

기술

AI가 PLC에 보낸 명령을 막는 장치는, 빠진 인터록을 왜 못 알아채는가

방동걸 (주)웨이스 대표이사

2026년 9월 21일

AI나 MES가 PLC에 직접 명령을 쓰는 공장에서는, 틀린 명령이 설비에 닿기 전에 한 번 막아 주는 장치가 필요합니다. 그 장치는 그 공장 PLC 프로그램을 그대로 옮겨 둔 검증용 프로그램에서 명령을 먼저 실행해 보고, 지켜야 할 조건이 깨지면 명령을 보내지 않습니다. 그런데 현장의 프로그램은 바뀝니다. 보전 작업 중에 임의로 빼 둔 인터록이 그대로 남기도 하고, 운전 중 편집으로 한 줄이 달라지기도 합니다. 그러면 검증용 프로그램은 낡은 프로그램으로 판정을 계속합니다.

제조 AI 운영체제 VEXPLOR를 만드는 웨이스가 여기서 자연스럽게 나오는 생각 하나를 실제로 실험해 봤습니다. 「검증용 프로그램이 아직 현장과 같은지를 동작으로 확인하고, 확인이 안 되면 판정 불가로 막자」는 것입니다. 답은 「잡는 것이 늘지 않는다」였습니다. 검사를 붙인 쪽과 뺀 쪽이 똑같은 23건을 놓쳤고, 그 가운데 사람이 다칠 수 있는 4건은 검사의 원리상 보일 수 없는 것이었습니다. 이 글은 그 4건이 왜 안 보이는지, 그리고 무엇을 먼저 두어야 보이는지를 적습니다.

정정 안내 (2026-09-22). 이 글의 바탕인 기술보고서 TR-2026-01 v1.0 에 사실 오류가 있어 웨이스가 v1.1 로 바로잡았습니다. 이 글이 「인터록이 제거된 뒤에 들어온 모터 가동 명령 4건」이라고 쓴 것은 실제로는 **PLC 가 정지해 있는 동안 들어온 가동 명령**이었습니다(4건 가운데 현장 프로그램에서 인터록이 빠진 것은 1건). 가동 중인 PLC 에서 인터록이 빠진 상태의 위험 명령은 시험 세트에 12건 있었고 모두 막혔습니다. 또 「조건식을 타이머 값으로 고쳐 쓰면 나머지 4건이 0이 된다」는 예상도 다시 재 보니 틀렸습니다. 그 4건은 현장 프로그램에서만 타이머 설정값이 바뀐 경우라 동작으로는 보이지 않고 프로그램 서명 대조로만 잡힙니다 — 동작으로 확인하는 검사가 보지 못하는 것은 인터록 제거가 아니라 이런 「아직 동작에 나타나지 않은 설정 변경」입니다. PLC 정지 중에 들어온 명령은 이후 넣은 스캔 카운터 확인이 24건 모두 막습니다. 결론인 「동작 일치 검사를 붙여도 놓치는 위험 명령은 줄지 않는다」는 다시 재도 같았습니다(두 구성 모두 278건 가운데 19건). 「검증용 프로그램은 최소 1초 전에 동기화된 상태」라는 설명도 틀렸습니다 — 검증용 프로그램은 판정할 때마다 동기화되고, 1초는 동작 일치 검사가 값을 확인하는 간격입니다. 조건식을 세 종류로 나눈 설명은 이후 설계에서 「과거 구간을 보는 연산자(once-held)가 동기화 시점 이전까지 거슬러 올라가는 조건식을 불러올 때 거부한다」로 바뀌었습니다. 정정 전문: [TR-2026-01 v1.1](#) · [정정 목록](#)

01

명령을 막는 장치는, 검증용 프로그램이 현장과 같다는 전제 위에 있습니다

이 글에서 다루는 장치를 짧게 「게이트」라고 부르겠습니다. 게이트는 명령을 내는 쪽(AI 계획기·MES·HMI)과 PLC 사이에 있습니다. PLC의 상태(입출력·내부 비트·타이머 경과값)를 검증용 프로그램에 맞춰 두고, 들어온 명령을 검증용 프로그램에서 몇 주기 앞서 실행해 본 뒤, 조건이 깨지면 그 명령을 PLC로 보내지 않습니다. 이 방식 자체는 새것이 아니고 공개된 선행 기술이 있습니다.

전제는 하나입니다. **검증용 프로그램이 현장 프로그램과 같아야 합니다.** 같지 않으면 게이트는 현장에 없는 인터록을 근거로 「안전하다」고 판정하게 됩니다. 문제는 그 전제가 깨지는 일이 드물지 않다는 것입니다.

- **보전 중 임시 우회** — 점검하려고 인터록 한 줄을 빼 두었다가 되돌리지 않은 채 가동합니다.
- **운전 중 편집** — 라인을 세우지 않고 프로그램을 고칩니다. 검증용 프로그램은 그 편집을 모릅니다.
- **의도된 변경** — 누군가 일부러 프로그램을 바꿉니다. 이것은 게이트가 막는 일이 아니라, PLC 쓰기 보호와 변경 관리가 막아야 할 일입니다.

3가지 경우 모두 결과는 같습니다. 검증용 프로그램과 현장이 어긋난 상태에서 명령이 들어옵니다.

02

그래서 검증용 프로그램이 현장과 같은지를 동작으로 확인하는 검사를 붙이게 됩니다

어긋남을 잡는 가장 그럴듯한 방법은 관찰입니다. 현장에서 태그가 어떻게 바뀌는지를 계속 보고, 검증용 프로그램을 같은 입력으로 실행했을 때 같은 결과가 나오는지 대조합니다. 어긋남이 보이면 「경계」 상태로 들어가고, 경계가 풀릴 때까지 명령을 판정 불가로 막습니다. 이 글에서는 이것을 **동작 일치 검사**(behavioral-equivalence check)라고 부르겠습니다.

말로 들으면 빈틈이 없어 보입니다. 프로그램이 다르면 동작이 다르고, 동작이 다르면 검사가 잡는다는 논리입니다. 실험은 그 논리의 앞 절반이 항상 성립하지는 않는다는 것을 보여 주었습니다.

03

검사를 붙인 쪽과 뺀 쪽은, 똑같은 23건을 놓쳤습니다

웨이스가 공개한 기술보고서 TR-2026-01의 실험은 게이트를 5가지 구성으로 나누어 같은 명령 869건을 넣은 것입니다. 구성은 하나씩 부품을 더하는 식이라, 넷째와 다섯째 구성의 차이가 곧 동작 일치 검사 하나의 효과입니다. 위험 명령 284건 가운데 판정 대상이 아닌 4건을 뺀 280건이 기준입니다.

구성	놓친 위험 명령	잘못 막은 명령	판정 불가(어긋남)
미리 실행 + 내부 상태 동기화	23/280 (8.2%)	6/309 (1.9%)	5.7%
위 구성 + 동작 일치 검사	23/280 (8.2%)	6/309 (1.9%)	7.0%

놓친 명령의 수는 같고, 잘못 막은 명령의 수도 같습니다. 달라진 것은 판정 불가가 5.7%에서 7.0%로 늘어난 것뿐입니다. 현장 입력이 0.07~0.3초만 유지되는 조건에서는 PLC를 세우지 않고 가동 중에 올린 25건 가운데 16건이 「검증용 프로그램이 최신인지 미확인」으로 막혔습니다(채택한 1.2초 유지 조건에서는 0건). 검사는 잡는 것을 늘리지 못했고 막는 것만 늘렸습니다.

놓친 23건을 위험 등급으로 나누면 사람 노출(사람이 다칠 수 있는 명령) 8건, 제품 6건, 기록 5건, 지표 4건입니다. 이 글의 주제는 사람 노출 8건 가운데 4건입니다. 그 4건은 인터록이 제거된 뒤에 들어온 모터 가동 명령이었습니다.



인터록이 빠진 뒤 가동 명령이 들어오기까지의 순서 — 태그는 움직이지 않고, 검사는 경계 상태로 들어가지 않으며, 가동 명령은 통과합니다

04

인터록이 빠진 프로그램은, 가동 명령이 올 때까지 아무 태그도 움직이지 않습니다

동작 일치 검사는 관찰된 변화를 근거로 어긋남을 찾습니다. 그런데 인터록 한 줄이 빠진 프로그램은 그 줄이 실행될 때까지 아무것도 바꾸지 않습니다. 인터록은 가동 명령이 들어올 때 비로소 평가되는 조건이기 때문입니다. 가동 명령이 오기 전까지 태그는 하나도 움직이지 않고, 움직임이 없으니 검사는 경계 상태로 들어갈 이유가 없습니다.

그리고 가동 명령이 들어오는 순간, 게이트가 읽는 허용 조건은 현장의 값이 아니라 검증용 프로그램의 값입니다. 검증용 프로그램은 최소 1초 전에 동기화된 상태이고, 그 안의 인터록은 「정상」입니다. 게이트는 정상이라고 판정하고 명령을 보냅니다. 검증용 프로그램에서 허용 조건을 읽는 게이트는, 낡은 버퍼에서 허용 조건을 읽는 PLC와 같습니다.

이것은 특정 구현의 결함이 아닙니다. 동작을 관찰해 최신인지 확인하는 검사 전체가 갖는 경계입니다. 실행된 적 없는 줄의 변경은 동작에 나타나지 않고, 동작에 나타나지 않는 것은 관찰로 잡을 수 없습니다. 검사를 더 정교하게 만들어도 이 4건은 같은 지점에서 다시 통과합니다.

05

조건식에 순간의 이벤트를 쓰면, 검사가 아무리 좋아도 판정할 수 없는 조건이 생깁니다

사람 노출 8건의 나머지 4건은 원인이 다릅니다. 경보음 지속 시간이나 로봇 팔의 대기 시간이 바뀐 상태에서 통과한 명령들인데, 이 경우는 조건식의 형태가 문제였습니다. 조건을 「이런 이벤트가 있었는가」(순간)로 적어 두면, 게이트는 동기화 이전에 이미 지나간 이벤트를 알 길이 없습니다. 같은 사실을 「PLC 타이머의 값이 이 범위인가」(상태값)로 고쳐 적으면 판정할 수 있습니다.

같은 실험의 조건식 30개 가운데 20개가 앞이나 뒤에 이벤트를 담고 있었고, 3가지로 나뉘었습니다.

종류	개수	판정 가능 여부
판정하는 명령 자체가 이벤트	8	그 시점 값으로 판정 가능
이벤트 앞섬, PLC 타이머 기억	3	상태값으로 고치면 가능
PLC도 기억하지 않는 이벤트	6~7	어떤 창으로도 판정 불가

셋째 종류는 조건식을 불러올 때 거부하고, 「막을 수 없는 위험」 목록에 이름을 올려 둡니다. 설계 규칙은 한 줄로 줄어듭니다. **PLC가 기억하지 않는 이벤트는 조건식에 쓸 수 없습니다.** 다만 앞 절의 인터록 제거 4건은

이 규칙으로 해결되지 않습니다. 그 4건의 조건식은 첫째 종류였고, 문제는 조건식이 아니라 검증용 프로그램이었기 때문입니다.

06

관찰 검사 앞에 프로그램 서명값 검사를 두는 것이 답이고, 이것은 새 권고가 아닙니다

동작으로 잡을 수 없다면 동작이 아닌 것으로 잡아야 합니다. PLC가 노출하는 프로그램 서명값(체크섬)을 읽거나, 프로그램을 올려 받아 해시를 구하고, 시운전 때 값과 다르면 「프로그램 변경됨」 상태로 들어가 가동 계열 명령을 전부 막습니다. 판정 순서는 아래처럼 바꿉니다.

- 1순위 — 프로그램 서명값. 시운전 때 값과 다르면 그 뒤는 보지 않습니다.
- 2순위 — 허용 목록. 명령을 낼 수 있는 주소와 값의 범위입니다.
- 3순위 — 동작 일치 검사. 서명값이 같을 때만 의미가 있습니다.
- 4순위 — 조건식 판정. 검증용 프로그램에서 미리 실행해 보는 단계입니다.

가동·허가 계열 명령은 한 가지를 더 합니다. 허용 조건 값을 검증용 프로그램이 아니라 판정 시점에 PLC에서 직접 읽습니다. 둘 다 실험으로 확인된 결과가 아니라 설계 결론이고, 다음 실험(7회차)에 「인터록 제거 4건은 변함없음, 조건식 형태 4건은 0으로」라는 예상 방향을 먼저 등록해 두었습니다.

이 방향은 새 권고가 아닙니다. 프로그램을 PLC에 넣기 전에 정적으로 검증하는 연구도, 운전 중인 로직을 내려받아 원본과 대조하라는 제안도 이미 있습니다. 이 실험이 보탠 것은 「왜 그것이 먼저여야 하는가」를 이런 게이트에서 숫자로 확인한 것입니다.

게이트가 명령 하나를 판정하는 4순위



게이트가 명령 하나를 판정하는 4순위 — 프로그램 서명값, 허용 목록, 동작 일치 검사, 조건식 판정의 순서

07

Modbus에는 서명값이 없어서, 이런 변경을 게이트가 볼 수 없습니다

중요한 단서가 하나 불립니다. 프로그램 서명값을 주지 않는 프로토콜에서는 1순위가 성립하지 않습니다. Modbus가 그렇습니다. 그런 현장에서는 게이트가 잠자는 로직 변경을 볼 방법이 없고, PLC 쓰기 보호와 변경 관리가 그 역할을 맡아야 합니다. 「게이트를 두었으니 프로그램 변경도 잡힌다」는 기대는 프로토콜을 확인한 뒤에만 성립합니다.

검사 자체의 비용도 적어 듭니다. 경계 상태에서 다시 풀리는 시간은 하한이 8초이고 상한은 확인되지 않았으며, 그 측정은 다음 실험으로 남겨 두었습니다. 입력이 1초 안에 바뀌는 라인이라면 검사는 잡는 것 없이 막기만 할 수 있습니다.

08

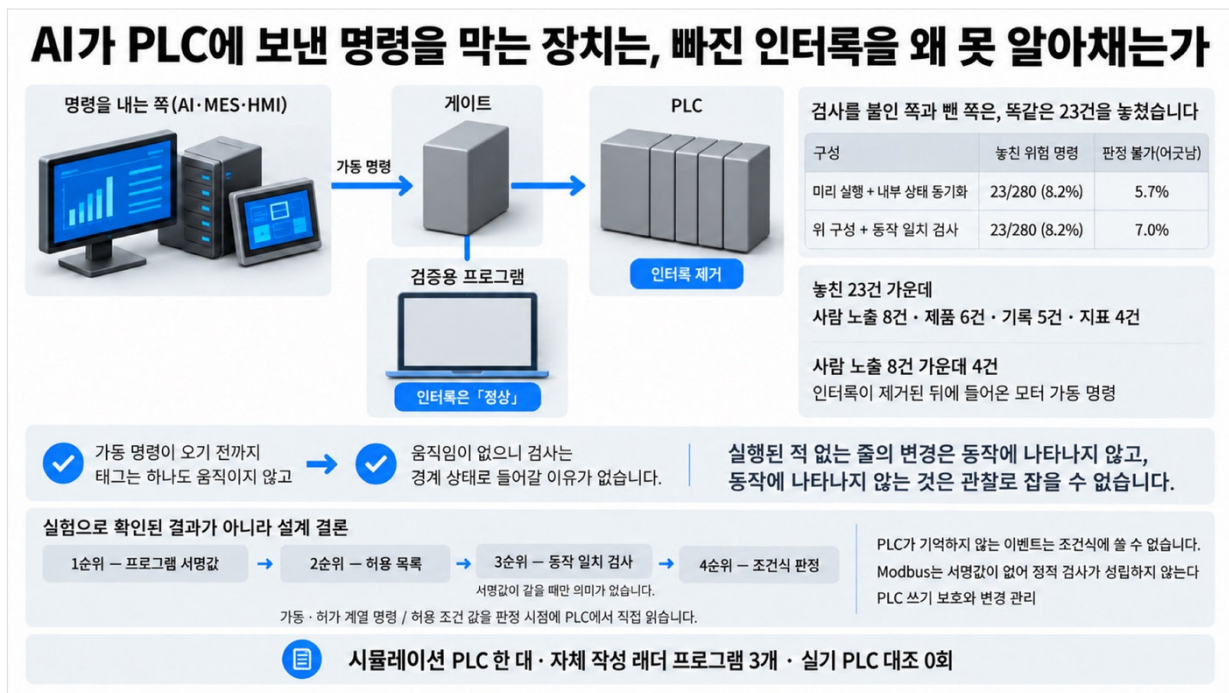
지금 쓰는 검증 장치에는 같은 질문 5가지를 던져 볼 수 있습니다

이 실험은 시뮬레이션 PLC 한 대와 자체 작성 래더 프로그램 3개로 한 것이고 외부 작성 프로그램도 실기 PLC 대조도 없어서, 숫자를 그대로 다른 현장에 옮길 수는 없습니다. 옮길 수 있는 것은 질문입니다.

- 프로그램 서명값을 읽는가 — 읽지 않으면 실행된 적 없는 변경은 어떤 검사로도 보이지 않습니다.

- **가동 명령의 허용 조건을 판정 시점에 현장에서 직접 읽는가** — 검증용 프로그램에서 읽으면 낯은 값입니다.
- **조건식에 PLC가 기억하지 않는 이벤트가 있는가** — 있으면 그 조건은 판정 불가이고, 목록에 이름을 올려야 합니다.
- **Modbus라면 쓰기 보호와 변경 관리가 있는가** — 게이트가 대신할 수 없는 일입니다.
- **경계 상태에서 나오는 시간의 상한이 있는가** — 없으면 가동 중인 라인에서 검사가 막기만 합니다.

5가지 질문 가운데 첫째가 「아니오」면, 나머지 4가지는 답해도 소용이 없습니다.



이 글을 한 장으로 요약하면 이렇습니다 — 동작 일치 검사가 잠자는 로직 변경을 못 보는 이유와, 서명값 검사를 먼저 두는 판정 순서

동작으로 최신인지 확인하는 검사는 동작이 있는 변경만 봅니다. 실행된 적 없는 줄의 변경은 동작에 나타나지 않으므로, 그 변경을 보려면 동작이 아니라 프로그램 자체를 확인하는 검사가 먼저 있어야 합니다.

WACE(웨이스)는 공장이 스스로 판단하고 멈추지 않게 하는 제조 AI 운영체제 VEXPLOR를 만드는 회사다.

09

함께 읽어보세요

- 설비 제어 로직을 소프트웨어로 실행해 보면, 처음 틀리는 곳은 3군데입니다
- AI에게 설비를 맡기려면, 틀린 명령이 나가기 전에 멈출 수 있어야 합니다
- 기술보고서 원문(영문, CC BY 4.0): WACE Technical Report TR-2026-01



원문	blog.vexplor.com/post/why-behavioral-checks-miss-dormant-logic
발행	2026년 9월 21일 · 방동걸 (주)웨이스 대표이사
회사	주식회사 웨이스 (WACE Inc.) · 대표이사 방동걸 · 사업자등록번호 439-87-02852
주소	(18462) 경기 화성시 동탄구 동탄영천로 150, 에이동(A동) 1702·1706호 (영천동, 실리콘엘리동탄)
연락처	전화 070-4730-0629 · contact@wace.me
홈페이지	wace.me · linkedin.com/company/wace-inc



원문



홈페이지



링크드인