Indokolatlan vészleállítási események fail safe PLC alkalmazásokban

Unexplained emergency stop events in fail safe PLC applications

JÁNYOKI Ákos Sándor
doktorandusz
témavezető: ***Dr. Nagy István*** egyetemi docens

Óbudai Egyetem Biztonságtudományi Doktori Iskola 1088 Budapest, József körút 6. Telefonszám: +36 (1) 666-7139

Abstract

Programmable logic controllers (PLCs) are the most widely used platforms in industrial process automation; for safety-critical applications they are available in fail-safe variants, typically up to Safety Integrity Level (SIL) 3. One of the most common safety functions is the emergency stop, whose purpose is to bring the equipment under control to a safe state. Industrial experience shows, however, that emergency stops may be triggered spuriously, often when the standard control program and the safety program share state information. This paper investigates the potential timing-related and architectural causes of such unjustified shutdowns. The analysis is based on Siemens S7-1200F/S7-1500F systems (F-runtime groups, F-process image snapshots, and OB priorities) and proposes measurement and diagnostic focal points to reproduce and mitigate the phenomenon.

Keywords: Safety PLC, S7-1500F, ciklikus interrupt, scheduling, jitter, monitoring timeout

Kivonat

Az ipari folyamatirányítás legelterjedtebb eszközei a programozható logikai vezérlők (PLC-k), amelyek biztonságkritikus alkalmazásokban hibabiztos (Fail Safe) kivitelben is elérhetők, jellemzően SIL 3 biztonsági integritási szintig. Az egyik leggyakoribb biztonsági funkció a vészleállítás, amelynek célja a vezérelt berendezés veszélytelen állapotba hozása. Ipari tapasztalatok szerint azonban előfordulhat, hogy a vészleállítás indokolatlanul aktiválódik, gyakran akkor, ha a standard és a safety programrészek közös állapotinformációkat használnak. A cikk ezen indokolatlan leállások lehetséges időbeli és architekturális okait vizsgálja. Az elemzés Siemens S7-1200F/1500F környezetre épül (F-runtime group, F-process image snapshot, OB-prioritások), és mérési/diagnosztikai fókuszokat ad meg a jelenség reprodukálásához és csökkentéséhez.

Kulcsszavak: Safety PLC, S7-1500F, ciklikus interrupt, ütemezés, jitter, monitoring timeout

1. BEVEZETÉS

A modern ipari automatizálási rendszerekben a funkcionális biztonság központi szerepet tölt be. A Safety PLC-k alkalmazása lehetővé teszi a komplex gépek és technológiai rendszerek magas biztonsági szintű üzemeltetését. Az ipari gyakorlat ugyanakkor egyre gyakrabban találkozik olyan vészleállásokkal, amelyek nem valódi veszélyhelyzetből, hanem látszólag indokolatlanul előálló körülmények hatására, egy nem kívánt, nem tervezett rendszerállapot során következnek be.

Ezek, a gyakran „spurious trip” néven említett biztonsági leállás események különösen ártalmasak, mivel termeléskiesést okoznak, nehezen reprodukálhatók és az előfordulásuk okai hagyományos hibakeresési módszerekkel nem, vagy csak nehezen felderíthetők.

A tapasztalatok szerint spurious trip esemény akkor fordulhat elő, ha a PLC standard (folyamatirányító) funkciója és a safety (biztonsági) funkciója valamilyen, gyakran kikerülhetetlen technológiai okból ugyanazt a logikai, vagy fizikai változót használja. Jelen cikk célja annak vizsgálata Siemens Safety PLC környezetben, hogy a spurious trip események visszavezethetők-e időbeli és architekturális okokra és milyen módon kerülhetők el.

A cikk bemutatja, hogy ezen jelenségek jelentős része nem konfigurációs vagy programozási hibából, hanem a standard és failsafe végrehajtási környezetek eltérő időmodelljéből ered. Matematikai időmodellek és

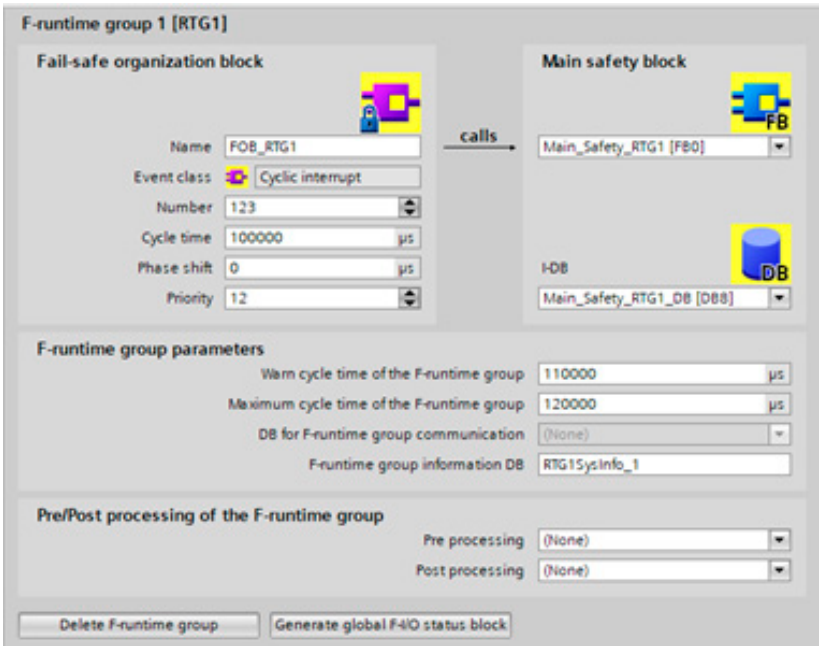
ipari példák segítségével elemezzük a standard ciklusidő, a failsafe ciklikus interrupt, valamint a process image (PI) alapú mintavételezés kölcsönhatását. A vizsgálat igazolja, hogy eseményjellegű információk közvetlen felhasználása diszkrét safety döntési környezetben determinisztikusan vezethet race window és monitoring-idő sérülésekhez. A cikk architekturális megoldási irányelveket fogalmaz meg az ilyen jelenségek megelőzésére.

2. VÉGREHAJTÁSI HÁTTÉR: SIEMENS SAFETY ARCHITEKTÚRA ÉS F-RUNTIME GROUP

A programozható logikai vezérlők (PLC-k) ciklikus működési elven alapulnak, amelynek során a vezérlő operációs rendszere ismétlődő végrehajtási ciklusokban olvassa be a bemeneteket, hajtja végre a felhasználói programot, majd frissíti a kimeneteket, kommunikál a környezetével és végez önellenőrzést. Siemens környezetben a működést az operációs rendszer által ütemezett programblokkok (OB-k) határozzák meg, amelyek különböző prioritással és indítási feltételekkel futnak. A standard vezérlési logika tipikusan ciklikus programblokkokban valósul meg, míg a biztonsági funkciók elkülönített, meghatározott időalapon futó programblokkokban kerülnek végrehajtásra. [9]

A standard programozás során az OB1-ből minden funkció blokkot (FB), és függvényt (FC) meg kell hívni, amely a technológiai folyamatot irányítja. Hagyományos, standard ciklikus interrupt OB-t és számos más típusú OB-t is tud a felhasználó a feladat jellegétől függően definiálni és abban programozni.

A biztonsági funkciók megvalósítására szolgáló, ciklikus interrupt által aktivált, gyakran jelszóval védett failsafe OB (általában OB123) teljesen függetlenül működik a standard OB-ktől. Az F-OB létrehozásakor automatikusan generálódik egy F-runtime group. Ez az F-runtime group a már említett ciklikus interrupttal működtetett failsafe OB-ból (F OB), az általa automatikusan aktivált failsafe FB-ből (Main_Safety_FB), és egy hozzá tartozó instance safety DB-ből (Main_Safety_ DB) áll. [8]



1. ábra. *F Runtime Group létrehozása Siemens TIA Portal mérnöki fejlesztő rendszerben*

Az F-runtime group komponenseit a rendszer hozza létre és a hívási struktúrát a rendszer kezeli; ugyanakkor az F-OB és az F-runtime group paraméterei (ciklusidő, fáziseltolás, prioritás, max. ciklusidő, main safety block stb.) a Safety Administration Editorban, a TIA Portal fejlesztő rendszerben konfigurálhatók. Általában az ábrán látható default beállításokkal a rendszerek legnagyobb része megfelelően működik.

A szükséges failsafe diagnosztikát és a safety akciót, tehát azt, hogy mit kell reagálni vész esetén, a safety FB-ben kell megírni, viszont más célú programrészeket nagyon nem javasolt itt megvalósítani. A safety FB programozása tehát általában csak egy egyszerű, vész logika megvalósítását jelenti. Lényeges viszont, hogy az információ, ami alapján ez a logika működik, minősített, megbízható legyen.

Egyetlen runtime group használata egy gyártási, termelési folyamatnál elegendő a folyamat felügyeletére.

3. JELÖLÉSEK ÉS RENDSZER-IDŐMODELL

A Siemens S7-1500 (és analóg módon S7-1200) végrehajtási modelljében az OB-k prioritás alapján, preemptív (megszakítható) módon futnak: ha magasabb prioritású esemény történik, a futó OB, pl. az OB1 megszakad, és a magasabb prioritású OB123 fut le. [1], [2]. Hogyan modellezzük a két OB-t az időtengelyen?

Standard ciklus (OB1): A standard ciklushatárt jelöljük t_S-sel, a (k+1)-edik ciklus kezdő pillanata:

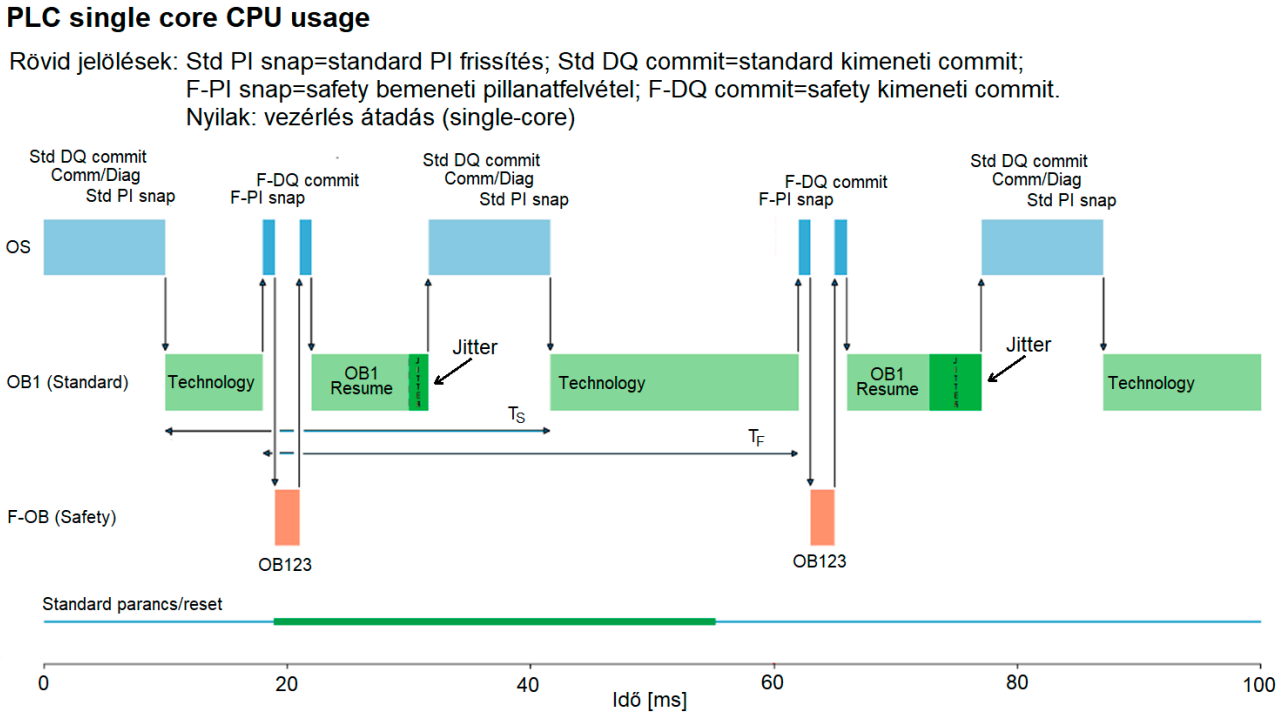
$$t_S(k+1)=t_S(k)+T_S(k) \quad k \in \mathbb{N}$$

ahol $T_S(k)$ a standard ciklusidő (jitteres, terhelés- és eseményfüggő).

Failsafe ciklus (F-OB, tipikusan OB123, ciklikus interrupt-tal): A failsafe futások kezdő időpontjai (a failsafe ciklus n-edik mintavételi időpontja:

$$t_F(n)=t_0+nT_F+\delta_F(n) \quad n \in \mathbb{N}$$

ahol t_0 a PLC indulásától az első ciklikus interrupt indulásáig eltelt idő, T_F az F-OB időalapja (interrupt periódusa, safety program tervezett futási gyakorisága), $\delta_F(n)$ pedig a késés/jitter, az n-edik failsafe futás eltérése az ideálistól, amely preemptálásból és OS-szintű ütemezésből származhat.

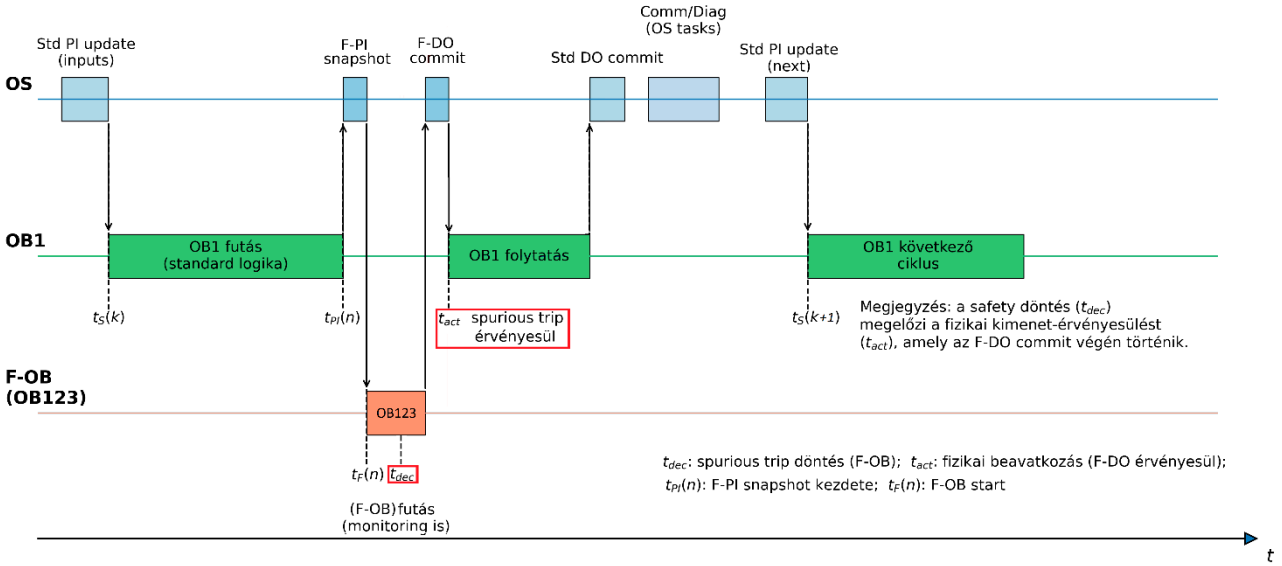


2. ábra. *A standard és a failsafe OB-k működése egy safety PLC-ben*

A modellezés real-time elméleti alapja a periodikus feladatok és rögzített prioritásos preemptív ütemezés klasszikus kerete. [3] A kiinduló konfiguráció tipikus default értékei: T_F=100 ms, OB123-prioritás P_F=12. Egymagos CPU-t feltételezve a programvégrehajtás időben egydimenziós: az OS és az OB-k váltakozva használják a CPU-időt. A standard ciklus valós futási időtartama (wall-clock time) ezért több komponensből áll:

$$T_S(k) = C_OS,pre(k) + C_{\mathrm{OB1}}(k) + C_OS,post(k) + \Sigma_j C_{preempt,j}(k)$$

ahol a $\Sigma_j$tag a magasabb prioritású megszakítások (pl. failsafe ciklikus interrupt, diagnosztika) által „betol-dott” idők összege.



3. ábra *Egymagos CPU OS, OB1 és F-OB végrehajtási idődiagramm, spurious trip döntés vs. fizikai beavatkozás*

4. CIKLIKUS INTERRUPT PARAMÉTEREK (T_F, P_F) JELENTŐSÉGE

Ütemezési és terhelési összefüggések

A két feladat egyszerű CPU-terhelési közelítése a PLC CPU átlagos kihasználtságát (utilization) becsüli meg két, egymással párhuzamosan létező végrehajtási kontextus hozzájárulásaként. A CPU-terhelés (kihasználtság) egyszerű közelítése két domináns végrehajtási kontextus (failsafe és standard) esetén:

$$U \approx \frac{\mathbb{E}[C_F]}{T_F} + \frac{\mathbb{E}[C_S]}{\mathbb{E}[T_S]}$$

ahol $C_F$és $C_S$rendre a failsafe, illetve standard végrehajtás CPU-futásideje, T_Fa failsafe ciklikus interrupt periódusa, míg T_S a standard ciklusidő (valós futási időtartam) átlagos értéke; $\mathbb{E}[\cdot]$időátlagot jelöl. Az OS-idő és az egyéb OB-k terhelése ebben a közelítésben a maradéktagban jelenik meg. A Liu–Layland típusú érvelés szerint Unövelése (különösen T_F csökkentésével) növeli a késések/jitterek kockázatát a rögzített-prioritású preemptív környezetben. [3]

„Időhiba” és OB80 kapcsolat Siemens környezetben

Siemens környezetben explicit követelmény: egy ciklikus interrupt OB futásideje legyen jelentősen rövidebb, mint az interrupt időköze; ellenkező esetben a rendszer time error eseményt generál, és OB80 (Time error interrupt OB) futhat, illetve a ciklikus interrupt később kerül végrehajtásra, vagy eltolódik. [4], [5], [6]

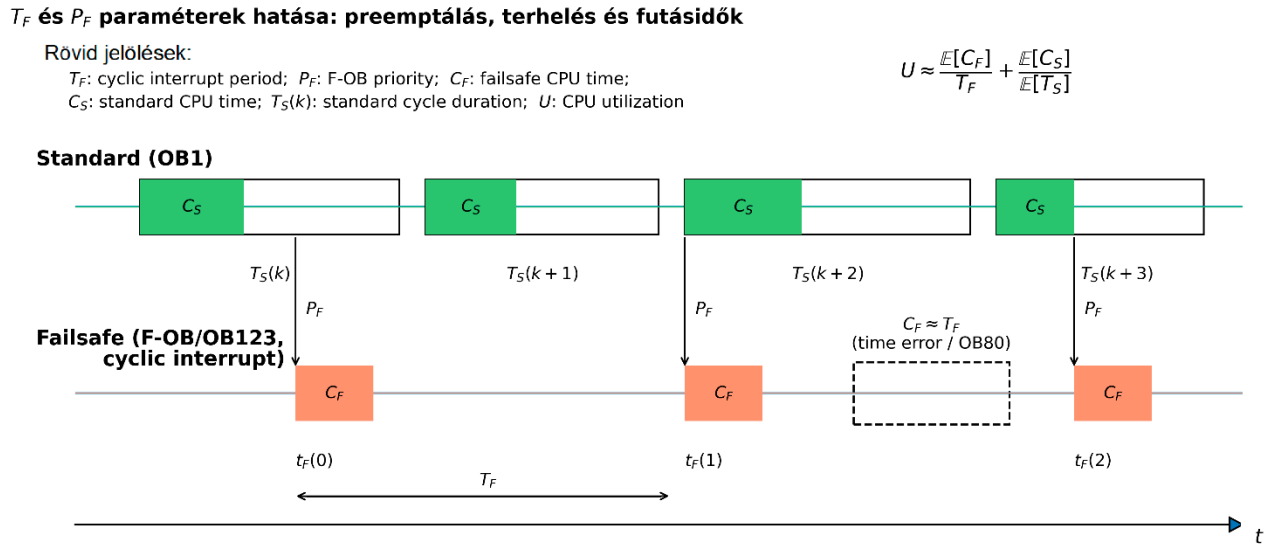
Ez a fenti modellben azt jelenti, hogy $\delta_F(n)$megugrik (vagy az effektív F-ciklusidő „szétesik”), ami safety monitoring időablakok sérüléséhez vezethet.

F-runtime group paraméterek és felügyelet

Láttuk, hogy F-runtime group-hoz F-OB és main safety FB tartozik; az F-CPU az F-ciklusidőt felügyeli, és ehhez F-runtime group information DB is generálódik. Ebből adódóan az F-ciklusidő és annak betartása safety szempontból felügyelt mennyiség. [7], [8]

Kedvezőtlen együttállás (spurious trip / indokolatlan leállás)

$T_F$csökkentése → $\mathbb{E}[C_F]/T_F$ nő → Unő → nő a standard jitter ($T_S(k)$ szórása), és nőhet $\delta_F(n)$is. Ha P_F (a failsafe OB123 végrehajtási prioritása) túl magas → OB1 erősen „szabdalt” lesz, így a standard jel-előállítás és adatátadás kevésbé determinisztikus lesz. $C_F$növelésére (pl. nem safety célú logika F-ben) → az időhibák valószínűsége nő. Ez egyértelműen megerősíti a gyakorlati szabályt: más célú programrészeket ne itt valósítsunk meg.



4. ábra. *$T_F$és P_F paraméterek hatása: preemptálás, terhelés és futásidők*

5. ÜTEMEZÉSI ANALÍZIS: OB1 VÁLASZIDŐ ÉS PREEMPTÁLÁSOK SZÁMA

Tekintsük az F-OB-t periodikus, magas prioritású feladatnak $T_F$periódussal és $C_F$futásidővel. Az OB1 egy alacsonyabb prioritású feladat C_{OB1} futásidővel. Fixed-priority rendszerben az OB1 valós futási időtartam szerinti válaszideje, vagyis a kezdéstől a befejezésig eltelt R_{OB1}idő egy rögzítettpont-egyenlettel felső becslést kaphat:

$$R_{\mathrm{OB1}} = C_{\mathrm{OB1}} + \left\lceil \frac{R_{\mathrm{OB1}}}{T_F} \right\rceil \cdot C_F + C_{\mathrm{OS,max}}$$

ahol $C_{\mathrm{OS,max}}$az OB1 futása közben előforduló OS/diagnosztikai szolgáltatások konzervatív felső becslése. A $\lceil R_{\mathrm{OB1}}/T_F \rceil$tag azt fejezi ki, hogy a teljes futás során hány ciklikus interrupt „fér bele”, azaz hányszor szakíthatja meg az F-OB az OB1-et. A képlet iterációval megoldható:

$$R^{(0)} = C_{\mathrm{OB1}} + C_{\mathrm{OS,max}}, \text{ majd } R^{(i+1)} = C_{\mathrm{OB1}} + \left\lceil R^{(i)}/T_F \right\rceil \cdot C_F + C_{\mathrm{OS,max}}.$$

A modell fontos következménye, hogy a standard ciklusidő nem lineárisan nő $T_F$csökkentésével, mert a preemptálások száma diszkrét lépcsőkben nő. Ez a lépcsősség magyarázza, hogy miért jelenhetnek meg ritka, de jelentős ciklusidő-csúcsok (outlierek), amelyek monitoring timeoutot okoznak.

CPU kihasználtság közelítése és jitter metrikák

A PLC-ben tipikusan több feladat verseng a CPU-időért; mérnöki közelítésként hasznos a terhelés bontása. A Liu–Layland típusú érvelés szerinti képlet (lsd 4.fejezet) intuitívan azt mondja: minél gyakrabban fut a failsafe (kisebb T_F), illetve minél hosszabb a failsafe futás (nagyobb C_F), annál nagyobb a safety részarány; ugyanez igaz a standard részre (C_S és T_S). A spurious trip ok-okozati lánca azonban nem az Uátlagértékében, hanem a jitterben jelenik meg, ezért célszerű explicit metrikákat vizsgálni: Var(T_S), P99(T_S), max(T_S), továbbá a failsafe időrács eltérése $\delta_F(n)$és annak varianciája Var(δ_F). A monitoring sérülések tipikusan a felső percentilisekhez kötődnek, nem az átlaghoz.

Time error (OB80) és a failsafe időrács torzulása

Ciklikus interrupt esetén időhiba akkor keletkezik, ha az interrupt kérés kiszolgálása késik, vagy az OB futása nem fejezhető be időben. A jelenség az időmodellben $\delta_F(n)$megugrásaként, illetve az effektív periódus növekedéseként jelenik meg. Monitoring szempontból ez kritikus, mert a failsafe logika sokszor ciklusszámlálón (n darab ciklus) és nem abszolút időn mér, így, ha a ciklus „szétesik”, a valós időablak is megváltozik. Ez rejtett, nehezen észrevehető kockázatot jelent, ha a monitoring paramétereket ciklusszám alapján választják meg.

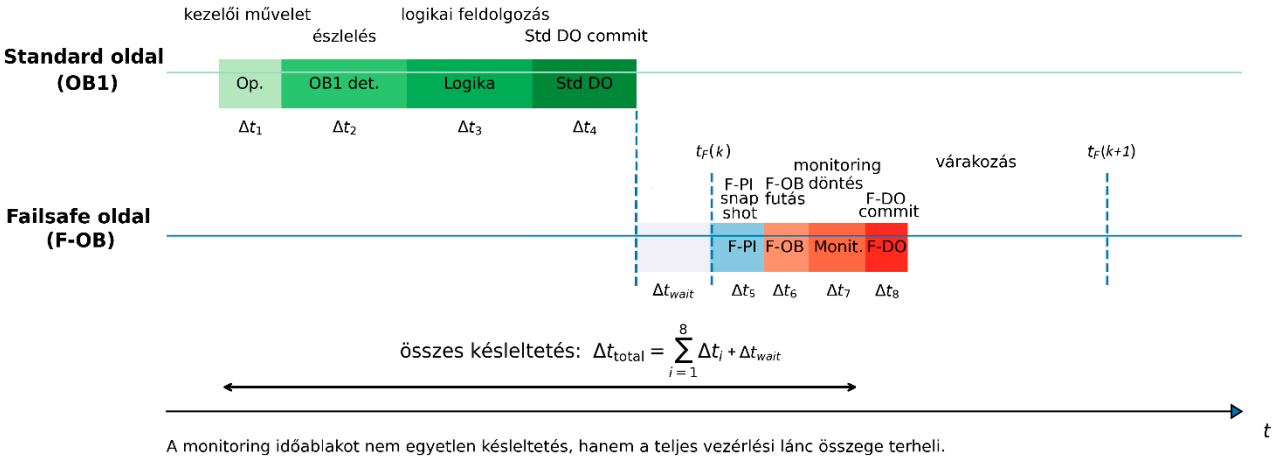
Monitoring ablak: késleltetési lánc és spurious trip feltétel

A monitoring sérülés akkor következik be, ha a teljes válaszidő meghaladja a megengedett ablakot:

$$\Delta t_{\text{total}} = \Sigma\, \Delta t\_i,$$

$$\Delta t_{\text{total}} > T_{\text{mon}} \Rightarrow \text{spurious trip.}$$

A $\Sigma\, \Delta t\_i$ komponensekhez tartozik: (1) standard jel keletkezése és OB1 feldolgozása, (2) standard DO commit, (3) safety PI snapshot és F-OB futás, (4) safety DO commit, (5) fizikai folyamat késleltetései (aktuátor/szenzor), (6) kommunikáció és diagnosztika. A vizsgálatban ezért célszerű a késleltetések komponensszintű felbontása (időbélyegek) és worst-case összegezése.



5. ábra. *Pipeline-késleltetések: a komponensek összeadódása a monitoring ablakhoz képest*

6. ESETTANULMÁNY: SPURIOUS TRIP KIALAKULÁSA A CPU IDŐDIAGRAM ALAPJÁN

Az 3. ábrán jelölt spurious trip pillanat tipikus „időzítési találkozás” eredménye. A standard OB1 a ciklus elején feldolgozza a reset/permissive logika egy részét, azonban a failsafe ciklikus interrupt megszakítja a standard futást. Az OS a megszakításra reagálva elkészíti a safety PI (process image) snapshotot, majd lefuttatja az F-OB-t, amely a snapshot alapján értékeli a monitoringot és a permissive feltételeket. Mivel a snapshot időpontja eltér a standard logika frissítési időpontjától, előállhat olyan minta, hogy a safety oldalon a permissive még nem aktív (vagy már nem aktív), miközben a standard oldalon a reset impulzus már „lefutott”. A safety FB így nem talál olyan ciklust, ahol a két feltétel egyszerre igaz lenne, ezért a monitoring időzítő lejár.

Más szóval a fizikai bemenetek és a standard logika önmagukban helyesek lehetnek, de a snapshot-időpontok és a preemptív ütemezés miatt a safety döntési pontok olyan időmetszetet látnak, amely a standard logika szemszögéből nem „egyidejű”. Ezért nevezhető a jelenség determinisztikus, de fázisfüggő mintavételi hibának.

7. KÍSÉRLETI BEÁLLÍTÁS ÉS ADATGYŰJTÉS JAVASLAT

A jelenség labor- vagy üzemi reprodukálásához célszerű kontrollált terhelési profilokat létrehozni:

- baseline: minimális kommunikáció és diagnosztika;
- kommunikációs csúcs: HMI nagy tag-szám, Profinet IO frissítés sűrítése;
- diagnosztikai csúcs: trace, online monitoring;
- kombinált csúcs: utóbbi kettő együtt.

Minden profil mellett mérendő: C_F (min/avg/max), $T_S$eloszlás, $t_F(n)$ idősor, valamint a spurious trip esemény időbélyege. A cél, hogy $\delta_F$és T_S outlierek megjelenése korreláljon a monitoring sérülésekkel, így a matematikai modell paraméterei ($C_{\text{OS,max}}$, Δt_i) becsülhetővé válnak.

8. ÖSSZEFOGLALÁS ÉS KÖVETKEZTETÉSEK

A vizsgált esetek, melyekben a standard és a safety oldal által közösen használt, nem safety forrású állapotinformációk indokolatlan vészleállásokat okozhatnak, időzítés szempontjából így összegezhető:

Ha a safety logika nem safety forrásból dolgozik, akkor a jel nem determinisztikusan mintavételezett, és két időalap között keletkezik.

Ha több bit/állapot együtt értelmezett, akkor a nem atomi frissítés miatt előállhat inkonzisztens mintavétel.

Ha a safety monitoring időablakokat (timeoutokat) használ, akkor a késleltetések összeadódása miatt előállhat, hogy a teljes válaszidő meghaladja a megengedett ablakot:

Fentiekből következő, lehetséges spurious trip okok Siemens környezetben:

- OB1–OB123 aszinkronitás
- A safety oldal olyan állapotkombinációt lát, amely fizikailag nem létezett.
- Standard oldali impulzusok elveszhetnek.
- A standard oldali állapot a safety oldal számára nem determinisztikus késleltetéssel jelenik meg (pipeline).
- A monitoring időablakok túl szorosak a valós késleltetési lánchoz képest.
- A CPU terhelés és az ütemezés outlier jellegű ciklusidő-csúcsokat okoz.
- Time error esetén a failsafe időrács torzul és a monitoring „megrövidülhet” ciklusszám-alapú beállításnál.

Összegzés és következtetés:

A spurious trip jelenségek vizsgálata során a fő következtetés az, hogy az indokolatlan vészleállások jelentős része az időzítési aszinkronitás következménye, amely a standard és a safety programrészek eltérő végrehajtási modelljéből származik. A PLC operációs rendszerének preemptív ütemezése miatt a standard OB1 és a failsafe OB123 futása nem determinisztikus módon váltakozhat, és a safety logika diszkrét snapshot-időpontokban hozza meg döntéseit. Ez olyan időablak jellegű jelenségeket (race window) eredményezhet, ahol a valós időben teljesülő feltételek nem esnek egyetlen safety mintavételi pillanatba sem.

A vizsgálat azt is igazolta, hogy a monitoring-idő sérülések gyakran nem egyetlen okra vezethetők vissza, hanem késleltetések összeadódására: standard program feldolgozás, OS process image kezelése, safety snapshot és commit, fizikai aktuátor/szenzor késleltetések, valamint kommunikációs jitter együttesen lépik át a safety által elvárt időablakot. Redundáns vagy több PLC-t érintő architektúrákban további kockázatot jelent a különböző PLC-k eltérő failsafe időalapja és a kommunikáció időbeli szórása.

Végül, a cikk a hibák megelőzésére két irányt emel ki:

- az időmodellhez illesztett, méréseken alapuló paraméterválasztást (T_F, P_F, T_{mon}), és
- a standard–safety interfészen a determinisztikus állapotátadást (handshake, konzisztencia), amely kizárja az impulzusvesztést és a félmásolásból adódó inkonzisztens állapotokat.

A spurious trip ezért nem pusztán „programhiba”, hanem időzítési és architekturális jelenség, amely csak rendszer-szintű tervezéssel kezelhető.

A safety funkciók a safety chain tagjai, tehát a safety chain elemeinek megfelelőségét igazolni kell. A safety chain elemei két csatornásak kell, hogy legyenek, rendelkezniük kell diagnosztikával (DC érték), és CCF követelményt (környezet, kábelezés, EMC) teljesíteniük kell. Egy standard jel egyiknek sem felel meg, azaz invalidálja a safety funkciót.

Következtetésként megfogalmazható, hogy a spurious trip gyakran nem „logikai bug”, hanem időzítési jelenség. A megoldás ezért

- időmodell felállítása,
- worst-case/percentilis alapú mérések,
- monitoring ablakok és ciklikus interrupt paraméterek tartalékkal történő megválasztása, valamint
- ahol standard↔safety interfész szerepel, ott determinisztikus állapotátadás alkalmazása (handshake, konzisztencia) – ez utóbbit egy készülőben lévő társtanulmány részletesen fogja tárgyalni.

Irodalomjegyzék

[1] Siemens. Events and OBs (S7-1500) – STEP 7 Manual Collection S7-1500/ET 200MP
[2] Siemens. Events and OBs – Response to triggers – Manual Collection S7-1500/ET 200MP
[3] Liu, C. L., Layland, J. W. (1973). Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *Journal of the ACM*, 20(1), 46–61. (doi: 10.1145/321738.321743)
[4] Siemens Industry Support. Cyclic interrupt OBs – STEP7 https://support.industry.siemens.com/cs/mdm/109011420 Siemens Support
[5] SIMATIC STEP 7 Basic/Professional V18 és SIMATIC WinCC V18 (Entry/ID: 109815056) https://support.industry.siemens.com/cs/mdm/109815056?c=133595508747&lc=en-HU SiePortal
[6] Siemens. Time error interrupt OB (OB80) – Fail-safe modules Manual Collection –docs.tia.siemens.cloud
[7] Siemens. Safety Integrated – Commissioning Manual https://cache.industry.siemens.com/dl/files/143/109817143/att_1129499/v1/ONE_SI_commis_man_0123_en-US.pdf Siemens Support
[8] Siemens. Procedure for defining an F-runtime group (S7-1200/1500) – TIA Portal dokumentáció
[9] Siemens. “Safety Programming Guideline for SIMATIC S7-1200/1500” (Entry ID 109750255)