life.augmented

Sicurezza funzionale in Sistemi elettronici: Principi e Applicazioni

Alessandro Bastoni

Esperto in sicurezza funzionale

STMicroelectronics

Lezione n. 6

Architetture di sicurezza formali (IEC61508, ISO13849) + metodi ISO26262 (decomposizione ASIL). Mappatura su casi d'uso automotive e robotica

Riepilogo:

- **Concetti generali (HFT)**
- **Architetture IEC 61508 (1oo1,1oo2,2oo2)**
- **Decomposizione ASIL, teoria ed esempi**
- **Considerazioni sul ruolo del software**

life.augmented

A proposito del concetto di “diversità”

La diversità è definita nella norma IEC61508-4 “diversi mezzi per eseguire una funzione richiesta”

In alcune circostanze può essere richiesta la diversità (ad esempio per utilizzare la regola di composizione SC+1 con canali ridondanti)

La diversità è la risorsa chiave per combattere i fallimenti delle cause comuni (e per far fronte alla loro potenziale analisi incompleta)

La diversità è possibile sia nell'hardware che nel software.

Per il software, è possibile ottenere anche la diversità temporale (lo stesso calcolo viene eseguito in momenti diversi) fornendo protezione contro gli errori software che hanno un impatto sulla CPU

La diversità è un vantaggio ma anche un costo (tempo di progettazione, risorse hardware, spazio di memoria, complessità, maggiore verifica, potenziali problemi di disponibilità).

La diversità può essere utilizzata come fattore di mitigazione per i guasti degli utensili (procedura di valutazione degli utensili per utensili T3)

Tolleranza ai guasti hardware (HFT)

La tolleranza ai guasti hardware di N significa che N+1 è il numero minimo di guasti che potrebbero causare una perdita della funzione di sicurezza.

Per determinare l'HFT non è possibile prendere in considerazione altre misure che controllano l'effetto dei guasti (come la diagnostica)

Alcuni guasti possono essere esclusi dalle considerazioni, sulla base di motivazioni specifiche basate sulla loro probabilità.

ÿHFT è fondamentalmente una proprietà dei sottosistemi, non dei componenti.

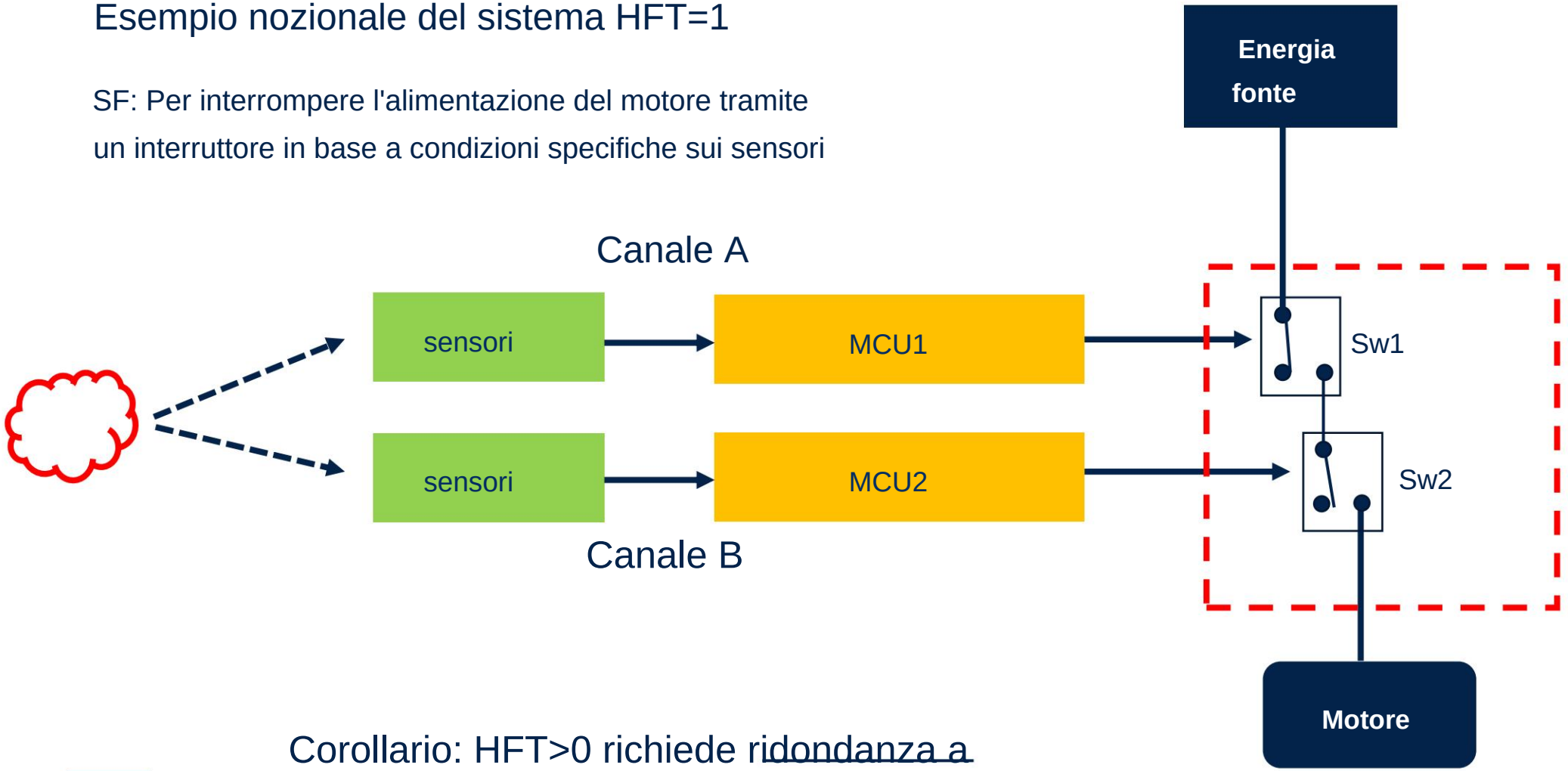
Ricorda: HFT è una delle voci per stabilire gli obiettivi SFF per i sistemi hw

Nota: IEC 61508 include requisiti di architettura speciali per circuiti integrati con ridondanza on-chip

Tolleranza ai guasti hardware (HFT)

Esempio nozionale del sistema HFT=1

SF: Per interrompere l'alimentazione del motore tramite un interruttore in base a condizioni specifiche sui sensori



Corollario: HFT>0 richiede ridondanza a un certo livello

Che cosa è un'architettura di sicurezza

In generale, una buona definizione di ***architettura*** è "la rappresentazione della struttura di un sistema che consente l'identificazione dei componenti fondamentali, dei loro confini e delle loro interfacce, e include l'allocazione dei requisiti a questi componenti". Questa è una sorta di definizione generale.

Non esiste una definizione esatta di "architettura di sicurezza" negli attuali standard di sicurezza. Questo può essere dedotto, analizzando il framework IEC 61508, in questo modo:

L'architettura di sicurezza definisce la struttura e l'organizzazione del sistema correlato alla sicurezza, comprese le funzioni di sicurezza, i meccanismi di sicurezza, gli stati sicuri e la loro allocazione agli elementi hardware e software per ottenere l'integrità di sicurezza richiesta.

IEC 61508: 1oo1

1oo1 è l'architettura a canale singolo (PE = Processing Element)

È lo schema architettonico più semplice ma anche il più impegnativo / NESSUN vantaggio!
Tutto si risolve all'interno della struttura semplice.

La corrente continua deve essere garantita dalla diagnostica intrinseca (HW/SW)

HFT = 0

La transizione di stato sicura potrebbe essere complessa

Non è possibile alcuna diversità nell'hardware ÿ SC hardware da raggiungere a livello di componente

Diversità nel software difficilmente possibile ÿ SC del software da raggiungere a livello di componente

IEC 61508: 1oo1 - formule

Modalità a bassa richiesta:

$$PFD_G = (\lambda_{DU} + \lambda_{DD})t_{CE} \qquad t_{CE} = \frac{\lambda_{DU}}{\lambda_D}\left(\frac{T_1}{2} + MRT\right) + \frac{\lambda_{DD}}{\lambda_D} MTTR$$

Modalità continua/alta richiesta:

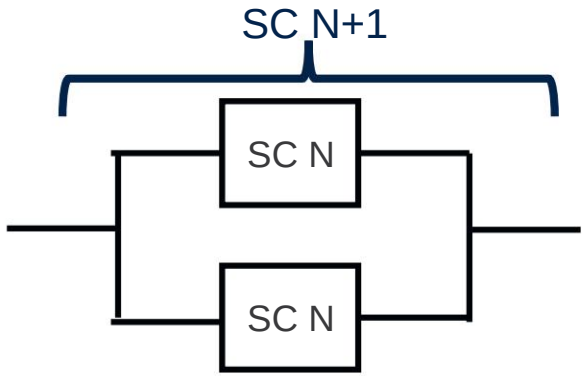
Se si presume che il sistema di sicurezza metta l'EUC in uno stato sicuro al rilevamento di qualsiasi fallimento, per un'architettura 1oo1 si ottiene quanto segue

$$PFH_G = \lambda_{DU}$$

IEC 61508: 1oo2

Sono possibili molteplici tipi di diversità nell'hardware ÿ l'hardware SC può essere più semplice grazie a Regola N+N = N+1

La diversità nel software è possibile ÿ il software SC è più semplice grazie alla regola N+N = N+1



Attenzione: ÿ

È richiesta l'indipendenza tra gli elementi ÿ Consentita solo una volta (non più composizioni a cascata)

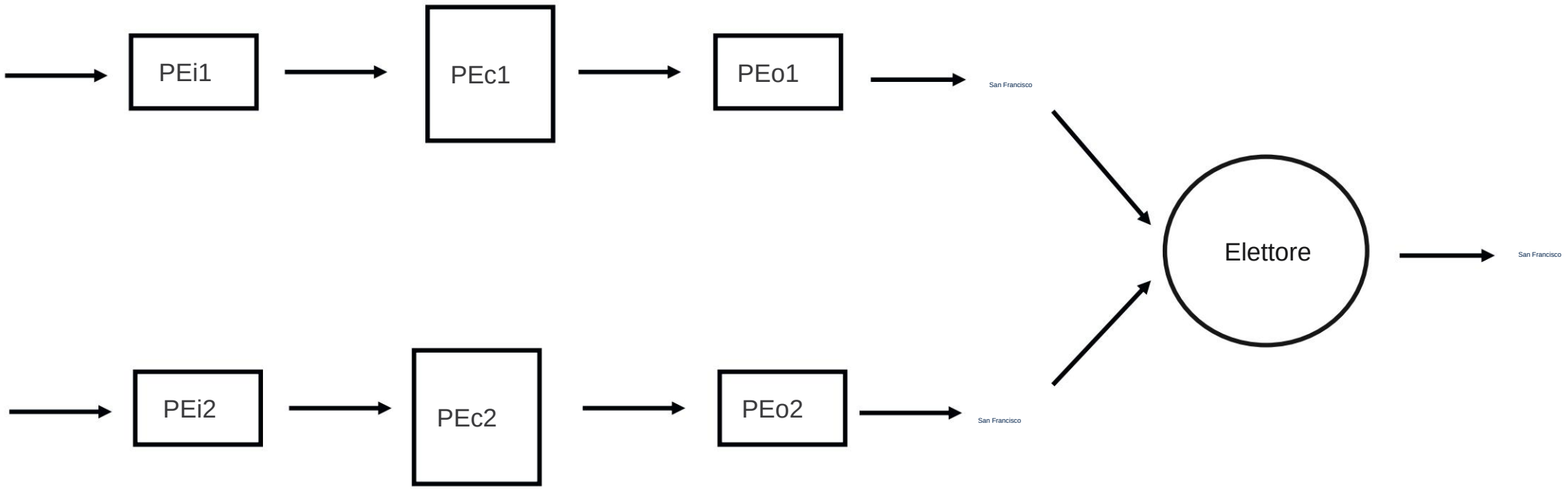
MA ancora alcune complicazioni

Cause comuni di guasti da esplorare (metriche di impatto sulla sicurezza!)

L'elettore deve essere HFT = 1 intrinsecamente

life.augmented

IEC 61508: 1oo2



Nota: tenere presente che la collaborazione incrociata tra i due MCU può portare ad un aumento dei fattori di penalità ÿ e ÿD (causa comune di guasti tra MCU)

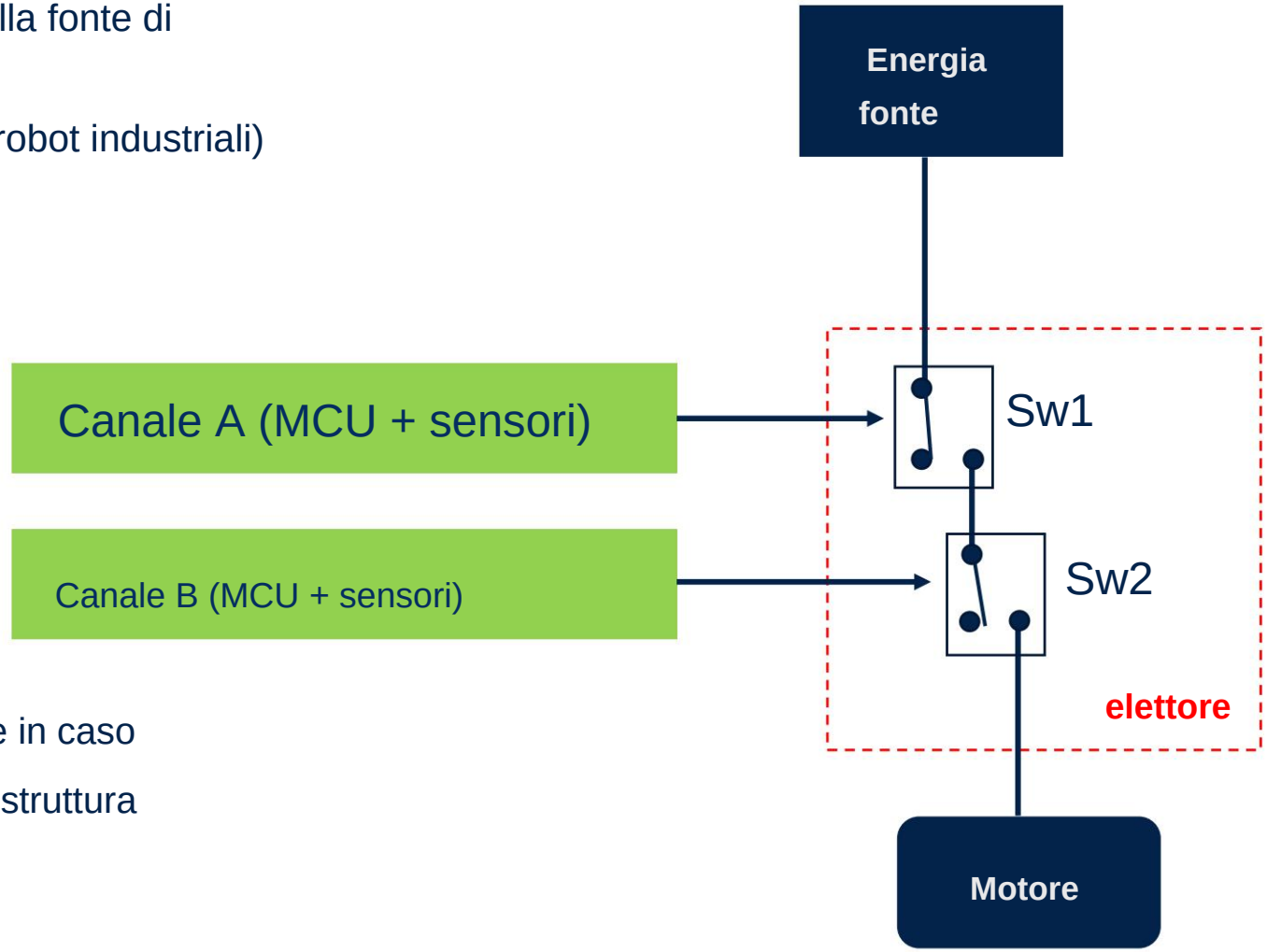
life.augmented

Esempio nozionale 1oo2

Funzione di sicurezza: "Collegamento motore aperto alla fonte di alimentazione in condizioni di segnali di ingresso specifici" (applicazione ad esempio barriera attiva per robot industriali)

Stato sicuro: collegamento elettrico APERTO

La natura di questa funzione di sicurezza consente una facile implementazione per gli elettori



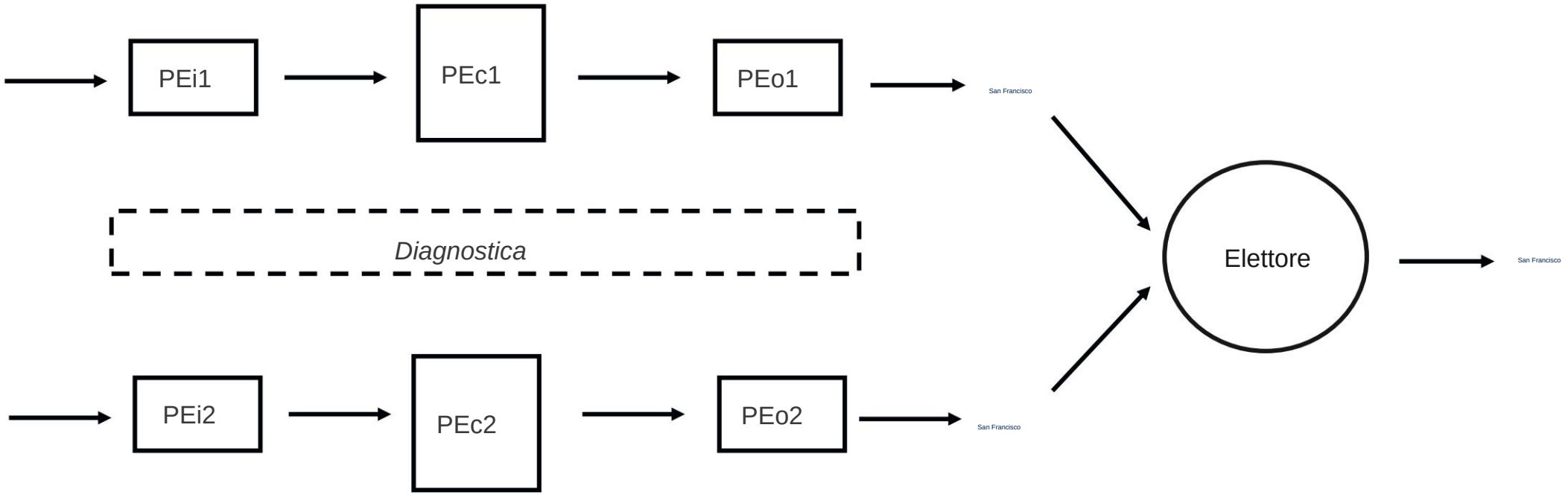
Nota: la struttura dell'interruttore forza la disconnessione in caso di guasto dell'alimentazione locale dell'elettore. Per una struttura HFT=1 reale dell'elettore, gli interruttori devono essere implementati in modo ridondante.

IEC 61508: 2oo2

2oo2 è costituito da due canali collegati in parallelo, in modo che entrambi i canali debbano richiedere la funzione di sicurezza prima che questa possa aver luogo.

La corrente continua deve essere garantita da una diagnostica intrinseca per ciascun canale. La diagnostica segnalerebbe guasti ma non modificherebbe il voto.

Lo schema è HFT = 0. PFH = 2ÿDU

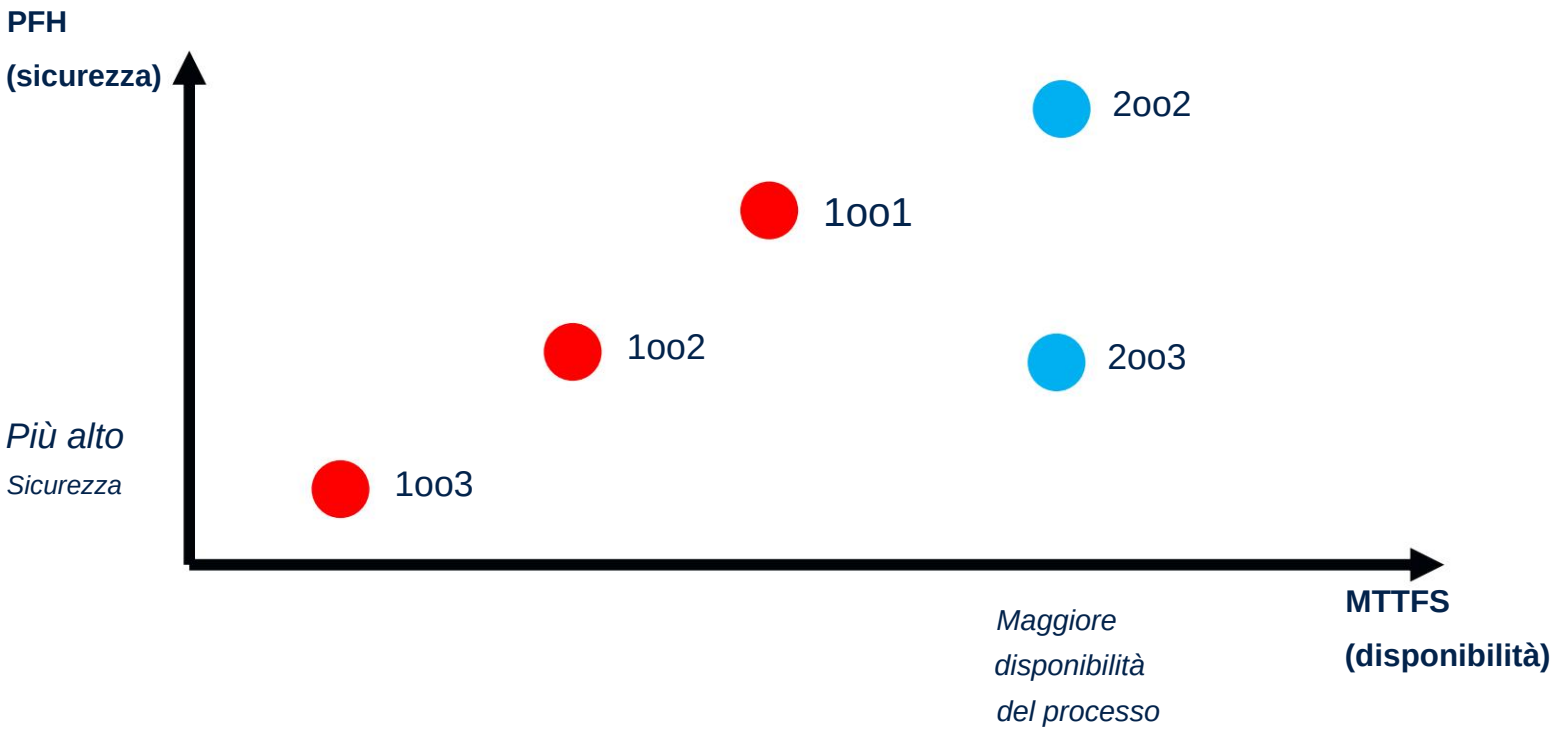


1oo2 contro 2oo2 contro 2oo3

1oo2: la priorità è la sicurezza (PFH inferiore)

2oo2: la priorità è sulla disponibilità (PFH più elevato ma meno transizioni di stato sicuro false positive)

2oo3: mantiene i vantaggi dei due schemi precedenti, a un costo più elevato



Note: MTFS = Tempo medio di guasto spurio; il grafico sopra è solo qualitativo

Decomposizione ASIL

La decomposizione ASIL è descritta nella clausola 5 della norma ISO 26262-9:2018.

Lo scopo principale dell'implementazione della decomposizione ASIL è quello di abbassare il livello ASIL.

Un requisito di sicurezza viene scomposto in requisiti di sicurezza ridondanti, che vengono poi assegnati a elementi sufficientemente indipendenti.

Il processo di decomposizione ASIL deve rispettare lo schema indicato nella Tabella 1 della norma ISO 26262-9:2018.

D	ASIL D(D) + QM(D)
	ASIL C(D) + ASIL A(D)
	ASIL B(D) + ASIL B(D)
C	ASIL C(C) + QM(C)
	ASIL B(C) + ASIL A(C)
B	ASIL B(B) + QM(B)
	ASIL A(B) + ASIL A(B)
A	ASIL A(A) + QM(A)

life.augmented

Decomposizione ASIL – regole generali

La ridondanza omogenea (ad esempio, tramite dispositivo o software duplicato) non è, in generale, sufficiente per ridurre l'ASIL a causa della mancanza di indipendenza tra gli elementi

Gli obiettivi di sicurezza non possono essere scomposti

I requisiti scomposti devono soddisfare in modo indipendente il requisito originale (definizione di ridondanza)

La decomposizione ASIL può essere applicata più di una volta (!) (grande differenza con la combinazione di elementi in IEC 61508)

Qual è la scommessa: i requisiti di sicurezza scomposti sono più semplici e/o più economici in termini di implementazione. Quindi: la scomposizione ASIL non è sempre una soluzione vincente...

Decomposizione ASIL – regole generali

(INVARIANTI)

L'ASIL decomposto risultante deve essere espanso con l'ASIL originale prima della decomposizione, ad esempio ASIL A(C)

La valutazione della sicurezza funzionale dell'articolo è definita dall'ASIL originale

L'indipendenza degli elementi decomposti deve essere valutata mediante l'analisi dei guasti dipendenti

I requisiti per il test e l'integrazione sono definiti dall'ASIL del livello di integrazione pertinente

I valori target per le metriche HW sono definiti dall'ASIL originale a livello di elemento

Software correlato alla sicurezza vs software non correlato alla sicurezza

Nelle architetture articolate, il software è spesso parte del concetto di sicurezza.

La coesistenza tra software correlato alla sicurezza e software non correlato alla sicurezza può verificarsi nei sistemi basati su microcontrollori (spesso per ragioni di costo, ad esempio riutilizzo di software open source per attività non correlate alla sicurezza).

Il problema principale è la separazione, ovvero garantire che il software non correlato alla sicurezza non possa interferire con la funzione di sicurezza:

- ÿ Interferenza spaziale: sovrascrittura della memoria, accesso alle periferiche ecc.
- ÿ Interferenza temporale: violazione del determinismo, occupazione delle risorse, jitter, mascheramento degli interrupt ecc.

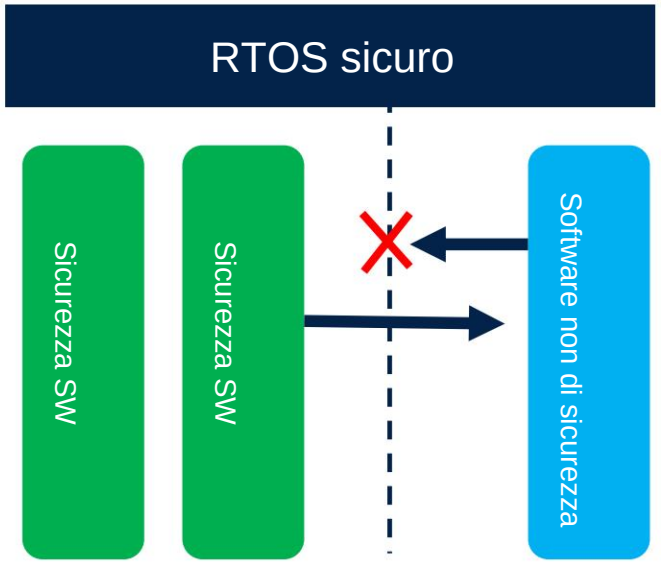
La segregazione dell'hardware può essere difficile poiché i tipici microcontrollori non hanno MMU complesse (spesso è disponibile solo una MPU di base)

Software correlato alla sicurezza vs software non correlato alla sicurezza

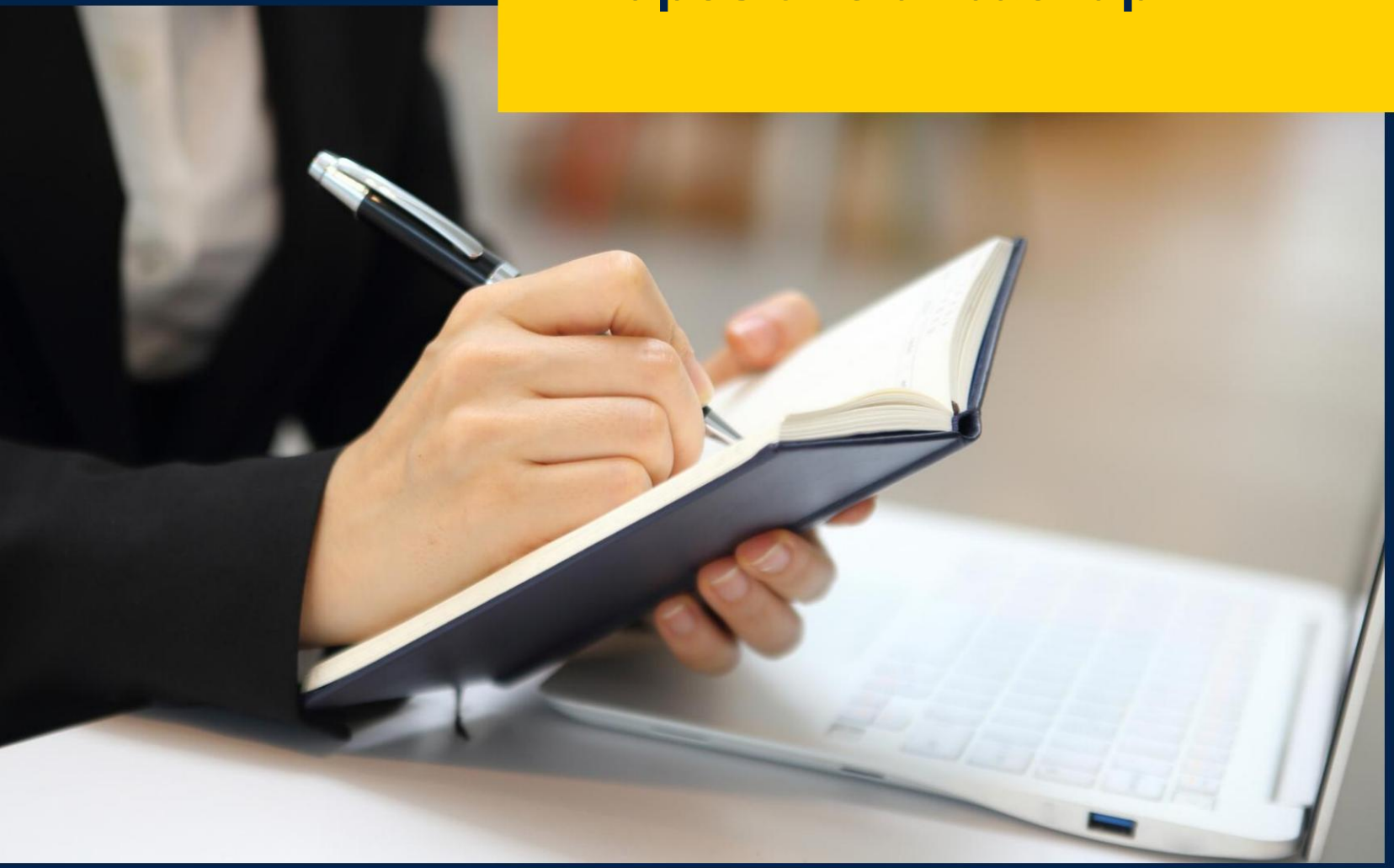
Possibili vettori per la soluzione del problema della coesistenza:

Il software non correlato alla sicurezza è semplice: potrebbe essere fattibile rimpatriarlo all'interno del modello V certificato per il software applicativo

Il software non di sicurezza è complesso: la segregazione del software tramite un RTOS sicuro potrebbe garantire a) l'isolamento delle potenziali interferenze spaziali b) il determinismo del software di sicurezza indipendentemente da quello non di sicurezza. I costi generali sono rappresentati dal costo e dalla disponibilità del RTOS sicuro.



Diapositive di backup



IEC 61508: 1oo2 - formule

Modalità a bassa richiesta:

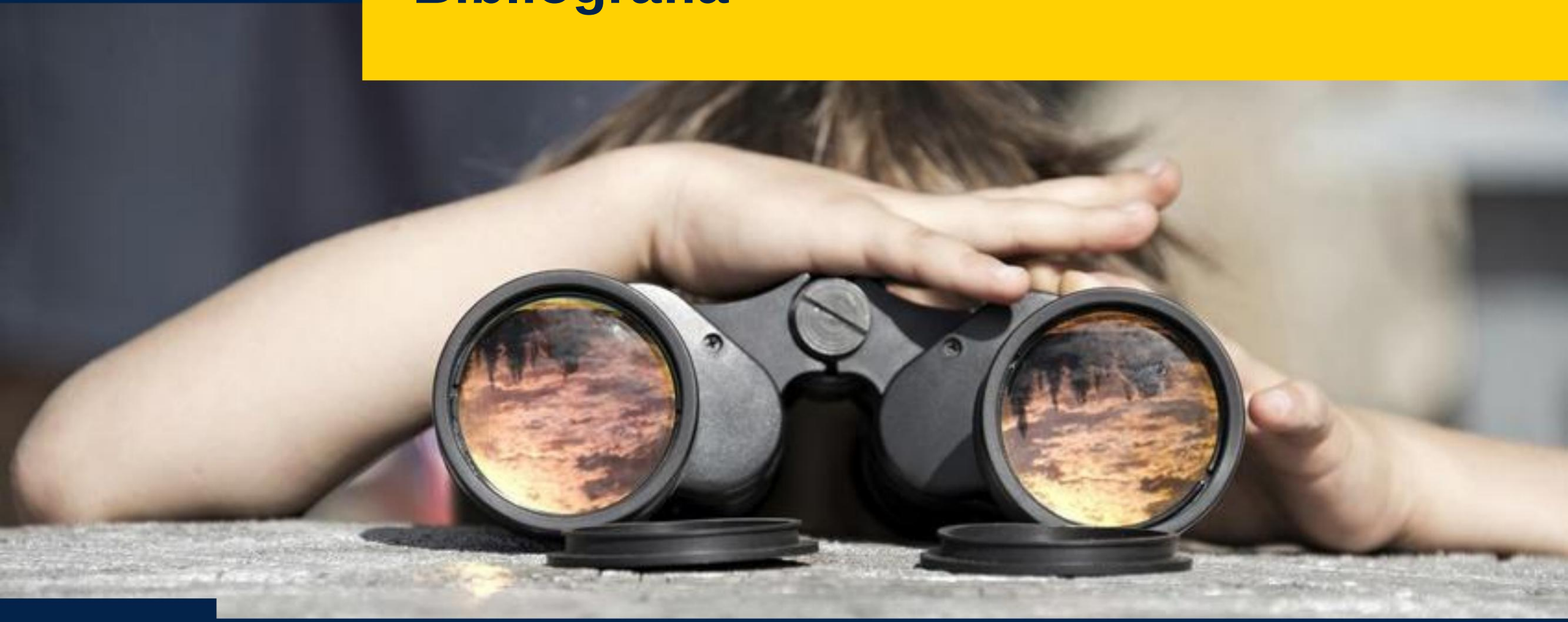
$$PFD_G = 2\left((1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\right)^2 t_{CE} t_{GE} + \beta_D \lambda_{DD} MTTR + \beta\lambda_{DU}\left(\frac{T_1}{2} + MRT\right)$$

Modalità continua/alta richiesta:

$$PFH_G = 2\left((1-\beta_D)\lambda_{DD} + (1-\beta)\lambda_{DU}\right)(1-\beta)\lambda_{DU} t_{CE} + \beta\lambda_{DU}$$

life.augmented

Bibliografia



Documenti di riferimento 1/2

[R1]: Affidabilità della microelettronica: modellazione basata sulla fisica dei guasti e valutazione della durata di vita - Jet Propulsion LaboratoryCalifornia Institute of Technology Pasadena, California

[R2]: : Manuale di affidabilità dei semiconduttori – Renesas Electronics, Rev.2.50 gennaio 2017

[R3]: ExoMars 2016 - Schiaparelli Anomaly Inquiry (ESA) scaricato da https://exploration.esa.int/web/mars/-/59176-exomars-2016-schiaparelli-anomaly-inquiry

[R4]: Manuale dell'albero dei guasti con applicazioni aerospaziali - Ufficio di sicurezza e garanzia della missione della NASA, V 1.1 2002 ,

[R5]: il software FTA aperto può essere trovato sul web, ad esempio https://www.fault-tree-analysis.com/free-fault-tree-analysis-software, oppure verifica il download di OpenFTA

Documenti di riferimento 2/2

[R6]: Manuale di sicurezza per TMS570LS31x e TMS570LS21x Hercules™ ARM®-Based Safety Microcontrollori critici

[R7]: Manuale di sicurezza della serie singlecore UM2331-STM32H7 STMicroelectronics – da https://www.st.com/en/embedded-software/x-cube-stl.html#documentation

[R8]: Manuale di sicurezza per l'unità di gestione dell'alimentazione (PMU) TPS65919-Q1

Grazie

life.augmented