12. 디버깅 및 유지보수

12.1 온라인 모니터링

(PLC 제어 시스템의 실시간 감시와 상태 확인을 위한 온라인 모니터링 설계 및 활용 기법)

✅ 개요

온라인 모니터링은 PLC 프로그램이 실행되는 상태를 실시간으로 감시하며:

- 프로그램의 현재 동작 상태 확인
- 변수 값, I/O 신호 확인
- 논리 흐름 실시간 추적
- 디버깅, 시험운전, 유지보수 지원

을 가능하게 만드는 **가장 기본적이면서도 핵심적인 유지보수 기능**이다.

✅ 1. 온라인 모니터링의 기본 정의

개념	설명
온라인 접속	현재 실행 중인 PLC 프로그램과 엔지니어링 소프트웨어가 실시간 연결
상태 모니터링	프로그램 논리 흐름, 각 변수 값, 타이머/카운터 상태 실시간 표시
실시간 디버깅	조건 만족 여부, 입력 신호 활성화 시점 확인 가능

✅ 2. 온라인 모니터링의 활용 목적

활용	설명
프로그램 디버깅	래더 프로그램의 흐름 실시간 확인
시운전 지원	시험운전 중 상태 확인 및 확인 작업
유지보수	고장 발생시 원인 신속 확인
교육 훈련	프로그램 동작 이해, 단계별 시뮬레이션 확인
이상 분석	논리 충돌, 조건 미달 확인

✅ 3. 모니터링 가능한 주요 요소

대상	확인 내용
입력 신호 (I/O)	센서, 버튼, 스위치 실시간 상태
출력 신호 (I/O)	릴레이, 모터, 밸브 출력 확인
내부 변수	플래그, 상태변수, 연산결과 등
타이머	누적시간, 타이머 완료 조건
카운터	누적 카운트값, 완료상태
PID 제어	SP, PV, MV 실시간 피드백

✅ 4. 실전 모니터링 화면 구성 예시

```
┌──────────────────────────────┐
| 래더 프로그램 화면             |
| [ ] [X001]   →──( ) [Y001]   |
|            ↑   ↑      ↑      |
|           상태 표시 활성화     |
└──────────────────────────────┘

- 활성 접점 실시간 점등
- 활성 코일 ON 실시간 점등
- 변수값 우측에 실시간 표시
```

✅ 5. 주요 제조사별 온라인 모니터링 기능 예

PLC 제조사	엔지니어링 소프트웨어	모니터링 특징
Siemens	TIA Portal	래더 흐름 점등, 변수 그래프 지원
Rockwell	RSLogix / Studio 5000	라인 흐름 점등, 태그 실시간 업데이트
Mitsubishi	GX Works	디바이스 레벨 실시간 I/O 확인
Omron	Sysmac Studio	구조화 태그 실시간 모니터
LS	XG5000 / XG500	I/O 상태, 변수값 실시간 트렌드 제공

✅ 6. 온라인 모니터링 단계별 활용 흐름

```
1️⃣ PLC 프로그램 다운로드 완료
        ↓
2️⃣ 온라인 접속 활성화
        ↓
3️⃣ 래더 프로그램 흐름 실시간 점등
        ↓
4️⃣ 센서 동작 → 조건 충족 여부 확인
        ↓
5️⃣ 출력 결과 및 변수 변화 실시간 확인
        ↓
6️⃣ 이상 발생시 원인 위치 즉시 파악
```

✅ 7. 실시간 모니터링 예시

▶️ 예: 컨베이어 감지 오류 추적

```
- 포토센서 감지 불량 발생
- 온라인 모니터링으로 센서 입력 X003이 미활성 확인
- 센서 감지거리 조정 → 즉시 X003 활성 점등 확인
```

- 물리적 문제 즉시 확인 가능 → 불필요한 프로그램 수정 없이 현장 조치 가능

✅ 8. 안전한 온라인 모니터링 설계 주의사항

주의 사항	설명
온라인 모드에서도 **쓰기 금지 (Read-Only)** 상태 유지 권장	
강제 값 설정 (Force Set) 기능은 유지보수 전문가만 사용	
운영 중 모니터링 시 프로세스 간섭 최소화	
이중 네트워크 구성으로 정상 제어와 엔지니어링 회선을 분리	

✅ 9. 고급 온라인 모니터링 기능

고급 기능	설명
트렌드 그래프	변수 실시간 그래프 확인 (PID 튜닝 활용)
상태 히스토리	조건 변화 시 시간순 기록
이력 재생	이벤트 재현으로 분석 가능
조건 모니터링 필터	특정 조건 충족시 자동 캡처

✅ 10. 실전 현장 적용 예시

분야	활용 예
생산 시운전	설비 조립 직후 초기 디버깅
유지보수	장애 발생시 센서·출력 실시간 상태 확인
교체부품 시험	센서 교환 후 즉시 정상동작 여부 확인
공정 개선	신규 시퀀스 수정 후 실시간 검증

✅ 정리

- **온라인 모니터링은 PLC 유지보수·디버깅·교육·장애진단의 핵심 도구**
- 실시간 상태 파악 → 빠른 고장 원인 판별 → 불필요한 프로그램 수정 최소화
- 제조사별 기능은 다르지만 모든 PLC 시스템에서 **가장 실용적이고 꼭 익혀야 할 핵심 기술**이다

12.2 강제 설정(Force), 모드 변경

(PLC 유지보수 및 디버깅에서 사용하는 강제 제어 기능과 운전모드 전환 구조 설계)

✅ 개요

강제 설정 (Force) 이란:
PLC의 **입출력 신호나 내부 변수 값을 프로그램 논리와 무관하게 임의로 고정시키는 디버깅 기능**이다.
이 기능은 프로그램 수정 없이 즉시 시스템 상태를 테스트하거나
장애 원인을 특정하기 위해 사용된다.

✅ 1. 강제 설정(Force)의 기본 원리

개념	설명
입력 강제 (Force Input)	센서가 미감지 상태여도 ON 상태로 강제
출력 강제 (Force Output)	논리가 출력 OFF라도 강제로 ON 유지
내부 변수 강제	프로그램 조건과 무관하게 내부 변수 값 고정

- **Force 활성 중에는 프로그램 논리보다 강제값이 우선 적용**

✅ 2. Force 기능의 활용 목적

목적	설명
센서 배선 점검	감지 불량 센서를 강제 ON 시켜 시스템 테스트
출력 부하 시험	릴레이, 솔레노이드 작동 여부 점검
시운전 단계별 단계 검증	복잡한 시퀀스 중간단계 강제 진입
장애 위치 추적	의심 신호 강제 설정 후 반응 확인
부품 교환 검증	교체한 부품의 정상작동 즉시 테스트 가능

✅ 3. 강제 설정(Force) 기본 사용 흐름

```
① 온라인 모드 접속
② 대상 신호 선택 (I/O, 변수)
③ Force Set → 강제값 입력 (ON/OFF 또는 수치)
④ 시스템 반응 확인
⑤ 정상 확인 후 반드시 Force Release (해제)
```

- **Force 해제 전 시스템 정상 복귀 안됨 주의**

✅ 4. 래더 논리 내에서 Force 동작 위치

```
[물리입력] → [Force 우선 적용] → [프로그램 래더 논리] → [출력 Force 가능]
```

- 래더 논리는 항상 Force 적용 결과를 최종 신호로 사용

✅ 5. 제조사별 Force 기능 특징

제조사	Force 기능	주의사항
Siemens TIA	"Force Table" 존재	프로그램 실행 중 가능
Rockwell	"Force Enable/Disable" 개별 설정	안전 프로토콜 철저
Mitsubishi GX	I/O 및 내부 변수 강제 가능	Force Mode 진입 명확 표시
Omron Sysmac	구조화 태그 강제 지원	통신 차단 중 불가

✅ 6. 강제 설정 사용시 주의사항

위험요소	방지대책
예상치 못한 기계동작	반드시 시스템 위험구간 정지 후 사용
안전장치 무력화 가능성	Force 중 비상정지 시스템 별도 유지
해제누락	강제 해제 잊지 않도록 운영절차 강화
Force 사용이력 미기록	모든 Force 적용 이력 자동 기록 권장

✅ 7. 운전 모드 변경 개념

PLC 시스템은 일반적으로 여러 "운전 모드"를 지원하며,
강제 설정과 병행하여 디버깅, 유지보수, 시험운전에서 사용된다.

운전 모드	설명
자동 모드 (Auto)	프로그램 시퀀스 자동 실행
수동 모드 (Manual)	오퍼레이터 직접 제어 가능 (버튼/스위치)
반자동 (Semi-Auto)	자동 일부 단계만 실행
테스트 모드 (Test)	유지보수 중 개별 장치 확인용
시뮬레이션 모드	PLC 프로그램 실행 테스트 (입출력 가상화)

✅ 8. 운전 모드 전환 구현 예시

▶️ 모드 전환 스위치 활용

```
| Manual_Switch |──[ ]──( Mode := MANUAL )
| Auto_Switch |──[ ]──( Mode := AUTO )
```

- 현재 모드에 따라 시퀀스 흐름 분기

▶️ 시퀀스 흐름 분기 예

```
| Mode = AUTO AND Start_Pressed |──[ ]──( 시퀀스 시작 )
| Mode = MANUAL |──[ ]──( 수동 개별 버튼으로 각 기기 직접 제어 )
```

✅ 9. 강제 설정 + 운전모드 연계 활용 예시

시나리오	강제/모드 활용
센서 배선 불량 추적	입력 Force ON 후 상태 변화 확인
모터 출력 확인	수동 모드 전환 후 개별 출력 시험
생산 라인 시운전	단계별 강제 전진 → 타이밍 확인
유지보수 시험	전체 자동 시퀀스 중단 후 개별 부하만 강제 시험

✅ 10. 실전 유지보수 적용 사례

분야	예시
컨베이어 라인	모터 구간별 독립 강제 구동 확인
자동문	센서 끼임감지 센서 강제 시험
온도제어	PID 입력 변수 강제 조작 후 튜닝
로봇 제어	안전구간만 한정 동작 테스트

✅ 정리

- **강제 설정 (Force)** 은 디버깅, 유지보수, 시운전 필수 도구
- 항상 **안전관리 우선 → 시스템 위험 최소화 조치 병행 필수**
- **운전 모드 전환**을 병행하면 현장 유지보수 효율이 비약적으로 향상
- **강제 사용 이력 관리 → 모든 Force 작업은 기록으로 남기는 습관이 표준**

12.3 이력 보기, 에러 로그 확인

(PLC 시스템 유지보수에서 핵심적인 이력 데이터 조회와 고장 원인 추적 설계)

✅ 개요

PLC 시스템이 장시간 안정적으로 운영되려면 단순히 **현재 상태만 모니터링**하는 것이 아니라
이전까지 발생한 모든 이력 기록을 축적하고 분석하는 시스템이 반드시 필요하다.

특히:

- 장애 발생 이력 (에러 로그)
- 동작 이력 (운전 상태 변천 기록)
- 조작 이력 (작업자 조작 기록)
- 품질 이력 (공정 변수 이력)

을 체계적으로 관리해야 **고장 예방, 유지보수 최적화, 품질 안정화**가 가능해진다.

✅ 1. 이력 데이터 관리 시스템의 구성 요소

항목	설명
에러 이력	장애 발생 시간, 에러코드, 원인
경보 이력 (Alarm History)	실시간 알람 발생/복귀 기록
운전 상태 이력	정지-운전-대기 상태 전환 시간 기록
작업자 이력	조작자 로그인, 수동조작 기록
품질 변수 이력	온도, 압력, 유량 등 장기 데이터 기록

✅ 2. PLC 내부 에러 로그 시스템

📌 기본 구조

- 대부분 고급 PLC는 **내부 이력 버퍼(Log Buffer)** 를 내장
- 일정 수량의 최근 에러코드, 발생시간, 해제 여부 저장

📌 PLC 프로그램 내 이력 기록 구현 예시

```
| 에러 발생 |──[↑]──( Error_Log_Buffer[N] := 현재 Error_Code )
| 시간 기록 |──( Error_Time[N] := 현재 시각 )
```

- 메모리 부족 방지 위해 순환버퍼 구조 사용 (FIFO)

✅ 3. SCADA 시스템 이력 보기 화면 구성 예시

시간	장비	에러코드	원인	확인여부
2025-06-17 08:42	Line-2	0201	모터 과부하	확인완료
2025-06-17 08:45	Line-2	0501	비상정지	확인중

- 이중 필터, 검색, 기간별 조회 기능 제공
- 클릭 시 상세 이력 팝업 → 작업자 조치내용 기록

✅ 4. 에러 이력 기록 예시 흐름

```
① 에러코드 발생
② PLC 로그버퍼 기록
③ SCADA 시스템 DB 기록 (이중 기록)
④ 운영자 알람 발생 확인
⑤ 조치결과 입력 → 로그 해제 완료 기록
```

✅ 5. 이력 데이터 수집 구조

방식	설명
PLC 내부 버퍼	짧은 기간 이력 저장 (RAM 한정)
SCADA Historian	장기 이력 (수개월~수년 보관 가능)
MES 통합 DB	전체 공장 이력 통합 분석

✅ 6. 이력 데이터 조회 예시 화면

▶️ 장애 이력 화면

```
[ 장비번호 ] [ 에러코드 ] [ 발생시각 ] [ 해제시각 ] [ 원인메시지 ]
```

▶️ 운전상태 이력 화면

```
[ 시각 ] [ RUN ] [ STOP ] [ ALARM ] 상태 변화 시간차 기록
```

▶️ 품질이력 (변수트렌드)

- 시간축 그래프 기반 실시간 및 과거 데이터 검색 가능
- 이상발생 직전의 온도, 압력, 유량 등 변화 확인 가능

✅ 7. 에러 로그의 활용 예시

활용 목적	분석 예시
반복장애 분석	센서 0103번 3개월간 반복불량 발생 확인 → 설비 교체결정
공정개선	과부하 발생 패턴 확인 → 속도감속 설계 변경
품질이상 연동	품질불량과 온도변동 상관관계 확인
예방정비 계획	이력기반 정기 점검 일정 최적화 가능

✅ 8. 이력관리 시스템 보안 및 품질 확보 설계

항목	관리 기준
이력 이중화	PLC ↔ SCADA ↔ MES 3중 기록 유지
이력 수정 금지	이력 데이터는 원본불변 원칙 적용
시간 동기화	모든 장비 시각 NTP 동기화 유지
이력 검색속도	장기이력도 빠른 검색 가능하도록 인덱싱 구축

✅ 9. 실전 현장 적용 예시

분야	적용 사례
전자	불량률과 장비이력 자동 연결 분석
자동차	품질클레임 발생시 과거 이력 재현 분석
식음료	충진불량 발생시 탱크 유량이력 추적
물류	배출불량 발생시 센서오류 이력 조회

✅ 10. 이력 보기 시스템의 확장

확장방향	설명
클라우드 이력	장기데이터 원격 통합 저장소
빅데이터 분석	장기 품질 데이터 머신러닝 분석 적용
예방정비 시스템	패턴분석 기반 Predictive Maintenance 구축
법적 품질추적	식품, 제약, 반도체 공정 Traceability 구현 필수

✅ 정리

- **이력 보기 시스템은 단순 기록 기능이 아니라 유지보수의 핵심 의사결정 근거 시스템**
- 정확한 장애원인 기록 → 재발 방지 → 품질 안정성 확보
- 초기 설계시 이력 체계 설계를 완벽히 구축하면
  장비 운영 수명이 갈수록 안정화, 데이터 중심 유지보수 실현 가능

12.4 펌웨어 업데이트 및 백업/복원

(PLC 시스템의 장기 안정성, 유지보수, 확장성을 위한 핵심적인 펌웨어 관리와 데이터 보존 설계)

✅ 개요

PLC 시스템은 장시간 안정적으로 운영되기 위해:

- **펌웨어 업데이트 (Firmware Update)**
- **백업/복원 시스템 (Backup & Restore)**

을 반드시 체계적으로 설계해야 한다.

이 두 영역은 단순히 데이터 문제가 아니라,
생산 중단 위험 방지, 장애 복구, 시스템 이중화 유지에 필수적인 안정성 확보 기술이다.

✅ 1. 펌웨어(Firmware) 개념

항목	설명
정의	PLC 하드웨어 내장 소프트웨어 (실행 커널)
포함 요소	실시간 OS, 통신 드라이버, 하드웨어 인터페이스
업데이트 목적	버그 수정, 신규 기능 추가, 보안 취약점 패치

- 펌웨어는 프로그램 논리와 별도로 동작하는 **컨트롤러의 핵심 운영층**이다.

✅ 2. 펌웨어 업데이트의 필요성

시나리오	설명
기능 확장	신형 통신 프로토콜 지원 추가
안정성 향상	간헐적 통신오류 버그 수정
보안 강화	산업용 네트워크 취약점 패치
신형 주변기기 지원	신규 I/O 모듈 호환성 확보

✅ 3. 펌웨어 업데이트 일반 절차

```
① 제조사 펌웨어 다운로드
② PLC 장비 백업 완료 확인
③ 시스템 안전 정지 (비상모드)
④ 엔지니어링 툴 통해 업데이트
⑤ 버전 확인 → 시스템 재구동
⑥ 기능 시험 → 정상 운영 재개
```

- **운영 중 온라인 업데이트 불가** (대부분 PLC는 정지상태 필요)

✅ 4. 펌웨어 업데이트 안전 설계

요소	방지책
전원 차단	UPS 이중화 적용
중간실패 대비	이중 Bootloader 설계 활용 (Rollback 가능)
버전 기록	전체 장비별 펌웨어 이력 문서화
승인 절차	업데이트 승인 체계 마련 (운영 승인 후 시행)

✅ 5. PLC 시스템 백업/복원의 기본 개념

항목	설명
프로그램 백업	래더/블록/태그/주석 포함 전체 프로그램
하드웨어 설정 백업	모듈 구성, 네트워크 파라미터, 슬롯 매핑
데이터 백업	레시피 데이터, 파라미터 설정값, 태그 초기값
시스템 복원	장애시 원상태 빠른 복구 가능

✅ 6. 백업 구성요소 분류

구성 요소	포함 내용
소스 프로그램 백업	PLC 논리 전체 (래더, STL, FBD, ST, 파라미터)
하드웨어 구성	슬롯맵, 통신설정, IP주소
운영 데이터	레시피 테이블, 수치 설정값
라이선스 파일	일부 고급 PLC는 라이선스 활성키 필요

✅ 7. 백업 전략 예시

백업 유형	주기	특징
정기 백업	월 1회	신규 프로그램 변경 반영
변경시 백업	프로그램 수정 직후	최대한 자주 수행 권장
이중 매체 백업	USB + 외장 NAS	물리적 이중 보관
클라우드 백업	원격 백업 서버	오프사이트 안전성 확보

✅ 8. 복원 시스템 설계

📌 완전 복원 흐름

```
① 신규 PLC 하드웨어 교체
② 백업 파일 복구 (하드웨어 설정 + 프로그램 로드)
③ 운영 데이터 복구 (레시피, 파라미터)
④ 통신 연동 테스트
⑤ 시퀀스 정상 확인 → 운영 복귀
```

- 하드웨어 사양 일치 필수 (슬롯 구성/통신옵션)

✅ 9. 제조사별 백업 복원 툴 예시

제조사	엔지니어링 툴	백업 기능 특징
Siemens	TIA Portal	전체 프로젝트 통합 백업 가능
Rockwell	Studio 5000	아카이브 패키지 자동 생성
Mitsubishi	GX Works	프로젝트 + 파라미터 동시 백업
Omron	Sysmac Studio	컨트롤러 설정 포함 패키지
LS	XG5000	전체 시스템 단일 패키지 생성 지원

✅ 10. 현장 유지보수 실전 예시

상황	활용
부품 교체	PLC CPU 고장시 즉시 교체 → 복원 수행
프로그램 수정	실험적 논리수정 전 백업 확보
장기운영 안정성	수년 운영 중 설정오류 누적 방지
다수라인 표준화	동일 설비 다수 라인 프로그램 배포

✅ 11. 고급 확장 설계 요소

요소	설명
자동 이중 백업 시스템	매일 자동 스냅샷 백업 구축
오프라인 백업 검증	복원 시뮬레이션 테스트 주기적 시행
버전 관리 시스템	Git 기반 PLC 프로젝트 관리 시도
변경이력 추적	PLC 변경 기록 로그파일 관리

✅ 정리

- **펌웨어 업데이트 → 시스템 기반 안정성 확보**
- **백업/복원 체계 → 장기 운영 안정성의 핵심 보험 시스템**
- 백업 설계가 철저하면 **장비 교체, 장애 복구, 다수라인 관리**까지 모두 신속 가능
- 초기 프로젝트 설계시부터 **"백업은 무조건 습관화"** 설계가 되어야 한다

12.5 통신 이상, 센서 오류 대응

(PLC 제어 시스템의 안정성과 복구력을 좌우하는 통신장애·센서이상 대책 설계)

✅ 개요

PLC 시스템은 본질적으로 **센서·네트워크·제어장비 간 실시간 통신**에 의존한다.
따라서 **통신 오류**나 **센서 장애**가 발생하면:

- 잘못된 동작
- 공정 정지
- 생산 손실
- 안전 사고

로 즉시 연결된다.

통신 이상과 센서 오류를 어떻게 설계단계부터 관리하는가가 시스템 품질을 결정한다.

✅ 1. 통신 이상 대응 (PLC ↔ I/O, PLC ↔ PLC, PLC ↔ SCADA)

📌 주요 통신 이상 유형

문제 유형	원인 예시
링크 단절	케이블 단선, 커넥터 불량
통신 타임아웃	상대장비 응답 지연
프로토콜 오류	패킷 손실, CRC 불일치
통신 혼선	네트워크 과부하, EMI 간섭
장비 재부팅	전원 불안정, 펌웨어 오류

📌 통신 이상 감지 논리 설계

```
| 통신 Watchdog Timer 초과 |——[ ]——( 통신 이상 플래그 ON )
```

- PLC가 주기적 통신 상태 점검 (통신 하트비트 감시)

📌 PLC 내 통신 오류 전용 태그 예시

변수명	설명
Comm_Status_OK	정상 여부
Comm_Error_Count	누적 오류 카운터
Last_Comm_Time	마지막 정상 통신 시각

📌 통신 이상 시 응급대응 설계

설계 항목	대응 방법
비상정지	주요 장비 즉시 정지 (Fail-Safe 우선 적용)
모드 전환	수동운전 모드 자동 전환
중립위치 이동	로봇, 이송장치 등 안전위치 자동 이동
오류 알람	SCADA/운영자 즉시 경고 표시
이력 기록	발생시간, 장비번호, 재발 빈도 기록

✅ 2. 센서 오류 대응 설계

📌 센서 오류 주요 유형

유형	원인
오프라인	케이블 단선, 전원 불량
오작동	노이즈, 감지 범위 벗어남
변동불안정	조도변화, 온도 영향, 센서 노후화
장기 Drift	오차누적 (특히 아날로그 센서)

📌 센서 진단 로직 예시

① 아날로그 입력 검사

```
| 센서값 < 최소허용 |——[ ]——( 센서 이상 플래그 )
| 센서값 > 최대허용 |——[ ]——( 센서 이상 플래그 )
```

② 비활성 타이머 기반 진단

```
| 동일 상태 지속시간 초과 |——[ ]——( 센서 고착 추정 )
```

③ 다중 센서 비교

- 동일 계측 대상에 2중 센서 설치 → 상호 비교
- 이중화 설계 (Redundancy)

📌 센서 이상시 대응 동작 설계

상황	대응
경미 이상	소프트 경고 → 작업자 점검 요구
치명 이상	즉시 인터록 → 장비 보호정지
이중 센서 이상	시스템 안전대피 모드 전환
반복불량	예방정비 대상 우선순위 등록

✅ 3. 센서 노이즈 대응 (하드웨어/소프트웨어)

유형	대응
접점 노이즈	하드웨어 디바운스 회로 (RC필터, 쉬미트트리거)
소프트 디바운싱	소프트웨어 지연 필터 적용
아날로그 변동	이동평균 필터 (Moving Average)
EMI 간섭	실드케이블, 접지 설계 강화

✅ 4. 고급 복구 설계 기법

📌 통신 이상시 단독운전 (Local Autonomy)

- PLC 간 통신 단절 → 자기 보존형 운전 유지 가능 설계
- 예: 컨베이어 라인은 로컬 영역 단독운전 유지

📌 소프트 세이프 모드 (Soft Safe)

- 통신·센서 오류 발생 → 미리 정의된 제한적 운전만 허용
- 최소한의 공정 유지 후 정지 (Soft Shutdown)

✅ 5. 실시간 SCADA 알람 연동 구조

항목	경고내용	대응
통신 이상	"PLC ↔ SCADA 통신 끊김"	즉시 관리자 호출
센서 고착	"Level Sensor No Response 10s"	현장 점검 필요
반복이상	"Flowmeter Drift Detected"	교체 대상 등록

✅ 6. 실전 현장 사례

분야	사례
반도체	탱크 레벨센서 Drift 이력 → 조기교체
물류	컨베이어 통신 단절시 라인간 차단구간 확보
식음료	온도 PID 센서 이상시 히터 자동 차단
전자	로봇 센서 고착시 중립대피 → 사고방지

✅ 7. SCADA 시스템 연계

- 모든 통신 및 센서 장애 발생시
  SCADA 이력기록 → 중앙 장애분석 → 예방정비 연계

📌 예: 반복 통신장애 통계 분석 → 특정 스위치 교체 계획

✅ 8. 예방정비 최종 설계 목표

지향 목표	시스템 결과
장애 조기 탐지	장애발생률 획기적 감소
유지보수 효율화	불필요한 정비 횟수 감소
시스템 가용률 향상	MTBF 증가, 생산성 향상
안전사고 방지	비상정지 최소화

✅ 정리

- **통신장애와 센서이상 대응설계는 PLC 안정성 설계의 절대 핵심**
- 감지-경고-보호-대피-기록-분석의 **다단계 장애관리 체계 구축**
- 이 체계가 완성되어야 진정한 **안정적 24시간 연속운전 시스템** 구축이 가능하다.

12.6 생산 중단 방지를 위한 Fail-safe 설계

(PLC 시스템에서 고장 발생 시 생산중단을 최소화하기 위한 고급 안정성 설계 전략)

✅ 개요

일반적인 PLC 시스템은 에러 발생 시 **정지(Stop)**로 대응하지만,
고급 산업 현장에서는 단순정지가 불가능하거나 큰 손실을 야기한다.
따라서 반드시:

- **고장 발생 → 정지 대신 제한적 안전운전 유지**
- **최소 기능 유지 → 단계적 복구 가능성 확보**

를 목표로 하는 **Fail-safe 설계**가 요구된다.

Fail-safe는 단순히 "정지"가 아니라 **"안전하게 계속 버틸 수 있는 상태로 자동전환"** 이다.

✅ 1. Fail-safe 기본 개념

개념	설명
Fail-stop	에러 발생시 즉시 전체 시스템 정지 (기본 PLC 구조)
Fail-safe	에러 발생 시 제한적 안전운전 유지, 위험요소 회피
Fail-operational	장애 중에도 전체 시스템 계속 운영 가능 (고급 이중화 시스템)

> 대부분 제조자동화는 **Fail-safe** 단계를 목표로 설계한다.

✅ 2. Fail-safe 설계 기본 원칙

원칙	설명
위험 제거 우선	인명·장비 손상 가능성 제거
최소 기능 유지	가능한 공정 부분은 계속 운영 유지
자동 복구 대기	장애 해소되면 스스로 복귀 가능
단계적 비상 모드 전환	중립대기 → 부분운전 → 복구 가능 흐름 유지

✅ 3. Fail-safe 상태전이 일반 흐름

```
정상운전
  ↓ (에러 발생)
Fail-safe 제한운전
  ↓ (에러 복구 불가 시)
완전정지
  ↓ (복구 후 재가동)
정상운전 복귀
```

✅ 4. Fail-safe 대상별 설계 예시

📌 센서 장애시

상황	대응
비중요 센서 오류	디폴트값 유지 → 제한 운전 유지
중복센서 보유시	정상 센서로 자동 전환

📌 통신 장애시

상황	대응
일부 I/O 단절	남은 영역만 제한운전 (Local Autonomous Mode)
중앙 SCADA 단절	현장단위 자율운전 유지

📌 모터·인버터 이상시

상황	대응
모터 구간 일부 불능	병렬라인 자동전환, 우회로 확보
인버터 통신 끊김	고정 속도 수동 백업모드 구동 유지

✅ 5. 실전 Fail-safe 제어 패턴 예시

▶️ 자동문 예시

```
집센서 이상 발생 → 문 완전 개방 고정 → 사람 안전 확보
문이 완전히 닫히지 않는 위험 방지
```

▶️ 컨베이어 라인 예시

```
센서 감지불량 → 다음 구간 제품 진입 자동 차단 → 충돌 방지
남은 제품은 수동 분리 가능
```

▶️ 혼합탱크 예시

```
온도센서 이상 → 히터 OFF → 교반 유지 → 품질 불량 방지
전면정지 대신 위험요소만 차단 유지
```

✅ 6. PLC 프로그램 내 Fail-safe 논리 삽입 예

▶️ 센서 이상시 안전기본값 대체

```
| 센서 이상 |──[ ]──( 사용값 := 안전 디폴트 )
| 센서 정상 |──[ ]──( 사용값 := 실시간 센서값 )
```

▶️ 제한운전 플래그 활성화

```
| Fail-safe 조건 발생 |──[ ]──( 제한운전 모드 ON )
```

- 이때 전체 프로그램 시퀀스가 제한운전 플래그를 고려하여 동작 변경

✅ 7. SCADA 연계 Fail-safe 상태 표시

상태	알림 표시
제한운전 활성	"Fail-safe Mode Active" 표시
센서 이상 디폴트전환	"Sensor Backup Mode" 경고점등
안전수위 유지	"Partial Production Active"

✅ 8. 실전 Fail-safe 적용 사례

산업 분야	적용 사례
반도체	초순수 공급 차단시 잔여공정 유지 후 자동종료
자동차 조립	일부 용접로봇 불능시 우회라인 자동전환
식음료	충진기 오류시 공정라인 분기 후 잔여제품 보호
물류	분류기 분기 오류시 메인 라인 적재대기 전환

✅ 9. Fail-safe 설계 핵심 지표

항목	목표
사고율	0% (인명안전 확보)
시스템 가동률	최대 유지
복구 소요시간	최소화
무중단 운전 시간	최대 확보 (MTBF 극대화)

✅ 10. Fail-safe 설계의 실전 적용 전략

설계 단계	내용
① 고장모드 정의	장비별 예상 가능한 장애 패턴 도출
② 우선순위 설정	생명/설비 → 공정품질 → 생산성 순
③ 단계적 제한운전 설계	수동 개입 여유시간 확보 설계
④ 이중화 적용 병행	하드웨어 이중화 설계 연계
⑤ 운영자 인터페이스 명확화	SCADA 경고 체계 일관성 유지

✅ 정리

- **Fail-safe는 PLC 제어의 '고장나도 안전하게 유지되는' 핵심 품질 설계**
- 초기 설계단계에서 장애를 가정하고 대응 흐름을 체계화해야 실전에서 강한 시스템이 만들어진다
- Fail-safe → 안정성 확보 → 유지보수 효율 → 생산성 극대화 → 전체 스마트팩토리의 근간이 된다.