slave**: 명확한 통제 구조, 산업 자동화 표준 구조
- **Broadcast**: 전체 대상 전송에 유리, 응답 불가 구조
- 시스템 요구에 따라 위 방식을 조합하여 설계해야 최적의 통신 구조를 확보할 수 있다.

9.2 시리얼 통신 (RS-232, RS-485)

(PLC 및 산업 제어 시스템에서 널리 사용되는 대표적 시리얼 통신 규격의 구조 및 차이)

✅ 개요

RS-232와 **RS-485**는 가장 오래되고 널리 사용되는 **시리얼 통신 표준**이다.
PLC, 센서, 인버터, HMI 등 많은 장비들이 이 두 방식 중 하나 이상을 지원한다.
저비용, 간단한 배선, **낮은 속도이지만 높은 안정성**이 장점이다.

✅ 1. RS-232: 포인트 투 포인트(Point-to-Point)

📌 구조

- 1:1 통신 (장치 2개 간 직결)
- 일반적으로 **DB9(9핀) 커넥터** 사용
- **Tx (송신), Rx (수신), GND (공통)** 필요

```
[PLC] Tx → Rx [장치]
[PLC] Rx ← Tx [장치]
[PLC] GND ↔ GND [장치]
```

📌 특징

항목	설명
통신 거리	약 15m 이하 (권장 5~10m)
전송 속도	~115.2 kbps
접속 수	1:1만 가능
신호 방식	비차동 (Ground 기준 전압)
전기적 특성	노이즈에 취약, 거리 증가 시 오류↑

📌 장점

- 구성 간단, 저렴함
- 설정이 쉬움

📌 단점

- **노이즈에 약함** (특히 공장 환경)
- 거리 및 속도 제약
- 다자간 통신 불가

✅ 2. RS-485: 멀티 드롭(Multi-Drop) 차동 통신

📌 구조

- **1:N 또는 N:N 통신 가능 (최대 32~128 노드)**
- **차동 신호 (A/B 라인)** 사용 → 전기적 안정성↑
- 일반적으로 **2선식 or 4선식** 사용

```
A ------------- A
B ------------- B
(모든 장비 공유)

예:
[PLC Master]
   ↕
[Slave1]
   ↕
[Slave2]
   ↕
[Slave3]
```

📌 특징

항목	설명
통신 거리	최대 1200m (속도에 따라)
전송 속도	최대 10 Mbps (거리 짧을 경우)
접속 수	다자간 가능 (32~128 노드)
신호 방식	**차동(Differential)**
전기적 특성	노이즈에 강함, 장거리 안정성 ↑

📌 장점

- 다중 장비 제어 가능
- 장거리 통신 가능
- 공장 자동화 환경에 최적
- 노이즈 억제력 강력

📌 단점

- 종단 저항, 라인 길이, 터미네이션 등 설계 필요
- 마스터/슬레이브 ID, 충돌 방지 프로토콜 필요

✅ 3. 주요 비교표

항목	RS-232	RS-485
연결 방식	1:1	1:N or N:N
통신 거리	15m 이하	1200m 이상
노이즈 내성	약함	강함
신호 방식	비차동 (GND 기준)	차동 (A-B 간 전압 차)
배선 수	보통 3선 (Tx, Rx, GND)	2선 또는 4선 (A, B, GND)
통신 속도	~115.2 kbps	최대 10 Mbps
사용 예	PC ↔ PLC, HMI 직결	PLC ↔ 다수 센서/인버터/노드

✅ 4. 실제 사용 예시

장비	통신 방식	설명
구형 인버터	RS-232	PLC에서 직접 제어
MODBUS RTU 네트워크	RS-485	PLC Master ↔ Slave 노드들
단일 HMI ↔ PLC 연결	RS-232	설정 간단한 환경에 사용
환경 센서 네트워크	RS-485	여러 노드를 직렬 연결하여 감지값 수집

✅ 5. 실무 설계 팁

- **RS-485 종단 저항** 필수: 네트워크 양 끝에 120Ω 저항
- **RS-232는 노이즈 필터링 고려**: 접지 잘 할 것
- **케이블 길이와 속도는 반비례 관계**
  - 예: 10 Mbps 시 15m, 9600bps 시 1200m 가능
- **PLC 포트가 DB9이거나 Terminal인 경우가 많으므로 핀맵 주의**

✅ 정리

요약	RS-232	RS-485
구조	단일 연결	다자간 연결
거리	짧음	김
신뢰성	낮음	높음

요약	RS-232	RS-485
용도	단순한 연결	산업용 다중 제어

RS-232는 간단한 **1:1 통신에 적합**,
RS-485는 **장거리/다중 장치**에 최적화된 통신 방식이다.
실제 공정에서는 **RS-485 + MODBUS RTU**가 매우 광범위하게 활용된다.

9.3 Modbus RTU/TCP

(PLC에서 가장 널리 사용되는 산업용 통신 프로토콜: 구조, 차이, 실무 적용)

✅ 개요

Modbus는 1979년 Modicon(현재 Schneider Electric)에서 개발한 **산업용 필드버스 통신 프로토콜**로,
PLC, 인버터, 센서, 계측기 등 **다양한 장비 간 데이터 송수신에 폭넓게 사용**된다.

> 주요 버전:

- **Modbus RTU**: RS-485 기반 시리얼 통신
- **Modbus TCP**: Ethernet 기반 IP 통신

✅ 1. Modbus RTU

📌 개념

- **RS-485 기반의 Master/Slave 구조**
- **이진(Binary) 데이터 전송** (Compact, 빠름)
- 일반적으로 **9600 ~ 115200 bps** 속도 사용

📌 프레임 구조

항목	설명
Slave ID	대상 장비 주소 (1~247)
Function Code	명령 종류 (읽기/쓰기 등)
Data	주소, 값 등
CRC	오류 검사용 (16비트)

예시:

```
[01][03][00 6B][00 03][76 87]
(슬레이브 1번, Holding Register 0x006B부터 3개 읽기)
```

📌 주요 Function Code

코드	명령	설명
01	Read Coil	디지털 출력 읽기
02	Read Discrete Input	디지털 입력 읽기
03	Read Holding Registers	아날로그 출력 값 등
04	Read Input Registers	센서 값 등 아날로그 입력
05	Write Single Coil	1개 디지털 출력 ON/OFF
06	Write Single Register	1개 레지스터 값 쓰기
16	Write Multiple Registers	다수 레지스터 값 쓰기

✅ 2. Modbus TCP

📌 개념

- **Ethernet/TCP 기반 Modbus**
- **IP 네트워크에서 동작** (PLC, SCADA, PC 등과 통신)
- 포트 번호: **502번**

📌 프레임 구조 (헤더 포함)

항목	설명
Transaction ID	요청/응답 구분용
Protocol ID	항상 0
Length	뒤 데이터 길이
Unit ID	RTU의 Slave ID 역할
Function Code + Data	RTU와 동일

예시:

```
[00 01][00 00][00 06][11][03][00 6B][00 03]
→ Transaction 1번, 슬레이브 11번, Holding Register 읽기
```

📌 특징

항목	설명
연결 방식	TCP/IP 기반, 최대 동시 다수 연결

항목	설명
속도	이더넷 (10/100/1000 Mbps)
프로토콜 구조	RTU보다 프레임이 큼
안정성	세션 기반, 전송 안정성 높음

✅ 3. RTU vs TCP 비교

항목	Modbus RTU	Modbus TCP
물리 매체	RS-485 시리얼	Ethernet
구조	Master-Slave (단방향 요청)	Client-Server (양방향)
통신 거리	최대 1200m	네트워크 기반 (LAN/WAN)
속도	최대 115.2kbps	수십 Mbps 이상
장치 수	최대 247개	수천 개 (IP 기반)
실시간성	높음 (직접 연결)	중간 (네트워크 부하 영향 있음)
확장성	제한적	뛰어남 (라우터, 스위치 활용 가능)
오류검사	CRC16	TCP 자체 오류검사 이용

✅ 4. 실무 적용 예시

적용 사례	방식	설명
인버터 제어	RTU	RS-485로 슬레이브 인버터 제어
SCADA ↔ PLC	TCP	Ethernet으로 중앙제어 시스템 연동
온습도 센서	RTU	RTU 방식으로 여러 센서 직렬 연결
PC ↔ PLC	TCP	모니터링/제어 프로그램 구축

✅ 5. 통신 구성 예

▶️ RTU 구성 예

```
[PLC Master]
   ↕
[Sensor Slave1]
   ↕
[Inverter Slave2]
   ↕
```

```
[Meter Slave3]
```

> RS-485 2선식, 슬레이브 ID 부여

▶️ TCP 구성 예

```
[PLC1 - IP:192.168.0.101]
[PLC2 - IP:192.168.0.102]
[PC with SCADA Client]
→ 모두 동일 네트워크에서 IP 기반 연결
```

✅ 6. 설정 주의사항

항목	RTU	TCP
Slave ID 중복 금지	✅	✅ (Unit ID 중복 시 주의)
Baudrate/Parity 통일	✅	❌ (IP만 맞추면 됨)
CRC 오류 검사	✅	❌ (TCP 자체 검증)
고장 대응	Polling Timeout 설정 필요	연결 끊김 감지 필요

✅ 정리

요약	RTU	TCP
장점	구조 단순, 실시간성 ↑	빠른 속도, 확장성 ↑
단점	거리, 노드 수 제한	네트워크 상태에 영향 받음
사용 환경	단일 라인 제어	분산 네트워크 제어

Modbus는 **간단하고 안정적인 표준 프로토콜**로,
소규모 공정부터 대규모 SCADA까지 범용적으로 활용된다.

9.4 Ethernet/IP, Profinet, DeviceNet

(산업용 이더넷 프로토콜 3대 표준 비교와 PLC 실무 활용)

✅ 개요

산업 자동화 환경에서의 이더넷 통신은 단순한 TCP/IP를 넘어
특화된 실시간 제어, 장비 자동 인식, 통합 진단 기능을 요구한다.
이를 만족시키기 위해 사용되는 주요 프로토콜이 바로:

- **EtherNet/IP** (Allen-Bradley / ODVA)
- **Profinet** (Siemens 중심)

- **DeviceNet** (CAN 기반 시리얼 네트워크)

✅ 1. EtherNet/IP (Ethernet Industrial Protocol)

📌 개념

- **표준 이더넷 + CIP(Common Industrial Protocol)** 기반
- TCP/UDP 포트: 44818 (CIP 통신)
- Rockwell Automation (Allen-Bradley) 주도

📌 특징

항목	내용
물리 계층	이더넷 (Cat5/6, RJ-45)
통신 구조	Producer-Consumer 모델
실시간성	우수 (UDP 기반 I/O 메시지)
토폴로지	라인 / 스타 / 링 가능
프로그래밍	태그 기반 주소 지정 (Tag 방식)

📌 실무 포인트

- Studio 5000 환경에서 장비 자동 탐지
- 다수 장비 동시 연결 가능 (스위치 기반 확장)

✅ 2. Profinet (Process Field Net)

📌 개념

- **Siemens 주도**, ISO/IEC 표준의 산업용 이더넷
- **Real-Time (RT)** 및 **Isochronous Real-Time (IRT)** 지원

📌 특징

항목	내용
물리 계층	이더넷
구조	Controller(Device) / Device 구조
실시간성	RT (1~10ms), IRT (≤1ms)
구성 방식	이름 기반 자동 검색 (Device Name)
통합성	PROFIBUS와 호환 가능 (게이트웨이 연결)

📌 실무 포인트

- TIA Portal에서 자동 구성
- 장비의 이름(Device Name) 설정이 매우 중요
- IRT는 전용 하드웨어 필요 (ex: Profinet IRT Switch)

✅ 3. DeviceNet

📌 개념

- **CAN (Controller Area Network)** 기반 산업용 버스
- ODVA 표준, 저속 통신에 적합 (속도 < 500kbps)
- **전원 + 데이터 4선 통합 케이블 사용**

📌 특징

항목	내용
매체	비차동 4선식 (2전원 + 2데이터)
통신 속도	125 ~ 500 kbps
장점	전원공급 및 데이터 통신을 하나의 케이블로
구성	Master/Slave 또는 Peer-to-Peer
제한	최대 거리, 최대 노드 수 한계 있음

📌 실무 포인트

- 인버터, 센서 등 구형 장비에서 여전히 다수 사용
- 전력과 신호가 동일한 배선 → 시공 편리
- RSLogix 5000 등에서 DeviceNet 모듈 필요

✅ 4. 비교 요약표

항목	EtherNet/IP	Profinet	DeviceNet
기반 기술	이더넷 + CIP	이더넷 + RT/IRT	CAN (비차동 시리얼)
최대 속도	100 Mbps 이상	100 Mbps (IRT 지원 시 ≤1ms)	최대 500 kbps
실시간성	중간 (UDP 기반 RT)	높음 (RT/IRT 지원)	낮음
구성	Producer-Consumer	Controller/Device	Master/Slave
배선	표준 LAN 케이블	표준 LAN 케이블	전원 + 통신 4선
대표 PLC	Allen-Bradley	Siemens	구형 Rockwell, Omron

항목	EtherNet/IP	Profinet	DeviceNet
진단 기능	좋음	매우 좋음	제한적
확장성	뛰어남	뛰어남	낮음
설정 방식	Tag 기반	이름 기반	노드 주소 기반

✅ 5. 실무 적용 예시

시스템	프로토콜	설명
Allen-Bradley + 인버터 + I/O	EtherNet/IP	Logix5000에서 자동 태그 연동
Siemens S7 + 센서 네트워크	Profinet	TIA Portal로 구성, IRT 가능
기존 설비 (센서 20개, HMI)	DeviceNet	배선 간단, 속도는 느림

✅ 6. 선택 기준 요약

조건	추천
최신 시스템, IP 기반 제어	EtherNet/IP
초고속 제어, 정확한 동기화	Profinet (IRT)
기존 구형 설비 통합	DeviceNet
저비용, 단순한 설치	DeviceNet
고정밀 위치 제어	Profinet IRT + Motion 장비

✅ 정리

- **EtherNet/IP**: 가장 범용적이고 유연한 IP 기반 통신
- **Profinet**: Siemens 기반 설비에 최적화, 동기화 제어 탁월
- **DeviceNet**: 구형 네트워크, 현재는 점차 EtherCAT, Profinet 등으로 대체 추세

📌 오늘날 공장 자동화에서는 **EtherNet/IP + Profinet**을 중심으로 네트워크 설계를 구성하며,
이더넷 기반 통신은 **빠른 속도 + 통합 유지보수 + 유연한 확장성**을 모두 만족시키는 핵심 수단이다.

9.5 OPC, MQTT 등 산업용 IoT 연동

(PLC 데이터를 클라우드, SCADA, 외부 시스템과 연결하는 IIoT 핵심 기술)

✅ 개요

전통적인 PLC 제어 시스템은 내부 장치 제어에 집중되어 있었지만,
스마트 팩토리·산업용 IoT(IIoT) 시대에는
외부 서버, 클라우드, 데이터 분석 시스템과의 연동이 필수가 되었다.

이를 위해 사용하는 주요 통신 프로토콜이 바로:

- **OPC (OLE for Process Control)**
- **MQTT (Message Queuing Telemetry Transport)**
- 그 외 HTTP/REST, AMQP, CoAP 등도 일부 활용됨

✅ 1. OPC (Open Platform Communications)

📌 개념

- **PLC ↔ SCADA/HMI ↔ IT 시스템** 간의 **표준 인터페이스**
- Microsoft COM/DCOM 기술 기반에서 시작 → 현재는 **OPC UA**가 주류

📌 OPC 버전별 비교

버전	설명	특징
OPC DA	Data Access	COM 기반, Windows 전용
OPC UA	Unified Architecture	플랫폼 독립, 보안 내장, XML/Binary 전송
OPC HDA	Historical Data Access	이력 데이터 처리
OPC A&E	Alarms & Events	이벤트 전용 인터페이스

📌 OPC UA의 특징

항목	설명
구조	Server-Client
프로토콜	TCP, HTTPS, WebSocket
보안	암호화, 인증, 서명 지원
장점	크로스 플랫폼 (Windows/Linux/Cloud), 방화벽 통과 가능
활용	SCADA ↔ PLC ↔ MES/ERP 시스템 연동

✅ 2. MQTT (Message Queuing Telemetry Transport)

📌 개념

- **경량 메시지 기반 Publish / Subscribe 구조**
- 원래는 위성 통신용으로 개발 → **IIoT 표준 프로토콜**로 진화

📌 구조

```
[PLC] -- Publish → MQTT Broker ← Subscribe -- [클라우드 / 앱 / SCADA]
```

- 중간에 **Broker (중앙 서버)**가 존재
- 장치는 특정 주제(Topic)에 대해 **발행(Publish)**하거나 **구독(Subscribe)**

📌 특징

항목	설명
메시지 구조	Topic 기반 (예: factory/line1/temp)
연결 방식	TCP/IP (1883 포트)
전송 형식	텍스트(JSON), 바이너리 모두 가능
QoS	0 (최소), 1 (확인), 2 (중복 방지)
장점	경량, 저대역폭, 모바일/클라우드 친화적
활용	PLC → 클라우드 서버 전송, 모바일 앱 상태 구독 등

✅ 3. OPC vs MQTT 비교

항목	OPC UA	MQTT
구조	Server-Client	Publish-Subscribe
통신 대상	SCADA, MES, 클라이언트 앱	클라우드, 브로커 기반 시스템
데이터 형식	구조화된 태그, XML/Binary	JSON, 바이너리, 경량
보안	내장 (TLS, 인증)	TLS 가능, 외부 보안 설정 필요
실시간성	비교적 우수	낮음 ~ 중간 (Broker 의존)
활용 분야	산업 현장 내 제어 통합	클라우드, IoT 디바이스 연동
설치 용이성	복잡 (서버 구성 필요)	가벼움, 오픈소스 다양 (Mosquitto 등)

✅ 4. 실무 적용 예시

시스템	연동 방식	설명
PLC + SCADA	OPC UA	WinCC ↔ S7 PLC 데이터 연동
PLC → 클라우드	MQTT	MQTT 브로커 통해 AWS IoT Core 전송
PLC + 모바일 알림	MQTT + Subscribe	알람 발생 시 스마트폰 알림
공장 전체 통합	OPC UA + MQTT	현장 → SCADA(MES) → Cloud 분석

✅ 5. PLC 연동 방법

✅ (1) MQTT 연동

- MQTT Client 라이브러리 내장 (예: Codesys, WAGO, Siemens IoT2040 등)
- 또는 외부 게이트웨이 사용 (e.g., Kepware, Ignition Edge)

✅ (2) OPC UA 연동

- Siemens S7-1500, Beckhoff, Omron NX 등 OPC UA Server 내장
- SCADA/Client에서 OPC UA Client 구성
- 보안 정책, 인증서 설정 주의

✅ 6. 실무 설계 시 고려 사항

항목	권장 사항
보안	TLS/SSL 적용, 인증서 기반 통신
연결 상태 감지	Heartbeat 설정 (MQTT Keepalive, OPC Ping 등)
데이터 필터링	중요 데이터만 송신하여 네트워크 부하 감소
브로커 이중화	MQTT 브로커 장애 대비 Backup 구성
시간 동기화	IoT 장비 간 타임스탬프 일치 필요 (NTP 등)

✅ 정리

요약	OPC UA	MQTT
PLC ↔ SCADA/IT 통합	매우 강력	간접 가능
클라우드 연동, 모바일 연동	어려움	매우 강력
정형화된 데이터 모델	제공	없음 (Topic 직접 정의)

요약	OPC UA	MQTT
실시간 제어	적합	비실시간 통신에 적합

📌 **현장 제어** → **OPC UA**, **클라우드 및 모바일** → **MQTT**로 이원화 설계하는 것이 가장 일반적이다.

9.6 다른 PLC 간 통신 (PLC-to-PLC)

(이기종 또는 동일 제조사 PLC 여러 대를 동기화하고 연계 제어하는 통신 구조)

✅ 개요

현장의 복잡한 공정에서는 여러 대의 PLC가 서로 **협조 동작, 데이터 공유, 상호 제어**해야 할 필요가 많다.
이를 위해 구현하는 것이 **PLC-to-PLC 통신 (Inter-PLC Communication)** 이다.

PLC-to-PLC 통신은 **확장성, 유연성, 분산제어, 통합진단** 측면에서 매우 중요한 기술이다.

✅ 1. 적용 사례

분야	적용 예시
대형 생산 라인	공정별 분산 PLC 제어기 협조
자동 물류 시스템	컨베이어 ↔ AGV ↔ 로봇간 제어 연계
원격 장비 통합	공장 ↔ 외부 저장조, 유틸리티 제어실 연동
이중화 시스템	메인 PLC ↔ 스탠바이 PLC 상태 동기화

✅ 2. 기본 통신 구조 패턴

구조	설명	특징
마스터-슬레이브	한쪽이 지휘, 한쪽은 응답	단순, 일방향
Peer-to-Peer	양쪽이 동등하게 송수신	실시간 협조
브로드캐스트	상태를 전체 PLC에 일괄 전파	이벤트 중심 통제

✅ 3. 주로 사용하는 통신 프로토콜

프로토콜	통신 방식	제조사
MODBUS RTU	RS-485 시리얼 기반	범용
MODBUS TCP	Ethernet 기반	범용
EtherNet/IP	Producer-Consumer	Allen-Bradley

프로토콜	통신 방식	제조사
Profinet IO Controller/Device	이더넷 기반	Siemens
FINS (Omron 전용)	이더넷/시리얼	Omron
MC Protocol	시리얼/이더넷	Mitsubishi
OPC UA Pub/Sub	고급 분산 통신	범용 차세대 IIoT 환경

✅ 4. 주요 통신 모델 비교

모델	특징	설명
데이터 맵 공유 (Global DB Mapping)	가장 단순	PLC 간 공유 변수 주소 매핑
메시지 기반 전송 (Explicit Messaging)	유연	특정 조건 발생 시만 송신
이벤트 기반 퍼블리시/서브스크라이브	고급	MQTT, OPC UA Pub/Sub 사용

✅ 5. 실무 설계 예시

▶️ 예 1: Siemens S7 → S7 통신 (S7 Connection)

- S7-1500 ↔ S7-300 간 직접 통신 가능
- Global DB 영역을 공유하거나 Put/Get 명령 사용

```
// S7-1500이 S7-300의 DB100.DBD0 값을 읽음
S7Client(Peer IP := '192.168.0.10',
         DB := 100,
         Offset := 0,
         Length := 4);
```

▶️ 예 2: Allen-Bradley Logix → Logix (Produced/Consumed Tags)

- Ethernet/IP 기반 태그 실시간 공유

생산 PLC	소비 PLC
`MotorSpeed` 태그 생성 (Produced Tag)	동일 태그 참조 (Consumed Tag)

→ 태그 생성 후 실시간 동기화됨 (이벤트 기반 데이터 교환)

▶️ 예 3: 이기종 PLC 간 MODBUS TCP 통신

- 모든 PLC가 지원 가능 (범용성 최고)

```
PLC1: MODBUS TCP Master (Client)
PLC2: MODBUS TCP Slave (Server)
```

- Holding Register 기반으로 데이터 송수신 (Function Code 03, 06 등 사용)

✅ 6. 통신 동기화 설계 고려사항

항목	고려 사항
데이터 일관성	전송 주기, 타임스탬프 일치
우선순위	중요한 변수는 빠르게 전송
실패 대응	통신 오류 발생 시 재시도 논리
충돌 방지	Write 충돌 방지 논리 필요 (누가 Master인가?)
데이터 속도	통신 과부하 방지 위한 적절한 주기 설정

✅ 7. 실시간 협조 제어 구조

(1) 빠른 응답이 필요한 경우 → Peer-to-Peer 고속 링크

- Profinet IRT, EtherNet/IP CIP Sync, 시리얼 고속 링크 등

(2) 공정 간 단순 상태 공유 → Polling 기반 교환

- MODBUS TCP 읽기/쓰기 주기적 수행

(3) 이벤트 중심 알람 통신 → MQTT, OPC UA Pub/Sub 연동

- PLC → 클라우드 상태 전송 후 전 공정 공유

✅ 8. 시각화 예시

```
+---------+       +---------+       +---------+
| PLC #1  | <-->  | PLC #2  | <-->  | PLC #3  |
| 공정 A  |       | 공정 B  |       | 공정 C  |
+---------+       +---------+       +---------+

- 전역 데이터 테이블: 생산량, 공정상태, 에러상태 실시간 공유
- 전원 ON 후 동기화: 각 PLC가 자기 상태 Self-Report
- 통신 중단 시 대기 모드
```

✅ 정리

- **PLC-to-PLC 통신은 분산 제어의 핵심**
- 동일 제조사 간 전용 프로토콜 활용 시 가장 쉽고 안정적
- 이기종 간은 MODBUS, OPC UA가 가장 범용적
- 실시간성 요구에 따라 Peer-to-Peer, Event 기반 통신 병행 설계

9.7 PLC와 HMI/SCADA 간 통신

(현장 자동화의 핵심: 운영자 인터페이스 시스템과 PLC 간 실시간 데이터 교환 구조)

✅ 개요

PLC는 제어 중심,
HMI/SCADA는 모니터링, 데이터 표시, 알람 관리, 오퍼레이터 제어 중심으로 역할이 구분된다.

이 둘은 항상 **통신을 통해 밀접히 연결**되어 운영되며,
전체 자동화 시스템의 실시간 감시 및 제어를 담당한다.

✅ 1. HMI와 SCADA의 차이

구분	HMI (Human Machine Interface)	SCADA (Supervisory Control and Data Acquisition)
위치	현장 로컬	중앙 통합 감시
규모	개별 장비 감시	다수 장비 통합 관리
기능	버튼, 수치 표시, 알람	이력 저장, 보고서, 원격 제어
통신 대상	주로 1개 PLC	다수 PLC, 외부 시스템

✅ 2. 통신 구조 일반도

```
[ PLC 1 ]  [ PLC 2 ]  [ PLC 3 ]
     ↕         ↕         ↕
  HMI Panel  SCADA Server
```

- **HMI ↔ PLC**: 실시간 I/O 값 표시, 버튼 입력
- **SCADA ↔ PLC**: 전체 생산 데이터 통합, 이력 관리, 알람 저장

✅ 3. PLC ↔ HMI/SCADA 간 통신 프로토콜

프로토콜	설명	제조사
OPC UA	표준 프로토콜 (산업 표준 통합용)	범용
MODBUS RTU	시리얼 기반	모든 PLC
MODBUS TCP	이더넷 기반	모든 PLC
Profinet	이더넷 실시간 통신	Siemens
EtherNet/IP	태그 기반 이더넷 통신	Rockwell
MC Protocol	시리얼/이더넷	Mitsubishi
FINS	Omron 전용	Omron

> 대부분의 HMI/SCADA는 다양한 드라이버 내장 → 서로 다른 PLC도 통합 가능

✅ 4. 통신 방식 종류

통신 방식	설명	예시
Polling 방식	주기적 반복 읽기	매 500ms마다 PLC 상태 읽기
Event 기반	조건 발생 시 즉시 전송	알람 발생 시 HMI로 즉시 전송
Subscription (OPC UA)	값 변화 시 자동 업데이트	태그 값 변경 시 실시간 갱신

✅ 5. 태그 기반 통신 설계

(1) 태그 예시

PLC 내부 태그	HMI 연동 태그
`MotorRun`	화면 표시등 점등
`Temperature`	온도 수치 표시
`Alarm_OverHeat`	경고 화면 활성화

(2) HMI 설계 시 태그 맵핑

```
PLC 태그 → HMI 태그 드라이버 → 화면 오브젝트 연계
```

- 대부분의 HMI/SCADA 소프트웨어는 태그 자동 스캔 기능 지원

✅ 6. 통신 속도와 주기 설계 기준

변수 종류	갱신 주기
긴급 알람	즉시 (100~500ms)
주요 동작 상태	500ms~1s
일반 생산 데이터	1~5초
이력 기록용 데이터	10초 이상

✅ 7. 실전 HMI/SCADA 소프트웨어 예시

소프트웨어	제조사	특징
WinCC	Siemens	Profinet 통합 최적
FactoryTalk View	Rockwell	EtherNet/IP 태그 연동
iFIX	GE	OPC UA 통합
Citect SCADA	Schneider	다수 PLC 통합
Ignition	Inductive Automation	MQTT, OPC UA, IIoT 특화
Proface	Schneider	독립형 HMI 패널

✅ 8. 통신 장애 대비 설계

상황	설계 포인트
통신 중단	Watchdog 타이머, 통신 상태 태그 구성
HMI 재접속	자동 재연결 및 상태 복구 기능 활용
통신 병목	태그 그룹핑, 불필요 데이터 송신 최소화

✅ 9. HMI/SCADA 시스템 보안 고려

위험	대응
비인가 접근	사용자 로그인, 권한 설정
네트워크 침입	VPN, 방화벽, 암호화
데이터 위·변조	OPC UA 보안정책, 인증서 기반 통신

✅ 10. 확장적 구성 예시 (실제 공장 적용 예)

```
[ PLC 1 - 생산라인 A ]
[ PLC 2 - 생산라인 B ]
[ PLC 3 - 포장공정 ]
[ PLC 4 - 창고 물류 ]
       ↓
[ SCADA 서버 - 공장 중앙제어실 ]
[ 클라우드 서버 - 이력/품질 분석 ]

- 실시간 제어: SCADA ↔ PLC
- IIoT 확장: OPC UA ↔ 클라우드 빅데이터 분석
- 모바일 모니터링: MQTT ↔ 현장 관리자 앱
```

✅ 정리

- **PLC ↔ HMI/SCADA 통신은 전체 자동화 시스템의 뼈대**
- 실시간 데이터 반영, 알람 전송, 히스토리 관리가 핵심
- **OPC UA는 표준 통합에 강력**,
- **MODBUS TCP는 범용성 탁월**,
- **제조사 전용 프로토콜은 통합성 뛰어남**