

SIEMENS

Ingegno per la vita

SIEMENS
SIMATIC
S7-1500

RUN

Fail-safe

Safety mode:
Enabled

Collective signature:
5CBE6409

Last fail-safe modification:
04/30/2012 04:44:51 PM

ESC

Guida SIMATIC Safety Integrated

Edizione 2020

[siemens.it/automazione](https://www.siemens.it/automazione)

Le informazioni riportate in questo manuale tecnico contengono descrizioni o caratteristiche che potrebbero variare con l'evolversi dei prodotti e non essere sempre appropriate, nella forma descritta, per il caso applicativo concreto. Con riserva di modifiche tecniche.

Tutte le denominazioni dei prodotti possono essere marchi oppure denominazioni di prodotti della Siemens AG o di altre ditte fornitrici, il cui utilizzo da parte di terzi per propri scopi può violare il diritto dei proprietari.

Copyright © 2020. Siemens AG. All rights reserved.

Introduzione

Questo documento vuole essere una guida introduttiva all'utilizzo di PLC della serie S7-1200 / S7-1500 / ET200SP CPU con funzioni di sicurezza integrata, avvalendosi dei software TIA Portal: [Step 7 + Safety Advanced](#).

Dopo una presentazione generale dell'uso dei PLC Safety nell'automazione industriale seguirà un esempio, con guida passo-passo, per una configurazione di questo tipo.

Rimane comunque un valido strumento il manuale del Safety Advanced scaricabile dal sito di supporto all'ID n° 54110126.

E' possibile inoltre, scaricare dal sito di supporto Siemens, manuali che analizzano nel dettaglio funzionalità e parametri delle varie tipologie di schede safety; in particolare al seguente link

[www.siemens.com/global/en/home/products/automation/topic-areas/safety-integrated/factory-](http://www.siemens.com/global/en/home/products/automation/topic-areas/safety-integrated/factory-automation.html)

[automation.html](http://www.siemens.com/global/en/home/products/automation/topic-areas/safety-integrated/factory-automation.html) è possibile trovare una spiegazione dettagliata sull'utilizzo safety per la costruzione di macchine.

Per seguire senza difficoltà questo documento è vivamente consigliata la conoscenza dei PLC della serie Simatic S7-1200 ed S7-1500 e del software Simatic STEP7 (TIA).

Sommario

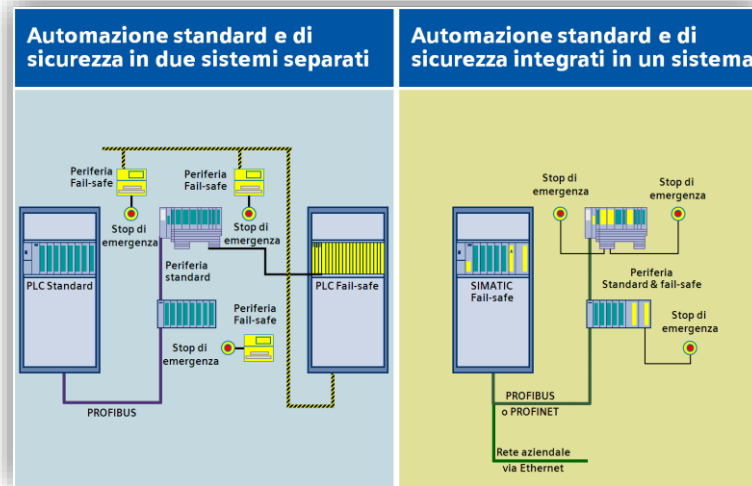
1.	Il concetto	5
1.1.	Requisiti HW e SW	5
1.2.	CPU Safety	8
1.3.	Esempi di configurazioni	10
2.	Guida pratica	11
2.2.	Configurazione hardware	11
2.2.1.	Parametri CPU F	13
2.2.2.	Safety Administration	13
2.2.3.	Parametri comuni a tutti i dispositivi Safety	16
2.2.4.	Parametri scheda F-DI 8x24VDC	17
2.2.5.	Parametri scheda 4 F-DO	20
2.2.6.	Power module Safety F-PM-E	23
2.4.	Software safety	25
2.4.1.	DB delle schede Safety	26
2.4.2.	Struttura programma Safety	27
2.4.3.	Scrittura del codice (KOP, FUP)	28
2.4.4.	Comando uscite Safety	29
2.4.5.	Controllo scrittura dati Safety	30
2.6.	Le funzioni Safety	31
2.6.1.	ESTOP1	31
2.6.2.	FDBACK	32
2.6.3.	ACK_GL	33
2.6.4.	TWO_HAND	34
2.6.5.	MUTING	34
2.6.6.	ACK_OP	34
2.6.7.	Contatori e temporizzatori Safety	35
2.6.8.	Gestione e conversione dati Safety	35
3.	Comunicazione safety	35
3.1.	Comunicazione safety tramite PROFISafe	35

3.1.1.	Esempio applicativo SENDDP & RCVDP in un progetto integrato	38
3.1.2.	Esempio applicativo SENDDP & RCVDP in un progetto NON integrato	41
3.1.3.	Esempio applicativo scambio dati safety tramite PN-PN coupler	41
3.2.	Comunicazione safety tramite TCP/IP – Flexible F-Link.....	47
3.2.1.	Flexible F-Link comunicazione F-CPU/F-CPU	48
3.2.2.	Flexible F-Link per F-Runtime Group communication	57
3.2.3.	Reintegrazione a seguito di errori di comunicazione	59

1. Il concetto

L'utilizzo del Safety Integrated permette di rendere più semplice, flessibile e integrata l'architettura della macchina dove è richiesta l'implementazione della sicurezza.

La gestione dell'automazione standard e della sicurezza diventa quindi possibile, utilizzando un solo PLC laddove prima erano richiesti 2 sistemi separati.

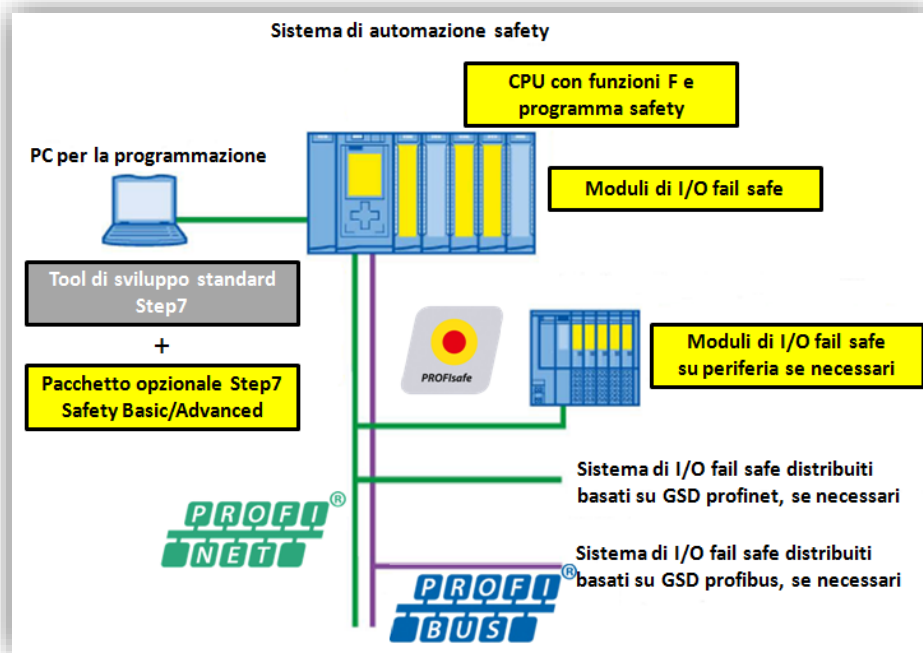


L'utilizzo di un'architettura integrata introduce **notevoli benefici e vantaggi** quali:

- **Stesso software** per lo sviluppo di tutte le funzionalità della macchina.
- **Stesso ambiente e linguaggio di programmazione** per la parte Standard e Safety (KOP o FUP).
- Il protocollo PROFISafe permette di utilizzare per il collegamento di dispositivi Safety tutti i bus di campo supportati dalle CPU Siemens: **Profibus, Profinet e anche Wireless su Profinet**.
- Il bus di campo standard **permette di integrare sullo stesso bus dispositivi Standard e Safety indifferentemente** (ad esempio una periferia ET200 può avere moduli sia Standard che Safety).
- In presenza di più PLC Safety sull'impianto è **possibile scambiare informazioni di sicurezza utilizzando semplici FB**.
- **Notevole flessibilità** in fase di progettazione e messa in servizio. Per esempio, se è necessario acquisire un ulteriore sensore di sicurezza, è sufficiente cablarlo alla periferia più vicina e inserire il suo contatto logico nel programma, all'interno della catena di sicurezza. Questa stessa operazione con dispositivi elettromeccanici potrebbe richiedere un aumento dei tempi di cablaggio.

1.1. Requisiti HW e SW

L'immagine seguente illustra i componenti necessari per la realizzazione di un "progetto Safety".






Per una programmazione Safety è necessario il software di programmazione STEP7 TIA Portal più il pacchetto Safety Basic (se si utilizza esclusivamente il PLC S7-1200) oppure il pacchetto Safety Advanced (in questo caso si potrà programmare l'intera gamma di CPU safety S7-1200 / S7-1500).

Entrambi i pacchetti Safety si integrano completamente nell'ambiente TIA e permettono:

- la realizzazione e la gestione della configurazione hardware con schede safety
- la realizzazione e la gestione del programma Safety con l'utilizzo della libreria Safety contenente una serie di funzioni realizzate e certificate (SIL3 – PLd) da Siemens, per svolgere le più comuni operazioni di sicurezza (gestione emergenze, bimanò, muting...)

La funzionalità Safety Integrated la possiamo trovare a partire dal PLC 1200F fino ad arrivare al PLC 1500F, passando naturalmente dall'ET200F CPU, di seguito tabella riassuntiva:

S7-1200	ET200-SP	S7-1500
		
S7-1214FC S7-1215FC S7-1212FC	S7-1510SP-F S7-1212SP F	S7-1511 F S7-1513 F S7-1515 F S7-1516 F S7-1517 F S7-1518 F

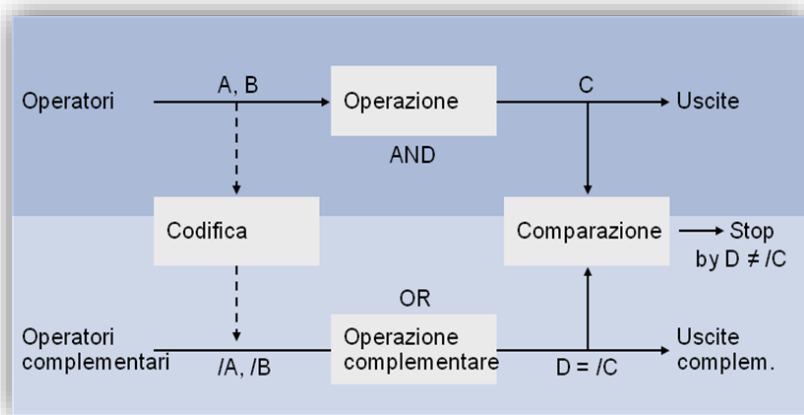
La molteplicità di schede di I/O Safety include anche schede IP67 per il montaggio diretto a bordo macchina.

PERIFERIA FAILSAFE	Profibus	Profinet	DI	DO	DI/DO	AI	Relay/Switch	Motor starter	Freq. Converter	Proprietà
ET 200MP	X	X	X	X	-	-	-	-	-	L'IO Modulare per applicazioni multicanale con 16 canali per modulo
ET 200SP	X	X	X	X	-	X	X	X	-	L'IO Modulare per applicazioni multicanale con fino a 8 canali per modulo
ET 200M	X	X	X	X	-	X	-	-	-	L'IO Modulare per applicazioni multicanale con 24 canali per modulo
ET 200S	X	X	X	X	X	-	X	X	X	L'IO Modulare per applicazioni multicanale con fino a 8 canali per modulo
ET 200pro	X	X	X	-	X	-	X	X	X	L'IO Modulare e multifunzionale ad alto grado di protezione IP 65/67
ET 200eco PN	-	X	-	-	X	-	-	-	-	Il blocco di apparecchiatura periferica ad alto grado di protezione IP 65/67

1.2. CPU Safety

I PLC che gestiscono le funzionalità Safety sono identificati dalla lettera F (Fail safe) dopo il numero identificativo del modello (es. S7-1511-F 1PN) e sono dotate di una memoria di lavoro e di dati più capiente, per poter ospitare anche la parte di codice sicuro.

L'esecuzione del programma safety avviene ad intervalli regolari (OB a tempo definito dall'utente). Per i blocchi safety che l'utente dichiara "F-block" (OB, FB, FC, DB e UDT) le CPU "F" in modo automatico e trasparente, in fase di compilazione, creano un programma complementare a quello scritto, in questo modo dopo averli eseguiti entrambi, controlla che il risultato sia complementare. In caso contrario, significa che la CPU "F" ha commesso un errore e quindi si porterà nello stato di STOP, ovvero lo stato '*sicuro*'. Di seguito tabella esplicativa:



Poiché questa procedura, in termini di tempo, impegna il processore della CPU, è necessario scrivere nei blocchi di programmazione relativi alla sicurezza, solo la parte di codice che effettivamente deve essere eseguito in sicurezza per non allungare inutilmente il tempo ciclo. Il resto del codice risiederà nei blocchi standard. Se si volesse avere un'indicazione di quanto potrebbe durare il tempo di esecuzione della parte safety in base alla CPU utilizzata, al numero di ingressi/uscite e al tipo di funzioni di sicurezza richieste dalla macchina, Siemens mette a disposizione un file excel da scaricare dal sito del support (ID del documento 93839056).

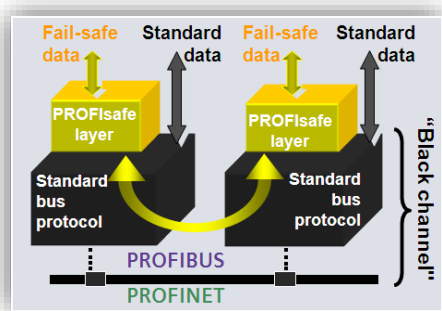
Le parti di programma standard e safety, possono scambiarsi dati utilizzando le aree di memoria a cui entrambi hanno accesso. Nella tabella seguente sono riportate le varie possibilità di accesso (lettura e/o scrittura) che i blocchi standard e safety hanno sulle aree di memoria del PLC.

Area di memoria	Blocchi STANDARD	Blocchi SAFETY
DB standard	Lettura/Scrittura	Lettura/Scrittura
<u>Merker</u>	Lettura/Scrittura	Lettura/Scrittura
<u>DB safety</u>	Lettura	Lettura/Scrittura
Input/Output Standard	Lettura/Scrittura	Lettura/Scrittura
<u>Input/Output Safety</u>	Lettura	Lettura/Scrittura

Quindi, tramite DB standard (o in alternativa l'area merker), è possibile riportare i risultati della logica standard al programma fail safe e utilizzarli per comandare delle uscite Safety in sicurezza. Viceversa, le DB-F che contengono le informazioni sullo stato della sicurezza della macchina, possono essere lette dai blocchi standard per accendere segnalazioni luminose o eseguire operazioni che non richiedono un livello di sicurezza.

Per questo motivo non è necessario spostare tutta la logica della macchina nel programma Safety anche se sono utilizzate uscite Safety per comandare gli attuatori.

Un'altra regola da tener presente nella stesura del codice safety è quella che obbliga i blocchi safety ad utilizzare i dati standard o solo in lettura o solo in scrittura, non entrambi (questo errore viene rilevato dal TIA Portal al momento della stesura del codice).



I tipi di dati utilizzabili nei blocchi di dati safety sono: BOOL, INT, WORD, TIME, DINT, DWORD e UDT-F.

Per quanto riguarda le schede I/O "F", analogamente alle schede standard, è possibile inserirle lato CPU oppure in periferia. In modo del tutto trasparente le CPU-F e le schede I/O F si scambieranno tra di loro una serie di dati aggiuntivi, tramite un livello supplementare del protocollo standard chiamato PROFISafe, che include controlli temporali e CRC per verificare possibili errori come contraffazione di indirizzo, perdita dei dati, etc.

1.3. Esempi di configurazioni

Per le informazioni viste fino ad ora risulta chiaro che la soluzione ottimale in caso di configurazione per una nuova macchina è quella di scegliere un'unica CPU F, dimensionata opportunamente, in grado di elaborare tutto il programma macchina, standard e Safety.

Come riportato nell'immagine a lato, sarà possibile collegare alla CPU dei dispositivi solo Standard, solo Safety o Standard e Safety, riuscendo a gestire in modo ottimale la flessibilità messa a disposizione dal sistema.

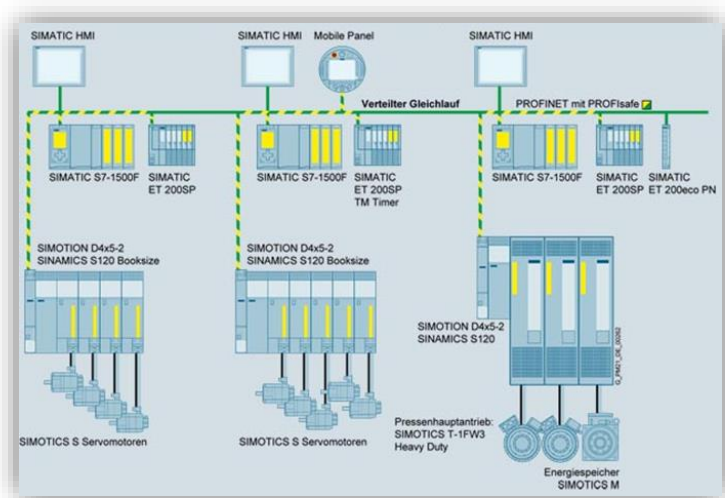
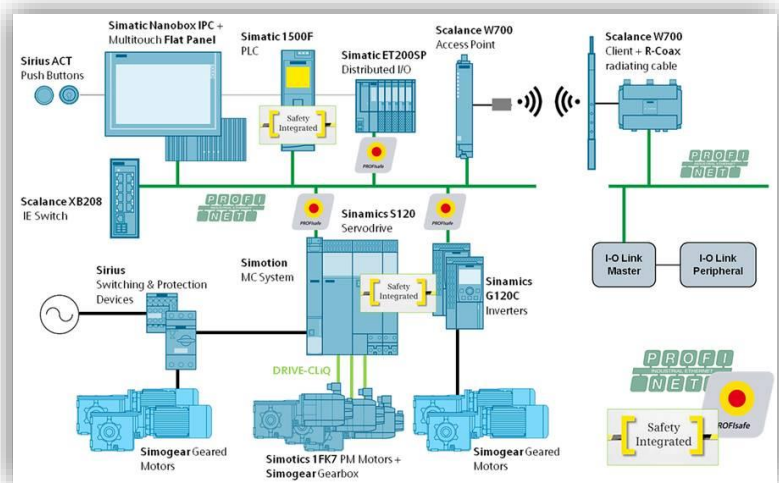
In caso di necessità di utilizzo di 2 CPU è possibile scegliere PLC F di taglia inferiore per svolgere solo la parte Safety, senza perdere i vantaggi dell'integrazione.

È possibile infatti usare un solo BUS di campo in quanto sia Profibus sia Profinet supportano una configurazione multi-master, consentendo di collegare sullo stesso bus fisico periferie che saranno gestite da una o l'altra CPU, senza dover stendere 2 linee differenti.

L'ultimo esempio di configurazione risulta molto utile in caso di grandi impianti o macchine modulari.

Questa soluzione prevede la gestione della sicurezza decentralizzata con una CPU per ogni zona. Tutte le CPU potranno scambiarsi dati di sicurezza, oppure comunicare informazioni sicure ad una CPU-F che faccia da concentratore della sicurezza della macchina/impianto.

La configurazione è praticabile con tutte le CPU Siemens. Le più utilizzate per questo tipo di architettura sono le CPU ET200SP o S7-1200, di piccola taglia e costo limitato, ma con la piena potenzialità per gestire programma Safety e Standard.



2. Guida pratica

Ora verranno illustrati i passi per realizzare un progetto Step7 con CPU-F e schede I/O Safety.

L'architettura scelta per l'esempio è costituita da:

- **CPU S7-1511F 1PN:** una CPU della famiglia S7-1500, dotata di buone prestazioni e costo ridotto, ottimale per piccole macchine o per la gestione locale di parti di impianto.
- **IM 155-6 PN STANDARD:** interfaccia standard di periferia ET200SP con moduli Fail safe e profilo Profisafe. (ET200S necessita un'interfaccia HF).
- **Schede di I/O ET200SP:**
 - **PM-E DC24V**
 - **DI 16x24VDC ST**
 - **DO 16x24VDC/0.5A**
 - **F-DI 8x24VDC HF**
 - **F-DO 4x24VDC/2A PM HF**

Le schede fail safe attuali non richiedono una separazione dalle schede standard (in questo modo le schede di I/O possono essere inserite nell'architettura hardware in modo misto, raggiungendo ugualmente il livello di sicurezza SIL3-Cat.4/Plc).

Questo rende il sistema particolarmente flessibile soprattutto quando si ha la necessità di modificare l'architettura dell'impianto.

IMPORTANTE: Un progetto Step 7 con CPU Safety richiede l'utilizzo di 2 password:

- **Password ONLINE:** la password da configurare sul PLC, sarà richiesta ogni volta che si voglia trasferire la configurazione hardware o il programma. È possibile associarla solo al programma Fail safe oppure sia a Fail safe che standard. Su S7-1500 e S7-1200 è possibile definire password diverse per accedere alla parte standard e non alla parte fail safe.
- **Password OFFLINE:** sarà richiesta per eseguire modifiche sia per quanto riguarda la programmazione Safety sia per quanto riguarda la configurazione hardware, nel progetto offline.

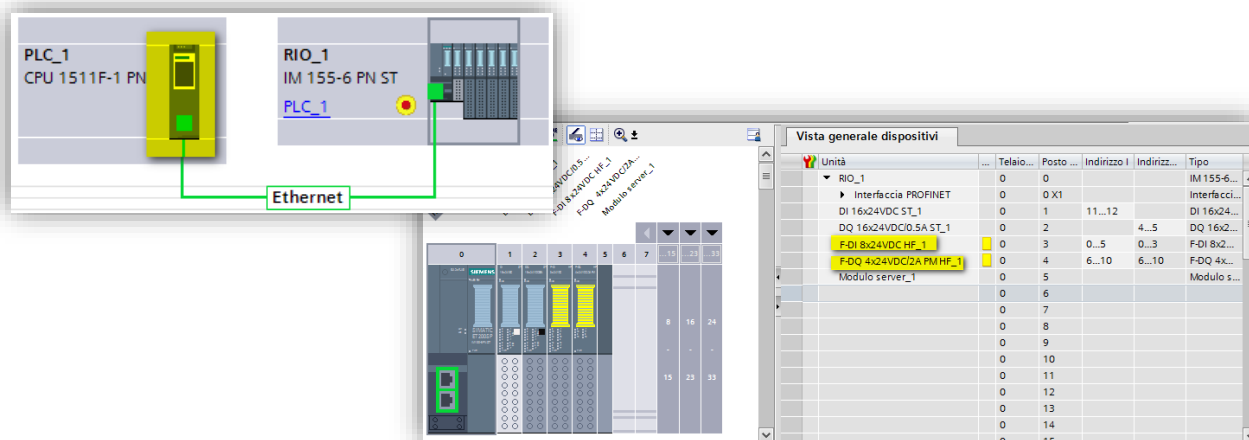
NOTA: Le password di default sono disattivate! Questo per rendere più agevole la realizzazione del progetto nella fase iniziale.

È consigliato attivare entrambe le password nel momento in cui il PLC viene consegnato all'utilizzatore finale.

2.2. Configurazione hardware

La configurazione hardware di un progetto safety viene gestita in modo analogo a un progetto standard: nello specifico i dispositivi safety vengono configurati esattamente come quelli standard con la differenza che avranno alcuni parametri specifici che li caratterizzano.

In questo capitolo verranno trattati nel dettaglio questi parametri e trascurati i passi base abitualmente richiesti da Step7 per realizzare la configurazione HW.

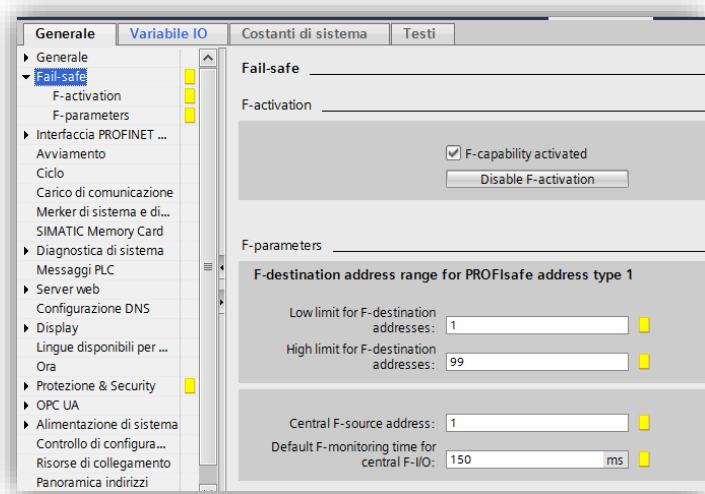


Nelle immagini qui sopra è illustrata la configurazione dell'architettura scelta per questa guida.

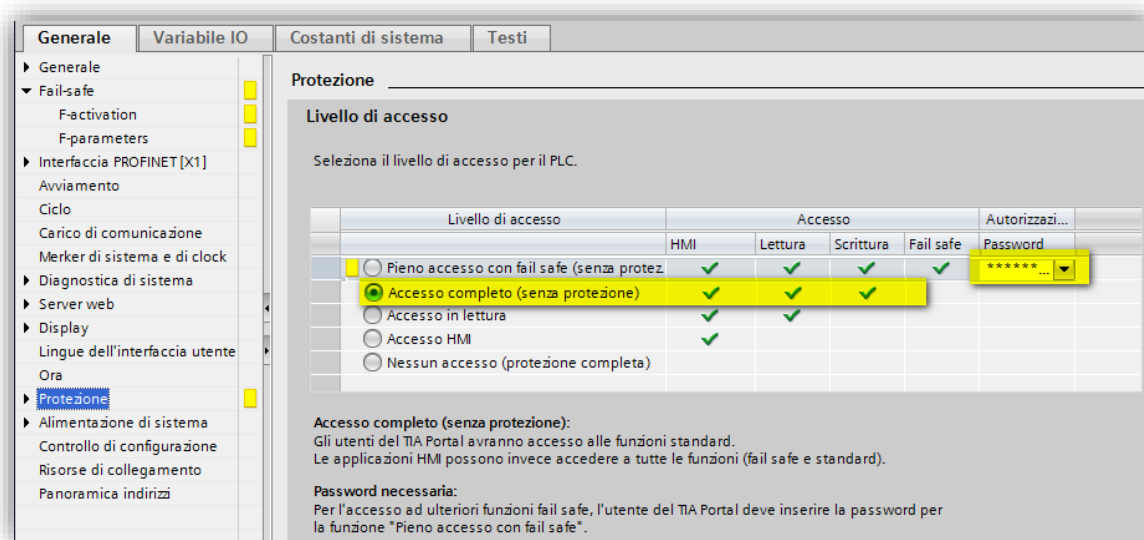
In giallo sono evidenziati i dispositivi Safety presenti: la CPU e le 2 schede I/O F.

2.2.1. Parametri CPU F

Una CPU di sicurezza, oltre ai normali parametri di configurazione che si trovano su tutte le CPU, ha in aggiunta 2 tendine di configurazione.



Dalla tendina fail safe è possibile scegliere se mantenere attiva (come default) o disattivare la funzione di Safety integrated della CPU ed impostare i parametri come l'indirizzo PROFISafe della CPU e il "monitoring time" per gli I/O centralizzati (vedere cap. 2.1.2 per il significato).

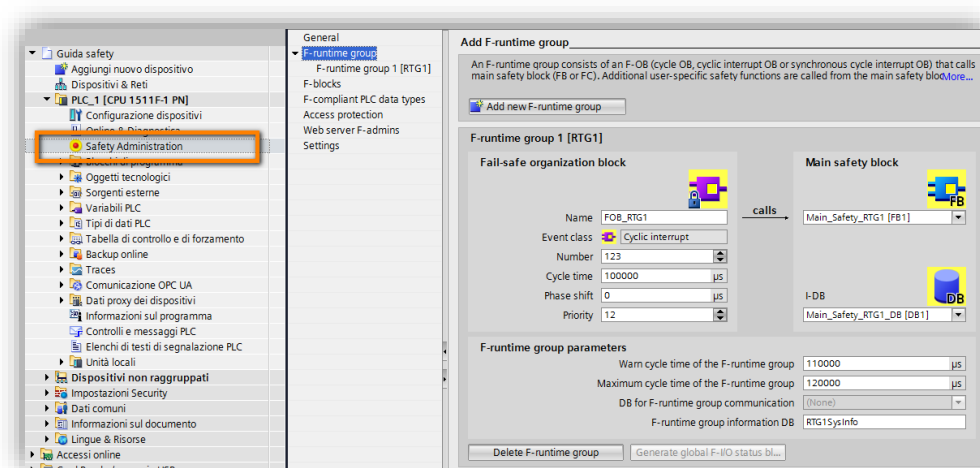


Dalla tendina **Protezione** è invece possibile attivare quella precedentemente definita come "password online": spuntando "Accesso completo (senza protezione)" si associa la password solo al programma di sicurezza, con le altre scelte si protegge in "scrittura" o "scrittura/lettura" anche il programma standard.

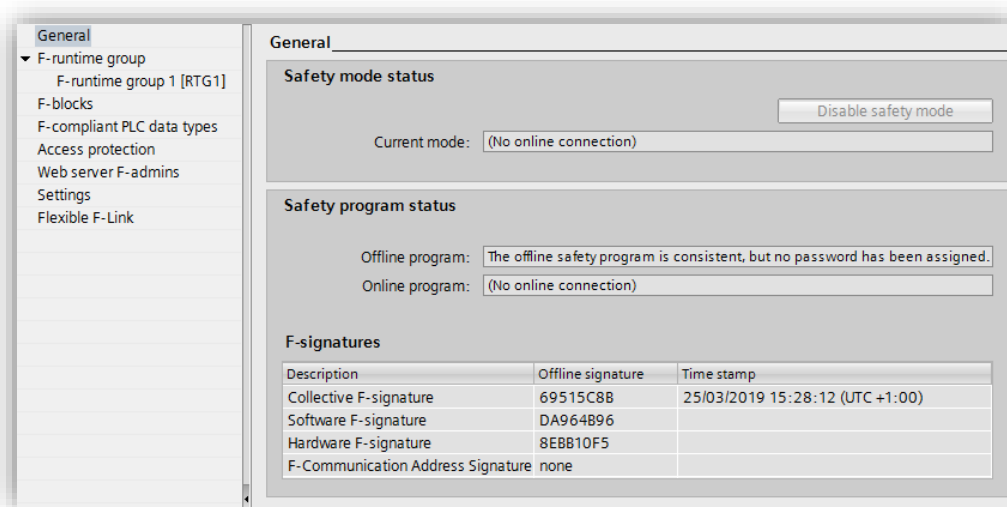
2.2.2. Safety Administration

L'editor Safety Administration contiene diverse informazioni riguardanti specifici parametri della parte safety del progetto. Questa sezione è molto importante in quanto, per la validazione della sicurezza della macchina, bisogna necessariamente allegare la documentazione che viene generata tramite questo editor.

In **Safety Administration** vi sono diversi menu:



- In **General**, come mostrato dalla figura seguente, sono presenti gli F-signature con i relativi time stamp. Tali codici alfanumerici vengono generati automaticamente dal software e rappresentano in modo univoco cambiamenti nella logica safety. In particolare, il *Collective F-signature* è una identificazione collettiva (generale), *Software F-signature* è un codice alfanumerico che identifica univocamente la sola parte software safety, *Hardware F-signature* è un codice alfanumerico che identifica univocamente le proprietà safety della parte hardware mentre *F-Communication Address Signature* è un codice alfanumerico che identifica la comunicazione Flexible F-Link di cui si parlerà in maniera dettagliata in questa guida. Una volta compilato il progetto in tutte le sue parti:
 - *Software F-signature* cambierà ogni qualvolta si modificherà la logica di funzionamento della parte software safety,
 - *Hardware F-signature* cambierà ad ogni modifica delle sole proprietà safety dell'hardware,
 - *F-Communication Address Signature* cambierà ad ogni modifica delle impostazioni della comunicazione Flexible F-Link,
 conseguentemente, ad ogni cambio di Software/Hardware F-signature, cambierà automaticamente anche il Collective F-signature.

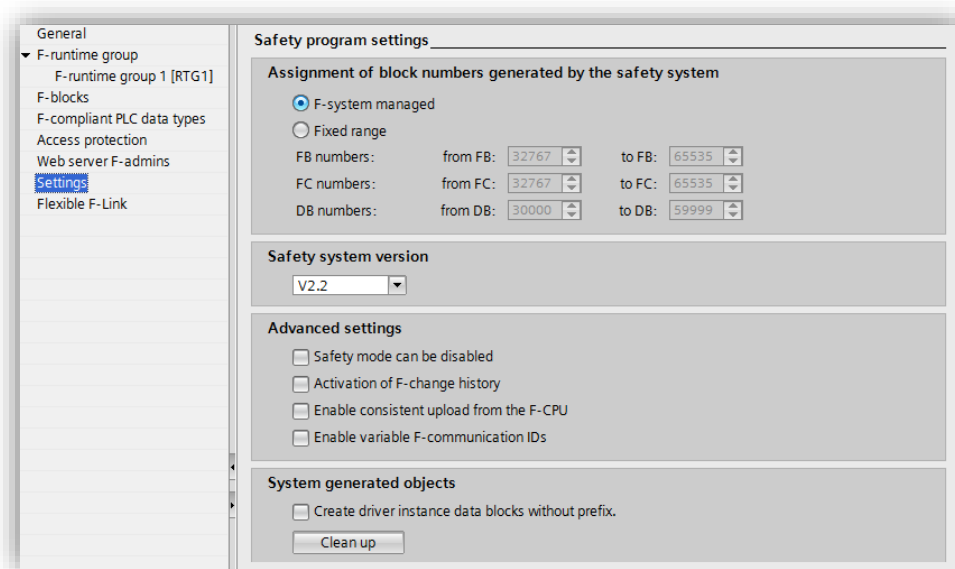


- In **F-runtime** group è possibile modificare la funzione che rappresenta il main safety e in quale OB a tempo richiamarla. È possibile inoltre impostare i tempi di ciclo massimo e la priorità dell'OB che richiama il programma safety
- In **Access Protection** è possibile attivare la password offline, per proteggere il programma Safety
- In **Settings** è possibile cambiare il range di numeri in cui la compilazione crea in automatico DB, FB e FC safety.

Se ad esempio le numerazioni scelte in automatico per la CPU ricadono in un'area utilizzata nel programma utente è possibile modificarle; nella scelta dei nuovi numeri è fondamentale mantenere l'ampiezza dell'intervallo almeno pari a quello presente in partenza.

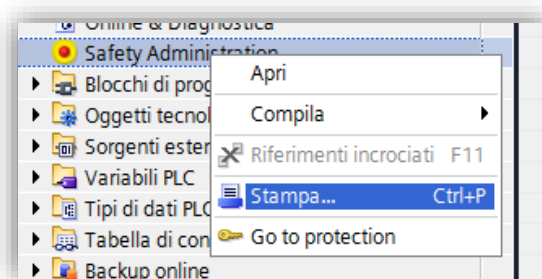
All'interno delle aree definite non è possibile creare alcun DB o FB utente, né standard né Safety.

Inoltre, sotto la voce **Advanced settings** abilitando la proprietà *Safety mode can be disabled* si ha la possibilità di modificare lo stato delle variabili presenti nel software safety senza che la parte di sicurezza ne blocchi il controllo; abilitando *Enabled consistent upload from the F-CPU* (solo per S7-1500) è possibile fare l'upload del software completamente (parte standard e parte safet



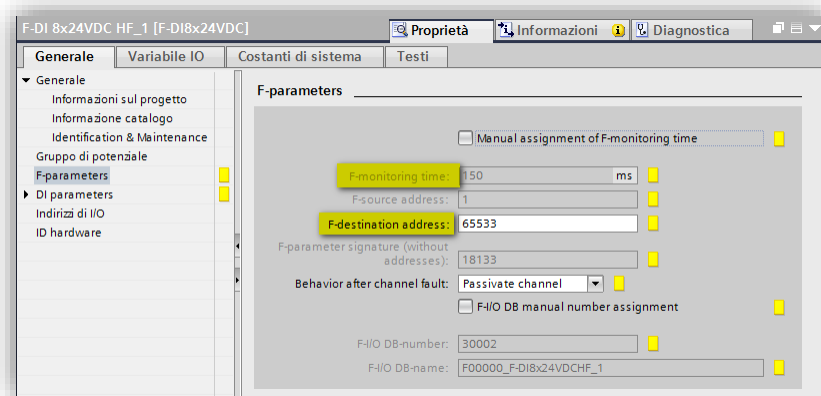
- In **Flexible F-Link** è possibile andare a configurare la parametrizzazione della comunicazione F-Link (per maggiori dettagli vedere il capitolo 3 di questa guida).

Per effettuare la stampa del documento relativo all'editor di Safety Administration, come mostrato in figura cliccare con il tasto destro del mouse su Safety Administration e dal menu scegliere *stampa*.



2.2.3. Parametri comuni a tutti i dispositivi Safety

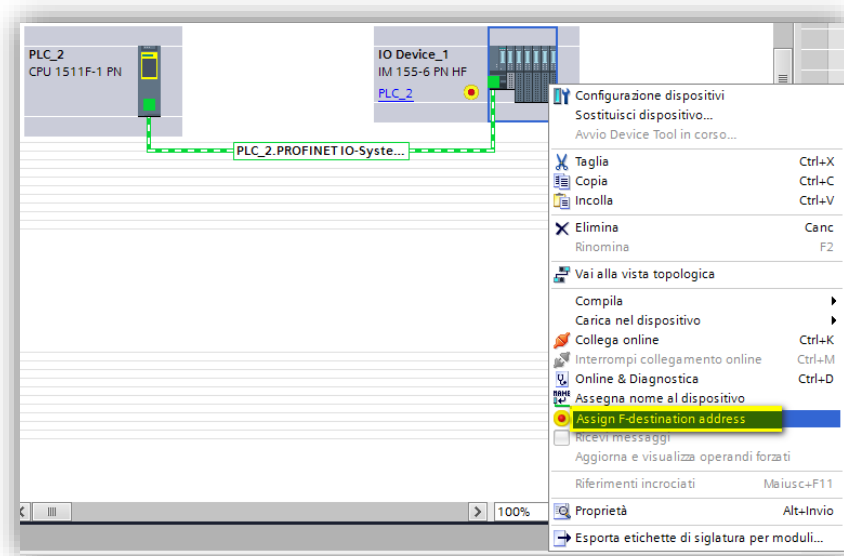
Qualsiasi dispositivo safety collegato alla CPU, che sia una scheda I/O, un drive con funzioni safety piuttosto che un dispositivo Profibus/Profinet (ad esempio un encoder safety), richiede la configurazione di 2 parametri comuni: "F-destination_address" e "F_monitoring_time".



F-destination_address è un indirizzo univoco dei dispositivi safety. Step7 lo assegna automaticamente, ma l'indirizzo può essere cambiato a piacimento. Viene verificata automaticamente dal software l'univocità dell'indirizzo assegnato.

Questo numero andrà poi riportato sui dispositivi fisici, con metodi diversi in base al tipo di dispositivo:

- Se si utilizzano schede ET200SP, è sufficiente, dopo aver caricato il software nel PLC ed aver assegnato il nome del dispositivo profinet alla stazione, assegnare il profisafe address tramite il TIA Portal cliccando sul relativo tasto nella vista di rete:



Tale procedura di assegnazione dell'F-destination address su moduli di periferia ET200SP è possibile effettuarla anche usando blocchi di programma software, per ulteriori informazioni consultare la guida presente sul sito del support al seguente ID 109748466;

- se si utilizzano schede ET200S o ET200M, devono essere utilizzati dip-switch presenti sul retro delle stesse (bisognerà copiare la conversione binaria del numero presente nella finestra di parametrizzazione);
- per i moduli di I/O-F di S7-1200 gli F-Destination address vengono assegnati automaticamente dal software e non necessitano nessuna procedura particolare di assegnazione;
- nel caso di dispositivi programmabili dovrà essere copiato nel tool di configurazione dedicato.

Questo indirizzo non è da confondere con l'indirizzo logico usato nel programma PLC per comunicare con i dispositivi, è solo un indirizzo identificativo dei dispositivi safety che, una volta impostato correttamente, non sarà necessario utilizzare in programmazione.

F-monitoring time è un tempo di controllo, che assume la nomenclatura watch-dog time su alcuni dispositivi. Questo tempo verrà usato sia dalla CPU che dal dispositivo per **verificare che non ci siano problemi di connessione**.

Prendendo come esempio la scheda F-DI: la CPU controllerà che tra due telegrammi consecutivi ricevuti dalla scheda non passi un tempo superiore a 150ms (tempo impostato di default sulle schede). Se questo tempo viene superato la CPU lo considererà come un possibile problema e porrà automaticamente la scheda in passivazione (tutti gli ingressi impostati a 0).

Per quanto riguarda la scheda di F-DO, in caso di superamento di questo tempo la scheda di uscita stessa deve autonomamente portare lo stato dei suoi canali a 0, cioè in condizione di sicurezza.

La corretta configurazione di questo tempo richiede la valutazione del tempo di richiamo del programma safety (tempo OB35), del tempo ciclo del programma safety e del tempo di aggiornamento del nodo sul bus nel caso di dispositivo decentrato.

Siemens mette a disposizione per il calcolo del minimo monitoring time impostabile e del tempo ciclo del programma safety, un file Excel che è possibile scaricare all'ID [58856512](#) del sito di supporto.

Questo tempo non influisce sul calcolo del tempo di reazione del sistema ad una richiesta di arresto di emergenza, è solo un controllo sulla comunicazione.

Il tempo assegnato di default è quello configurato in corrispondenza dell'interfaccia a cui è collegata la periferia. Cambiando il tempo sull'interfaccia viene cambiato in automatico il tempo di tutte le periferie collegate sotto quella rete. Questo tempo potrà comunque essere parametrizzato individualmente scheda per scheda.

2.2.4. Parametri scheda F-DI 8x24VDC

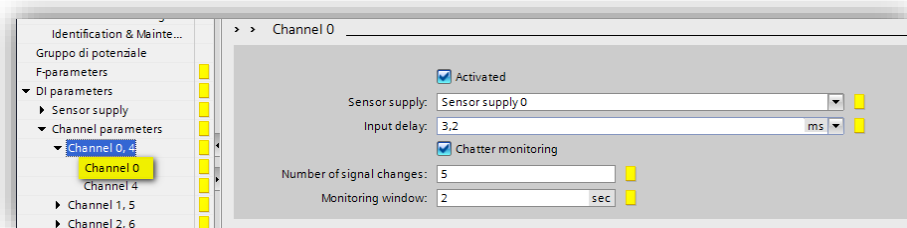
Il parametro **Behavior after channel faults** è comune a tutti i canali (presente nella tendina F-parameter vista sopra). Con questo parametro si definisce il comportamento della scheda all'incorrere di un errore su un canale. I canali safety in presenza di un errore vengono messi in uno stato di passivazione (cioè considerati non più validi), il PLC leggerà lo stato sicuro, ovvero il valore 0. Lo stato di passivazione inizia quando si verifica l'errore, continua anche quando l'errore viene corretto e si ripristina solo successivamente ad un acknowledgement della scheda.

Le possibili configurazioni di questo parametro sono: passivazione dell'intero modulo oppure passivazione del singolo canale su cui è stato riscontrato l'errore (solo su moduli fail safe ET200SP è possibile fare questa scelta).

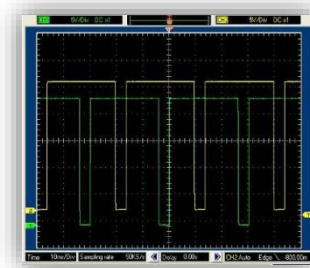
“Passivate entire module” al verificarsi di fault su di un qualsiasi canale, l'intero modulo viene passivato e tutti i canali (anche quelli privi di errore) vengono portati al loro stato sicuro.

“Passivate of the channel” al verificarsi di un fault viene passivato solo il canale interessato. Grazie a questo si hanno una serie di vantaggi: consente di continuare a leggere lo stato dei canali non in errore e permette di avere la diagnostica del canale dove si è generato il fault in modo automatico sul DB associato alla scheda (vedi paragrafo 2.2.1).

Per ogni canale abbiamo poi i seguenti parametri:



- **Sensor supply:** Possiamo scegliere se il canale in questione utilizza alimentazione interna od alimentazione esterna. Se si utilizza alimentazione interna sarà possibile abilitare anche i test di corto-circuito.
- **Input delay:** filtro sull'ingresso
- **Chatter monitoring:** possibilità di rilevare lo 'sfarfallio' degli ingressi. Se un ingresso cambia di stato più volte (*Number of signal changes*) entro la '*Monitoring window*' impostata, la scheda mette in passivazione l'ingresso.



Per ogni canale di alimentazione interna, abbiamo ulteriori proprietà:

- **Short-circuit test:** come detto, il modulo può verificare la presenza di corto-circuiti su singoli canali. Abilitando questo parametro, l'alimentazione a 24V sulle coppie di canali è con un impulso di trigger sfasato tra i canali per permettere il test di cortocircuito (immagine a lato). Senza questa abilitazione l'alimentazione è fornita tramite un 24V fissa
- **Time for short-circuit test:** se è stato richiesto il test di corto-circuito, è possibile con questo parametro, stabilire quanto deve durare l'impulso di trigger della relativa alimentazione.

Gli altri parametri a disposizione sono relativi ad ogni singola coppia di canali.

Essendo richiesto il doppio canale per raggiungere i livelli più alti della sicurezza, è previsto che la parametrizzazione avvenga per coppia di canale, lasciando così piena flessibilità al programmatore: nello specifico sulla stessa scheda è quindi possibile collegare sensori a singolo o doppio canale, parametrizzandoli coppia per coppia.

Caratteristica importante delle schede safety Siemens è che la gestione e il controllo dei canali viene fatto dall'hardware.

Sia che il sensore sia gestito a singolo canale o a doppio canale, il controllo di cortocircuito o di discrepanza tra i canali è demandato alla scheda, per questo motivo la stessa invierà un bit che verrà interpretato dal PLC: quando questo presenta lo stato 1 il sensore non è interessato e non c'è nessun errore, quando il bit è a 0 il sensore è stato interessato oppure si è verificato un errore. Per discriminare la causa dello stato di 0 è

necessario verificare lo stato del bit di diagnostica (*value status*) specifico del canale, che troviamo direttamente nell'immagine di processo di ingressi della scheda stessa:

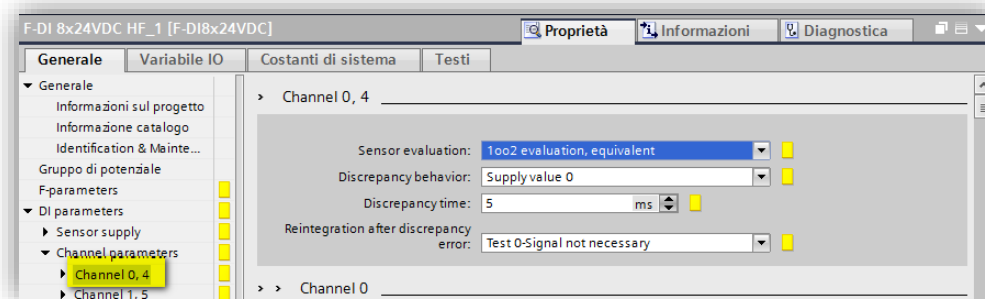
Byte input della CPU	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x+0	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
x+1	Stato diagnostica canale 7	Stato diagnostica canale 6	Stato diagnostica canale 5	Stato diagnostica canale 4	Stato diagnostica canale 3	Stato diagnostica canale 2	Stato diagnostica canale 1	Stato diagnostica canale 0

Se il bit dello stato di diagnostica (*value status*) è pari a **true**, il canale non ha errori altrimenti significa che la scheda ha rilevato sul canale un problema hardware ed ha quindi passivato il relativo ingresso. È poi disponibile una diagnostica generale della singola scheda che verrà approfondita nel *capitolo 2.2.1*. La diagnostica di singolo canale è attiva solo se, per la scheda in questione, è stata impostata la passivazione del singolo canale.

Per comodità di scrittura del codice i canali sono accoppiati 0-4, 1-5, 2-6, 3-7; questo fa sì che in un utilizzo di tutti i canali doppi non sia necessario tenere conto dell'offset dei secondi canali (che non bisogna interrogare in quanto rimarranno sempre a zero) ma si possano utilizzare direttamente i primi 4 bit della scheda 0-1-2-3.

Nel momento in cui viene selezionata la coppia per la lettura di due sensori a singolo canale non è più possibile parametrizzare il tempo di discrepanza e la reazione al suo superamento in quanto non è necessario.

Di seguito sono analizzati i parametri per configurare un sensore a doppio canale che richiede una parametrizzazione più completa, rispetto alla parametrizzazione di un sensore a singolo canale

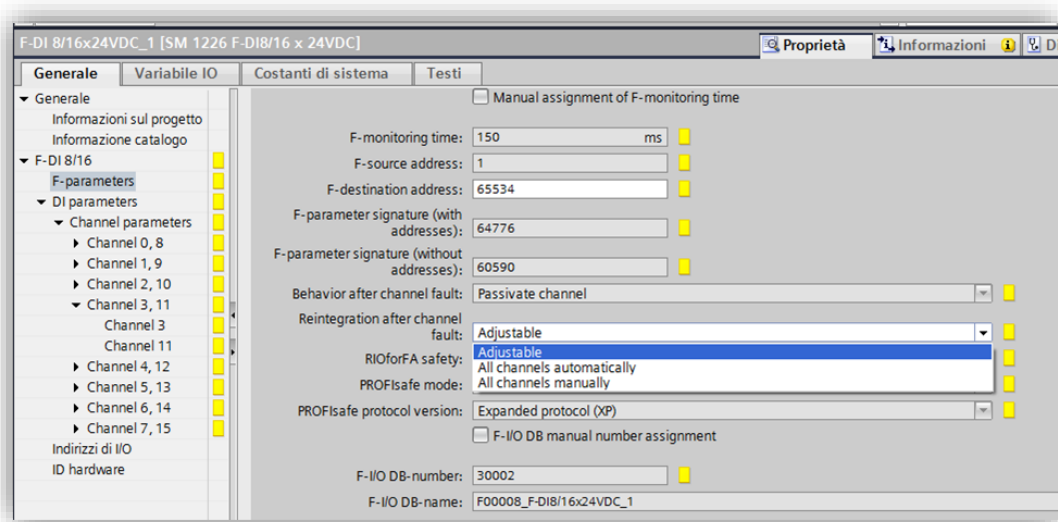


- **Sensor evaluation:** con questo parametro si seleziona se il sensore è singolo canale (1oo1 evaluation) oppure doppio canale (1oo2 evaluation). Come accennato prima, scegliendo 1oo1 si disabilitano i parametri successivi. È anche necessario decidere, in caso di doppio canale, se i due canali sono equivalenti (entrambi NC), oppure non-equivalenti (un canale NO e uno NC).
- **Discrepancy behavior:** con questo parametro è possibile impostare una reazione immediata alla variazione di uno dei due canali, importante se il tempo di discrepanza è particolarmente alto. Per esempio, se il sensore in questione ha un tempo di discrepanza alto e deve scatenare un arresto di emergenza, si può fare in modo che all'insorgere della discrepanza venga immediatamente segnalato "zero" alla CPU, evitando di aspettare che anche l'altro canale reagisca oppure che venga superato il tempo di discrepanza. Più precisamente le scelte a disposizione sono:
 1. Supply last valid value: prima di cambiare il valore del bit associato al sensore si attende che entrambi i canali assumano lo stesso valore oppure che il tempo di discrepanza venga superato

2. Supply value 0: appena uno dei due canali cambia stato, il segnale che viene inviato al PLC passa subito allo stato di zero
- **Discrepancy time**: qui si imposta il valore in ms del tempo massimo di discrepanza, cioè il tempo massimo in cui i due canali possono avere un valore differente, al superamento del quale il sensore viene considerato guasto. Questo tempo è una caratteristica del sensore. Il parametro successivo è legato a questo tempo e all'errore che ne consegue.
 - **Reintegration after discrepancy error**: quando l'errore di discrepanza viene corretto la scheda richiede un acknowledge per uscire dallo stato di passivazione, con questo parametro è possibile forzare un test di zero prima che sia possibile reintegrare l'errore. Impostando "Test 0-signal necessary" l'operatore deve portare il sensore in posizione di zero e poi di nuovo a uno. Solo dopo questa operazione la scheda richiederà l'acknowledge.

N.B. per i moduli di I/O fail safe su S7-1200 ed ET200MP esiste un'ulteriore parametrizzazione. Come mostrato in figura:

- **Reintegration after channel fault**: dopo aver corretto l'errore verificatosi sul canale è possibile scegliere come uscire dallo stato di passivazione; le scelte sono:
 - *Adjustable*: per ogni singolo canale è possibile configurare, direttamente nella parametrizzazione dello stesso, il modo di reintegrazione
 - *All channel automatically*: ogni canale viene reintegrato automaticamente
 - *All channel manually*: ogni canale deve essere reintegrato manualmente.

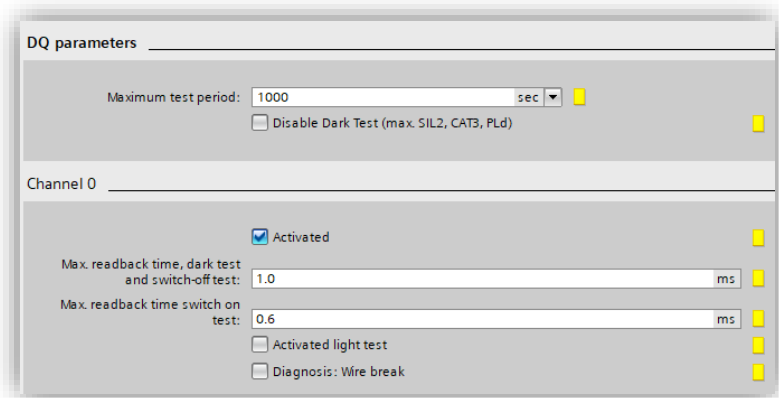


N.B. per poter testare lo stato di funzionamento degli ingressi della scheda safety all'interno di una **tabella di controllo** è necessario richiamare all'interno del software safety almeno uno degli ingressi della scheda.

2.2.5. Parametri scheda 4 F-DO

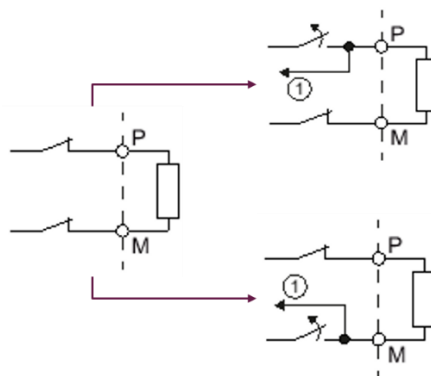
La scheda di uscita Safety presenta lo stesso parametro della scheda di ingresso per quanto riguarda la gestione della passivazione: è possibile scegliere di passivare solo il canale che è andato in errore oppure

tutta la scheda. Resta valido il discorso della diagnostica di canale: impostando la passivazione del canale è possibile leggere direttamente dal DB della scheda dove è avvenuto il fault (vedi paragrafo 2.2.1).

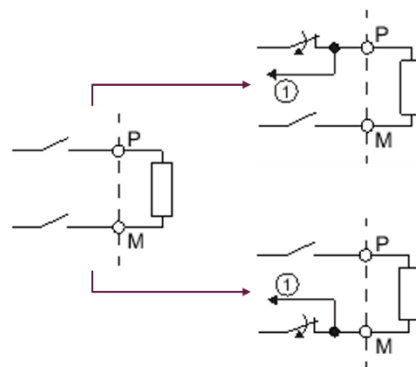


I parametri specifici di canale sono:

- **Activated:** per quanto riguarda i canali di uscita è necessario disattivare quelli non collegati ad un carico per evitare segnalazioni di errore
- **Disable dark test:** è un test che viene fatto in automatico dalla scheda e, tramite l'apertura alternata (verso P o verso M) del circuito quando il canale è attivo ("1"), rileva corto circuiti tra canale e L+ o canale ed M, cortocircuiti trasversali tra canali, guasti interni P/M switches. Su alcuni moduli non è disattivabile, disattivabile per 48h oppure disattivabile del tutto

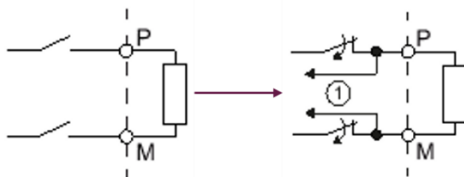


- **Switch on test:** ha lo stesso funzionamento del dark test solo che viene effettuato quando il canale non è attivo ("0"). A differenza del dark test, switch on test non è disattivabile su nessun modulo



- **Diagnostic wire break:** questo parametro attiva il controllo di rottura filo

- **Activated light test:** è un test opzionale e, tramite l'apertura simultanea del circuito su P ed M a canale non attivo ("0"), rileva il corto circuito con segnale a "0" e la rottura filo se abilitato anche il diagnostic wire break



- **Max readback time dark test/switch on test:** con questo parametro si configura la durata dell'impulso di test che la scheda genera di default ogni 1000 secondi, quando il segnale di uscita è a 1/0. Questo test serve per testare la corretta apertura dei due contatti interni alla scheda. Il tempo minimo impostabile corrisponde a 600µs, un tempo che difficilmente porta alla reazione di un dispositivo elettromeccanico. Se il canale è collegato ad un ingresso digitale di un dispositivo sicuro (per esempio un azionamento) e non è possibile impostare sullo stesso un filtro superiore a 600µs, diventa necessario interporre un relè tra canale d'uscita e dispositivo da comandare (nel caso di dark test).

Il tempo è da aumentare in caso di carico altamente capacitivo per evitare che la scheda, non rilevando variazioni di corrente, segnali un errore che in realtà non c'è (nel caso di dark test).

Attenzione: questo modulo esegue un taglio sicuro "PM" cioè esegue un taglio sul positivo (P) e uno sul negativo (M), per cui affinché il comando degli attuatori sia ridondato e la funzione di diagnostica attiva, è necessario che vengano collegati alla scheda con entrambi i contatti P e M, quindi gli attuatori devono essere isolati da terra e non è possibile mettere in comune i negativi.

Anche su questa scheda sono presenti dei bit di diagnostica specifici dei vari canali che troviamo direttamente nell'immagine di processo di ingressi della scheda stessa:

Byte input della CPU	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x+0	-	-	-	-	Stato diagnostica canale DQ3	Stato diagnostica canale DQ2	Stato diagnostica canale DQ1	Stato diagnostica canale DQ0

Se il bit dello stato di diagnostica (*value status*) è pari a **true** il canale non ha errori, altrimenti significa che la scheda ha rilevato sul canale un problema hardware ed ha quindi passivato la relativa uscita. È poi disponibile una diagnostica generale della singola scheda che verrà approfondita nel *capitolo 2.2.1*. La diagnostica di singolo canale è attiva solo se, per la scheda in questione, è stata impostata la passivazione del singolo canale.

N.B. per poter testare lo stato di funzionamento delle uscite della scheda safety all'interno di una tabella di controllo è necessario richiamare all'interno del software safety almeno uno delle uscite della scheda.

2.2.6. Power module Safety F-PM-E

Il power module safety dell'ET200SP, oltre ad avere a bordo una coppia di ingressi e un'uscita fail safe (che hanno le stesse proprietà dei moduli visti fino ad ora), in quanto power module, questa scheda ha un funzionamento che merita di essere approfondito.

Il power module F-PM-E, ha la funzionalità di avviare un gruppo di potenziale e di conseguenza alimentare tutte le schede collegate a valle, sino ad un nuovo gruppo di potenziale. L'alimentazione di tali schede a valle viene controllata direttamente dal power module grazie alla coppia di relè al suo interno, comandabili o con un bit sull'immagine di processo oppure dalla combinazione del sensore cablato sull'F-DI a bordo e dal bit sull'immagine di processo.

Essendo un modulo safety, il bit d'uscita che comanda i relè deve essere comandato dal programma sicuro, secondo la logica di sicurezza.

Comandano questo bit, vengono chiusi i relè e alimentate le schede. Portando il bit a 0, cioè in condizione di sicurezza, vengono aperti i relè e disalimentate le schede.

Per esempio: laddove ci sia la necessità di comandare in modo sicuro un numero elevato di elettrovalvole, invece di collegarle una ad una a singole uscite di sicurezza è possibile utilizzare moduli d'uscita standard preceduti da un F-PM-E. Così facendo, invece di tagliare tutte le singole uscite sicure, viene tagliata l'alimentazione cumulativa delle schede standard.

I limiti di una soluzione di questo tipo sono:

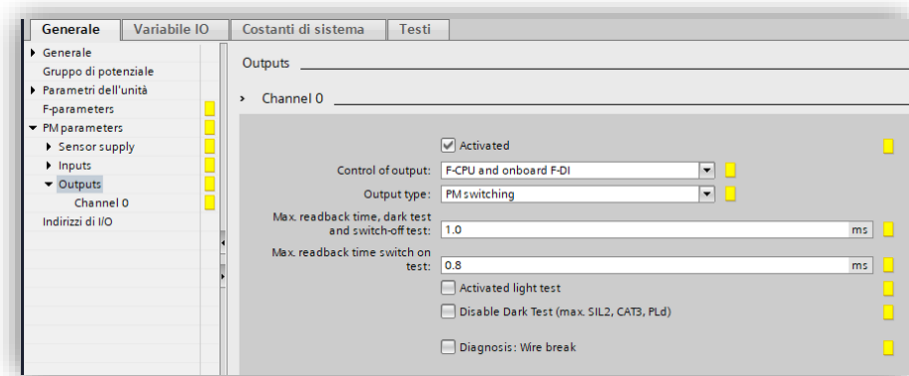
- il livello di sicurezza massimo raggiungibile è SIL2/Plid.
- è necessario un gruppo di potenziale per ogni gruppo di elettrovalvole legate alla stessa funzione sicura. Ovviamente tutte le elettrovalvole collegate dopo il PM sono tagliate in contemporanea, diversamente dalla soluzione con singoli canali d'uscita Safety.

Nota 1: questa soluzione è esattamente analoga alla soluzione utilizzata in caso di gestione della sicurezza con elettromeccanica: tramite una coppia di contattori elettromeccanici viene tagliato il 24V di alimentazione del PM-E standard, facendo sì che le schede a valle vengano disalimentate.

In entrambi i casi il livello di sicurezza massimo raggiungibile è SIL2/Plid, la soluzione con PM-E F però presenta notevoli vantaggi e risparmi in termini di cablaggio e software.

Nota 2: lo scopo del power module safety non è quello di alimentare schede Safety, questa soluzione infatti non è conveniente e introduce inutili complicazioni software.

Oltre ai due parametri comuni a tutte le schede e a quelli relativi ai suoi canali di ingresso e uscita, si aggiungono i seguenti:



- Control of output:** Con questo parametro si sceglie come deve essere determinato lo stato dell'uscita a bordo del power module. Le scelte sono:
 1. F-CPU: l'uscita viene gestita esattamente come un'uscita del modulo F-DQ e quindi lo stato viene determinato dal programma di sicurezza;
 2. F-CPU and onboard F-DI: l'uscita viene portata a 0 se richiesto dal programma di sicurezza o se l'ingresso cablato a bordo del F-PM va a 0
- Output type:** con questo parametro si determina come deve essere eseguito il doppio taglio sull'uscita a bordo del modulo. Si può scegliere se eseguire un doppio taglio del positivo (PP switching) oppure un taglio sul positivo e uno sul negativo (PM switching). Se si esegue un doppio taglio del positivo, è possibile collegare gli attuatori solo con il positivo e quindi utilizzare pacchi di elettrovalvole con le masse in comune e attuatori non isolati da terra.

Esistono dei bit di diagnostica specifici per i canali di ingresso e uscita che troviamo direttamente nell'immagine di processo di ingressi della scheda stessa:

Byte input della CPU	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x+0	-	-	-	-	-	-	DI1	DI0
x+1	-	-	-	-	-	-	Stato diagnostica canale DI1	Stato diagnostica canale DI0
x+2	-	-	-	-	-	-	-	Stato diagnostica canale DQ0

Se il bit dello stato di diagnostica è pari a **true**, il canale non ha errori altrimenti significa che la scheda ha rilevato sul canale un'anomalia hardware ed ha quindi passivato il relativo canale. È poi disponibile una diagnostica generale della singola scheda che verrà approfondita nel *capitolo 2.2.1*. La diagnostica di singolo canale è attiva solo se, per la scheda in questione, è stata impostata la passivazione del singolo canale.

2.4. Software safety

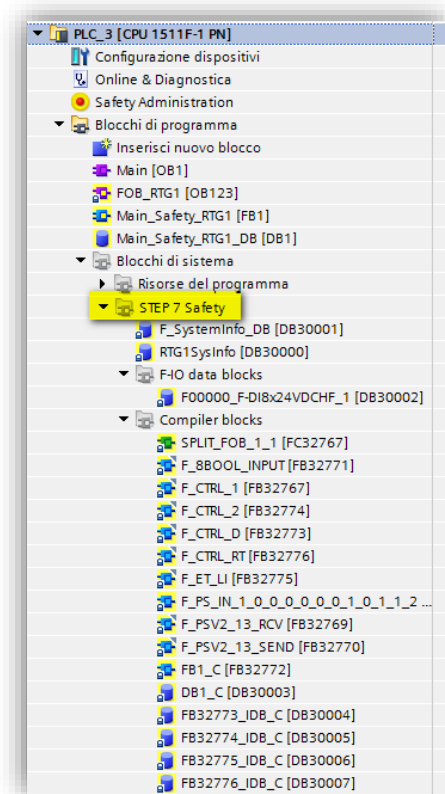
La parte introduttiva della gestione del software Safety è stata trattata nel capitolo "1.3 CPU Safety" ed è la base per quello che segue.

Il software Safety è fortemente legato alla configurazione hardware, in quanto, per la gestione corretta dei canali delle schede, è necessario che il software sappia come essi siano stati configurati affinché a livello di immagine di processo vengano elaborati in modo opportuno.

Una volta realizzata la configurazione hardware, illustrata nei capitoli precedenti, la cartella blocchi del PLC si presenta come l'immagine a fianco.

Sono stati creati in automatico nella cartella "blocchi di sistema - STEP 7 Safety":

- una serie di funzioni di sistema che quindi non sono utilizzati direttamente dal programmatore ma vengono usati in modo trasparente dal programma Safety
- il DB "RTG1SysInfo" di sistema, con alcuni parametri utilizzabili dal programmatore: stato di attivazione del programma di sicurezza, durata del tempo ciclo safety, firma del programma di sicurezza
- gli "F-IO data blocks", uno per ogni scheda, blocchi dati ai quali viene dedicato il prossimo capitolo.



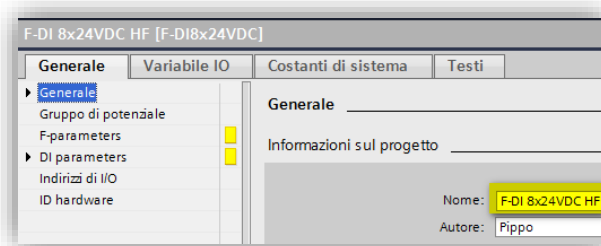
2.4.1. DB delle schede Safety

Tra i blocchi di sistema viene creato un DB per ogni dispositivo safety collegato alla CPU: schede I/O safety, azionamenti con funzionalità safety integrate oppure qualsiasi dispositivo collegabile sul bus che abbia funzionalità safety (encoder o altri sensori o attuatori safety).

Questi DB e le schede a cui sono associati sono identificabili tramite il simbolo a loro assegnato (per esempio **F00000_F-DI8x24VDCHF**).

Il simbolo del DB è costituito dalla "F" di fail safe seguita dall'indirizzo logico di partenza della scheda in questione (00000 per la scheda che ha area di ingressi e uscite che inizia dall'indirizzo 0) ed è concluso con il nome della scheda. Di default questo nome identifica la tipologia di scheda ed è uguale per tutte le schede dello stesso tipo. Nell'esempio si tratta di una scheda d'ingresso quindi: F-DI8x24VDCHF.

In caso di un numero elevato di schede della stessa tipologia, suddivise in più stazioni di periferia, è possibile cambiare il nome, nelle proprietà della scheda in configurazione



hardware, per rendere più immediato il riconoscimento della scheda e quindi della relativa DB.

La struttura di questi DB è identica per ogni tipologia di modulo e all'apertura si presenta come l'immagine seguente.

	Nome	Tipo di dati	Valore di avvio	Istantanea	A ritenzio...	Visibile in ..	Valore di i..	Commento
1	Input							
2	PASS_ON	Bool	false			<input checked="" type="checkbox"/>		1=Enable passivation
3	ACK_NEC	Bool	true			<input checked="" type="checkbox"/>		1=Acknowledgment for reintegration required
4	ACK_REI	Bool	false			<input checked="" type="checkbox"/>		1=Acknowledgment for reintegration
5	IPAR_EN	Bool	false			<input checked="" type="checkbox"/>		Tag for parameter reassignment of fail-safe DP.
6	Output							
7	PASS_OUT	Bool	true			<input checked="" type="checkbox"/>		Passivation output
8	QBAD	Bool	true			<input checked="" type="checkbox"/>		1=Fail-safe values are output
9	ACK_REQ	Bool	false			<input checked="" type="checkbox"/>		1=Acknowledgment requirement for reintegr...
10	IPAR_OK	Bool	false			<input checked="" type="checkbox"/>		Tag for parameter reassignment of fail-safe DP.
11	DIAG	Byte	16#0			<input checked="" type="checkbox"/>		Service information
12	InOut							
13	Static							

I bit significativi in lettura sono quelli dichiarati come "out", cioè in uscita dalla scheda e in lettura nel programma ed hanno il seguente significato

- **QBAD**: può essere interrogato per conoscere se sulla scheda è presente un errore, informazione cumulativa di tutti i possibili errori della scheda (0 = non sono presenti errori; 1 = sono presenti errori)
- **ACK_REQ**: la presenza di questo bit a "1" segnala che gli errori presenti sulla scheda sono stati corretti ed è quindi possibile ripristinarne il normale funzionamento previa conferma tramite pressione di un pulsante di riconoscimento (acknowledge) dall'operatore

Per inviare quest'informazione (acknowledge) al modulo è possibile collegare (nel programma safety) il bit del pulsante di riconoscimento direttamente al bit **ACK_REI** presente nei parametri "Input" della DB della scheda. Quest'operazione è da fare per ogni modulo.

Se si desidera, nella libreria safety fornita da Siemens, è presente una funzione **F_ACK_GL** che permette di ripristinare simultaneamente tutte le schede che lo richiedano, collegando il bit del pulsante di riconoscimento all'unico bit d'ingresso della funzione.

IMPORTANTE: quando il bit di **ACK_REQ** è = **TRUE**, e fino a quando non viene ripristinata la scheda, le informazioni presenti sui vari bit **QBAD** restano memorizzate. Questo permette di poter ancora leggere su quale canale si fosse generato l'errore, anche quando questo è stato corretto. Questa funzione permette di archiviare tutti gli errori, anche quelli che si correggono automaticamente e rapidamente.

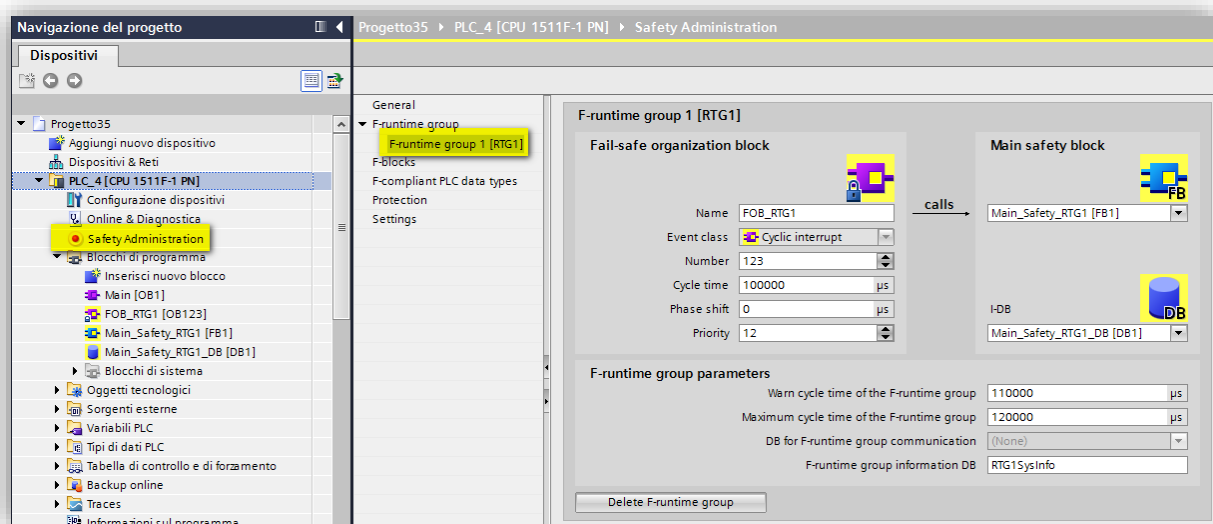
2.4.2. Struttura programma Safety

Come spiegato nel capitolo 1.3, le funzioni del programma safety convivono nella cartella blocchi insieme alle funzioni standard.

Il programma safety può essere strutturato nel numero desiderato di FC e FB (anche con multiistanza). Il requisito è che esista una funzione principale (un Main-Safety) al cui interno vengano richiamate tutte le altre.

La funzione Main-Safety è creata di default dal TIA come FB1 e richiamata automaticamente nell'OB123.

Il programmatore non deve fare altro che scrivere all'interno della FB1 tutto il codice safety e i richiami alle eventuali altre funzioni safety.



Nella tendina "F-runtime group", in Safety Administration, è possibile cambiare il Safety-Main e l'OB in cui viene richiamato, scegliendo altre numerazioni.

2.4.3. Scrittura del codice (KOP, FUP)

Nella scrittura del programma safety è possibile utilizzare solo i linguaggi KOP e FUP (non è possibile utilizzare AWL e SCL) con le stesse modalità di un programma standard per quanto concerne l'utilizzo dell'editor.

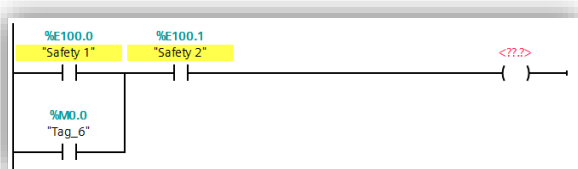
L'elenco delle operazioni che è possibile inserire è presente, come sempre, sulla destra dell'editor, con la comodità che, nel caso di elaborazione di una funzione safety, le operazioni ammissibili in una logica di sicurezza sono già state limitate e filtrate.

L'utilizzo del codice a contatti (KOP) permette di accorgersi immediatamente se è stato fatto un parallelo che potrebbe corrispondere ad un bypass illegale di una funzione di sicurezza.

Durante la scrittura di una funzione nel TIA Portal tutti i dati sicuri vengono evidenziati in giallo.

Di conseguenza ci si accorge subito se è stato utilizzato un dato non sicuro (quindi non giallo): questa segnalazione ci permette a colpo d'occhio di verificare se l'utilizzo del dato "non giallo" è ammesso.

Nel banale segmento KOP d'esempio, riportato nell'immagine qui a lato, si nota immediatamente che un ingresso sicuro E100.0 viene bypassato da un dato NON Safety M0.0 (riconoscibile in quanto bianco, e non giallo).



IMPORTANTE: in fase di compilazione non viene segnalato l'utilizzo inopportuno di un dato non safety, è compito del programmatore verificare che la logica rispetti i requisiti imposti dalla normativa (che potrebbe differire da progetto a progetto).

Un bypass non sicuro di un dato safety difficilmente potrà aver senso, tuttavia l'utilizzo di dati non sicuri per alcune funzioni è ammesso.

Si consiglia di consultare la normativa per avere maggiori dettagli in tal senso.

Un esempio per valutare se esiste la possibilità di utilizzare un ingresso (o merker o variabile di DB/DB-F) NON safety per una determinata funzione potrebbe essere sufficiente porsi una domanda: un guasto di questo dato porta ad un rischio per l'operatore?

Se la risposta è NO potrebbe significare che l'utilizzo del dato non safety è ammesso.

Una delle principali carenze dei dati non safety è la mancanza di una sua diagnostica. Questa mancanza impedisce che una corruzione del dato venga rilevata, rendendolo appunto non sicuro nel momento in cui si potrebbe verificare un rischio per l'operatore.

Queste considerazioni sono un aiuto, ma non sollevano programmatori e sviluppatori dalla consultazione delle normative specifiche per il progetto.

L'utilizzo più comune di dati non safety per funzioni sicure è per i riconoscimenti (acknowledge) e i feedback:

- gli **acknowledge** per il ripristino delle emergenze e dei guasti delle schede non richiedono l'uso di dati safety in quanto lavorano sul fronte di salita! Se il dato utilizzato si guasta, cioè rimane incollato a 1 o 0, questo non causa nessun ripristino involontario proprio perché è richiesto un fronte di salita.
- i **segnali di feedback** per le uscite safety richiedono un funzionamento sincrono con il comando dell'uscita stessa. Infatti, ci si aspetta che al comando dell'uscita, il canale d'ingresso a cui è

collegato il feedback dell'attuatore reagisca cambiando di stato, per la precisione con un segnale opposto al segnale di comando, ad esempio relè o contattori di carico con contatti a guida forzata o contatti speculari; quindi quando l'uscita safety va a 0, cioè è richiesta l'apertura dei contattori a lei collegata, il segnale di feedback (che corrisponderà alla serie dei feedback degli attuatori collegati all'uscita) deve andare a 1 entro il tempo di controllo sul feedback impostato. Il collegamento del circuito di feedback a un DI è in molti casi sufficiente. Nei seguenti casi, potrebbe essere ragionevole o necessario collegare il circuito di feedback a un F-DI:

- configurazione di attuatori a singolo canale, tuttavia è necessaria un'elevata copertura diagnostica
- alcune funzioni di diagnostica (ad esempio blocco STEP 7 "FDBACK") non sono possibili
- utilizzo di un modulo fail-safe in un I/O distribuito per utilizzare i meccanismi di sicurezza di PROFIsafe.

2.4.4. Comando uscite Safety

Ora una breve parentesi sul comando delle uscite safety.

Come spiegato nel capitolo 1.3 "CPU Safety", le uscite sicure (come i DB safety) possono essere scritte solo dal programma sicuro.

Un accesso in scrittura alle uscite (o a qualsiasi altro dato di DB safety) da parte di una funzione standard causa lo stop del PLC. In fase di compilazione il TIA segnala come avvisi questi accessi (solo se il codice è scritto in KOP): è molto importante correggerli!

Due semplici suggerimenti possono aiutare molto:

- mantenere gli indirizzi logici delle schede standard e safety su aree lontane, per evitare errori di distrazione
- definire l'area di creazione dei DB safety lontana dall'area di numerazioni usata per i DB standard.

Per spiegare come queste limitazioni in scrittura dei dati safety influiscano nel lavoro del programmatore, di seguito è presentato un semplice esempio di migrazione da un progetto che prima gestiva le funzioni di sicurezza con elettromeccanica ad una soluzione con PLC Safety, con riferimento alla gestione di un arresto sicuro tramite la chiusura di elettrovalvole.

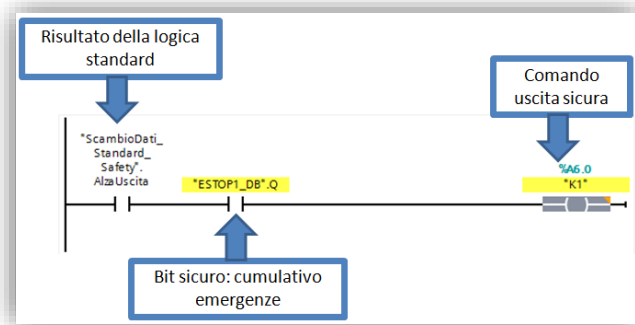
Nella soluzione elettromeccanica l'elettrovalvola veniva comandata da un PLC standard con la sua logica dedicata e in caso di richiesta di arresto sicuro veniva tagliata direttamente l'alimentazione dell'elettrovalvola causandone la chiusura.

Passando ad una soluzione con PLC Safety è possibile collegare l'elettrovalvola direttamente ad un'uscita safety, permettendo in alcuni casi di alzare il livello di sicurezza dell'arresto aumentando i controlli sull'uscita.

Il fatto che l'elettrovalvola sia collegata ad un'uscita safety non implica che tutta la sua logica di controllo debba essere spostata nel programma safety!

Infatti è possibile mantenere la logica nel programma standard, e quindi copiarla direttamente dal vecchio progetto limitando di parecchio i tempi di sviluppo dell'applicazione. L'unica modifica da apportare è scrivere il risultato della logica, invece che sull'uscita diretta, su un DB standard o un merker.

Dal programma safety, DB standard e merker possono essere letti direttamente, per cui ci si troverà con un segmento di questo tipo:



il comando dell'uscita sicura tramite un bit di logica standard è "messa in sicurezza" utilizzando in serie uno o più bit sicuri.

Nel caso dell'esempio il bit sicuro rappresenta un cumulativo delle richieste d'emergenza, generato tramite la funzione ESTOP1 (cap 2.3.1).

Nel momento in cui i bit di sicurezza sono aperti (zero) qualsiasi informazione venga trasmessa dalla logica standard non arriverà mai all'uscita

essendo tagliata in serie dai bit sicuri.

Nel caso sia richiesto dal livello di sicurezza il controllo del feedback dei dispositivi sicuri, sarà necessario usare la funzione feedback prima di comandare direttamente l'uscita.

2.4.5. Controllo scrittura dati Safety

Come spiegato ampiamente nei capitoli precedenti, **non è ammessa la scrittura di dati safety da parte di programma standard!**

Questo vuol dire che:

- le funzioni standard non devono assolutamente contenere operazioni di accesso in scrittura a dati Safety
- il pannello operatore non deve scrivere dati safety
- in una comunicazione S7 (standard) con un'altra CPU nessuna PUT (funzione di scrittura) deve puntare a dati safety.

In queste tre condizioni è sempre usato "NON DEVE", questo perché in realtà la scrittura è possibile e in molti casi non viene segnalata in fase di compilazione.

Qui sotto le due semplici regole riportate anche nel capitolo precedente per ridurre il rischio di errori:

- **mantenere gli indirizzi logici delle schede standard e safety su aree lontane, per evitare errori di distrazione**
- **definire l'area di creazione dei DB safety lontana dall'area di numerazioni usata per i DB standard.**

Anche in scrittura di funzioni standard, i dati safety sono evidenziati in giallo così da avere un aiuto per evitare di usarli in scrittura.

Se il linguaggio utilizzato per la scrittura della funzione standard è KOP o FUP l'accesso in scrittura dei dati safety viene segnalato con warning di sintassi, sia in fase di scrittura sia ogni volta che si esegue la compilazione.

Nel caso di warning di accesso in scrittura segnalati in compilazione è importante correggerli!

Un ulteriore consiglio per la ricerca di questo tipo di errori è quello di fare un controllo nella tabella variabili: selezionando ad una ad una le uscite safety è possibile aprire la schermata sotto "informazioni - Riferimenti incrociati" e verificare laddove sono state utilizzate.

2.6. Le funzioni Safety

Per agevolare il programmatore, Siemens mette a disposizione una serie di FB e FC in grado di implementare le principali funzioni richieste nella stesura di un programma sicuro. Queste sono disponibili tra le istruzioni nel momento in cui si apre l'editor di una funzione safety.

Queste funzioni sono certificate dal TÜV e velocizzano la scrittura del codice. Ciononostante, è anche possibile non utilizzarle e sviluppare autonomamente il codice per implementare queste funzionalità previa certificazione.

Nelle pagine seguenti della guida verranno trattate approfonditamente le funzioni più usate e importanti, mentre sarà solo accennato il funzionamento delle altre.

Il manuale "SIMATIC Safety - Configuring and Programming", scaricabile alla pagina [54110126](#) del sito di supporto, tratta in modo approfondito tutte le funzioni e tematiche Safety.

La maggior parte delle funzioni della libreria sono delle FB, quindi richiedono un DB d'istanza, che in questo caso sarà un DB Safety.

La comodità di avere un DB d'istanza rende superfluo appoggiare su DB o merker i valori d'uscita delle funzioni (e anche d'ingresso) poiché questi dati sono già disponibili come dati sicuri da utilizzare nel programma; servendosi dell'indirizzamento simbolico risulta veramente semplice puntare ai dati.

Le variabili del DB sono utilizzabili nel programma safety per gestire gli arresti sicuri ma anche nel programma standard per gestire gli arresti della parte non safety di macchina e per la supervisione. Quindi anche **i pannelli operatore possono accedere direttamente, solo in lettura, a questi dati**

2.6.1. ESTOP1

Questa funzione è alla base della realizzazione del programma safety e serve a generare un cumulo emergenze con gestione del ripristino e arresto ritardato.

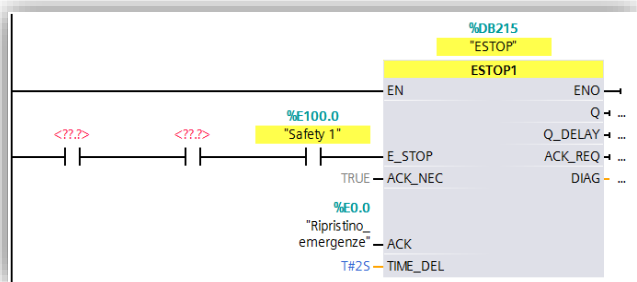
In fase di valutazione dei rischi (che dovrebbe essere stata ultimata prima dello sviluppo dell'applicazione) vengono definiti gli arresti sicuri richiesti dall'applicazione, attivati da diversi sensori.

Nel programma potrà essere richiamata una funzione ESTOP per accumulare tutte le condizioni che attivano un arresto sicuro e in

ingresso a ciascuna, sarà collegata la relativa serie di bit dei singoli sensori. (*)

La funzione presenta 2 bit d'uscita:

- **Q**: è il segnale di arresto immediato. Q è ad 1 finché le condizioni di sicurezza sono rispettate, cioè tutti i bit collegati all'ingresso E_STOP sono a 1. Nel momento in cui uno dei bit va a 0 anche Q immediatamente si porta a 0. Al ripristino di tutte le condizioni di sicurezza, la funzione segnala la possibilità di ritorno al normale funzionamento alzando il bit di uscita relativo alla richiesta di acknowledge (**ACK_REQ**). Quindi al verificarsi di un fronte positivo del bit d'ingresso **ACK** (per esempio collegato ad un pulsante di ripristino azionabile dall'operatore) il segnale Q torna allo stato di 1 (questo riconoscimento dell'errore è attivo solo se sul parametro **ACK_NEC** è su true)



- **Q_DELAY**: ha lo stesso comportamento di Q, con la differenza che il passaggio 1->0 viene ritardato del tempo impostato in **TIME_DEL** (nell'immagine 2 secondi). Con questa uscita verranno gestiti gli arresti ritardati.

(*) I pulsanti di arresto di emergenza possono essere collegati in serie fino a PL e (secondo ISO 13849-1) o SIL 3 (secondo IEC 62061) solo se si può escludere un malfunzionamento e la pressione contemporanea dei pulsanti di arresto di emergenza.

2.6.2. FDBACK

Per raggiungere i livelli di sicurezza più alti nell'ambito dell'automazione industriale è richiesto che i dispositivi collegati ad ogni uscita safety siano monitorati con un controllo di feedback.

Dove è richiesto questo controllo bisogna inserire una funzione di feedback prima del comando di ogni uscita.

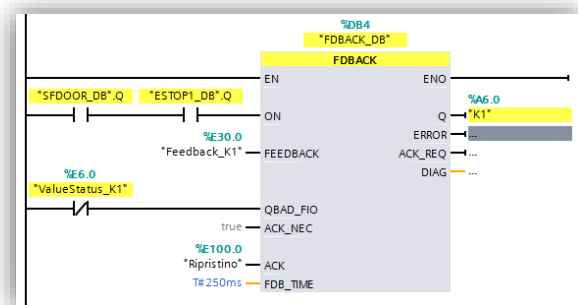
La funzione di feedback è un filtro tra la logica di comando e l'uscita che, in caso di errore di feedback, impedisce che venga riportata a 1 l'uscita anche se le condizioni logiche sono tutte ok.

Alle uscite safety del PLC è possibile collegare più

di un attuatore in parallelo. Questo richiede che i feedback di tutti gli attuatori collegati alla stessa uscita siano letti in serie e controllati dalla stessa funzione FDBACK. Di seguito una descrizione dei parametri d'ingresso:

- **ON**: a questo ingresso vanno cablati tutti i bit di comando dell'uscita: possono essere bit di logica e bit di arresto (come spiegato nel capitolo 2.2.4 "Comando uscite Safety"). Nell'immagine è stato cablato il bit corrispondente all'uscita Q della funzione ESTOP.
- **FEEDBACK**: va collegato qui il bit d'ingresso a cui è stata cablata la serie dei feedback. Il sensore di feedback non è necessario ai fini della sicurezza cablarlo su di un ingresso safety; può essere sufficiente anche collegarlo ad un ingresso standard. (*)
- **QBAD_FIO**: a questo ingresso va cablato il bit di diagnostica dell'uscita. Questo è presente nell'immagine di processo di ingressi della scheda stessa come descritto nel capitolo 2.1.
- **ACK**: come nel caso della funzione ESTOP, il segnale di riconoscimento (acknowledge) della funzione di feedback può essere un segnale non sicuro, quindi un canale d'ingresso standard oppure un merker controllato da una logica opportuna.
- **FDB_TIME**: dipende dall'architettura hardware del sistema e indica il tempo accettabile di ritorno del segnale di feedback. In caso di superamento di questo tempo viene segnalato un errore.

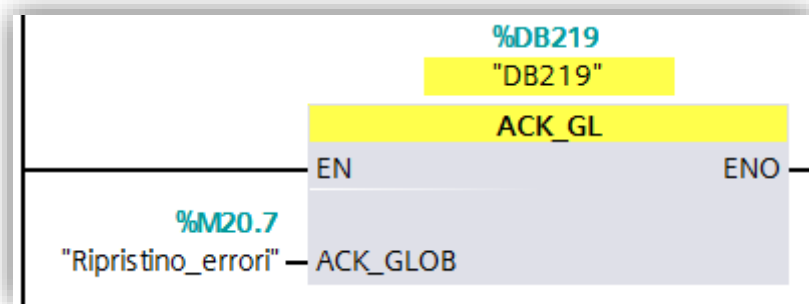
Al bit di uscita **Q** va cablata direttamente l'uscita, mentre tramite il bit **ACK_REQ** la funzione segnala che l'errore di feedback è stato corretto e con un fronte di salita su **ACK** è possibile tornare al normale utilizzo dell'uscita (sempre con **ACK_NEC** = true).



- (*) Il collegamento del circuito di feedback a un DI è in molti casi sufficiente.
- Nei seguenti casi, potrebbe essere ragionevole o necessario collegare il circuito di feedback a un F-DI:
- configurazione di attuatori a singolo canale, tuttavia è necessaria un'elevata copertura diagnostica
 - alcune funzioni di diagnostica (ad esempio blocco STEP 7 "FDBACK") non sono possibili
 - utilizzo di un modulo fail-safe in un I/O distribuito per utilizzare i meccanismi di sicurezza di PROFIsafe.

2.6.3. ACK_GL

La funzione ACK_GL permette di ripristinare le schede in passivazione i cui errori siano stati corretti. Come già accennato nel capitolo 2.2.1, le schede safety, prima di tornare al normale funzionamento dopo la correzione di un errore, richiedono un riconoscimento. Sarebbe possibile gestire il riconoscimento in modo automatico ma è sconsigliato per non perdere traccia di errori momentanei presenti sulle schede.



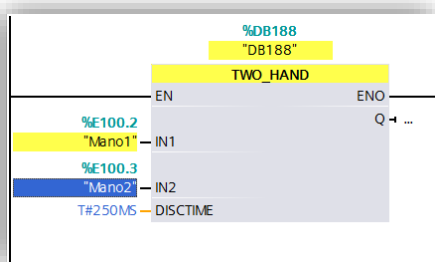
Agendo sull'unico bit d'ingresso di questa funzione è possibile ripristinare tutte le schede che lo richiedano senza variare il funzionamento delle altre, quindi:

- una scheda che sta funzionando normalmente non verrà interessata da questa funzione e continuerà a funzionare
- una scheda in passivazione con gli errori non ancora corretti continuerà a restare in passivazione
- tutte le schede in passivazione i cui errori siano stati corretti, verranno ripristinate in una volta sola evitando di doverle andare a ripristinare ad una ad una.

N.B. Come mostrato in precedenza, è possibile ripristinare un fault/guasto anche abilitando su fronte i bit di ACK_REI presenti negli F-I/O data blocks. Utilizzando il blocco funzione ACK_GL è possibile ripristinare simultaneamente tutti i fault/guasti presenti nel sistema; con il bit di ACK_REI si ripristina solo ed esclusivamente il modulo F-I/O corrispondente per cui bisognerebbe attivare tanti bit di ACK_REI quanti sono i singoli moduli da ripristinare.

2.6.4. TWO_HAND

Questa funzione gestisce il controllo di 2 sensori per generare un'abilitazione: controllo BIMANO.

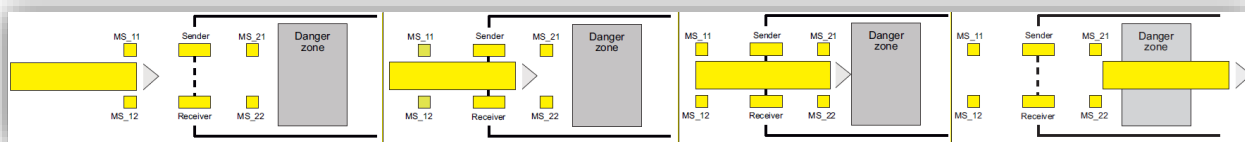


Se i due segnali d'ingresso vengono attivati con una discrepanza temporale massima di "DISCTIME", l'uscita Q viene tenuta alta per tutto il tempo in cui i due segnali restano entrambi attivi.

Esiste anche la funzione TWO_H_EN che è analoga, con un bit in più di Enable: se non è attivo (true) la funzione non alzerà mai Q.

2.6.5. MUTING

Questa funzione permette di gestire il muting di una barriera secondo i requisiti normativi.



Viene gestito l'asservimento automatico di una zona di lavoro pericolosa passando attraverso una barriera. Per lavorare la funzione ha bisogno di 2 o 4 fotocellule (a seconda della configurazione scelta). La sequenza richiesta con 4 fotocellule è la seguente: l'interessamento dei due sensori a monte della barriera attiva la funzione di muting (maschera la barriera); i due sensori a valle della barriera devono essere interessati quando i primi non sono ancora stati rilasciati, a questo punto possono essere rilasciati i primi e successivamente quelli a valle. Anche in questo caso esiste la versione MUT_P con funzionamento analogo e bit di Enable per attivare la funzionalità di muting.

2.6.6. ACK_OP

Per usare un pannello operatore per attivare riconoscimenti (acknowledge) è necessario usare questa funzione.

Un pulsante touch non è adatto per ripristinare funzioni sicure in quanto non è immune a tocchi involontari. La Siemens mette a disposizione questa funzione che permette di usare il pannello direttamente evitando di affiancare al pannello un pulsante fisico. La ACK_OP richiede che da pannello sia scritto su una variabile INT prima il numero 6 poi il numero 9 nell'intervallo massimo di 1 min. Se questo avviene in uscita viene alzato un bit che è possibile utilizzare per gli acknowledge.

2.6.7. Contatori e temporizzatori Safety

Nella libreria Safety sono presenti 3 contatori (CTU, CTD, CTUD) e 3 temporizzatori (TP, TON, TOF) di sicurezza.

Queste funzioni sono analoghe a contatori e temporizzatori IEC, presenti nella libreria standard, ma in versione sicura.

2.6.8. Gestione e conversione dati Safety

I DB safety si creano e si gestiscono come quelli standard, con una differenza fondamentale nella gestione dei dati: i dati safety devono essere usati (nel programma safety) con il formato con cui sono stati creati. Questo vuol dire che se è stata creata una word (DB1.DBW0) non è possibile accedere direttamente ad un bit della stessa (DB1.DBX0.0), ma è necessario usare funzioni di conversione ed inoltre non è possibile utilizzare la proprietà di *ritenzione* dei dati.

Qui un elenco delle funzioni disponibili:

- SHL e SHR: funzioni per shift a destra o a sinistra dei bit di una word safety, di un numero di bit passabile come parametro d'ingresso
- BO_W e W_BO: funzioni per conversione bool → word e viceversa
- SCALE: funzione per la scalatura di valori interi

3. Comunicazione safety

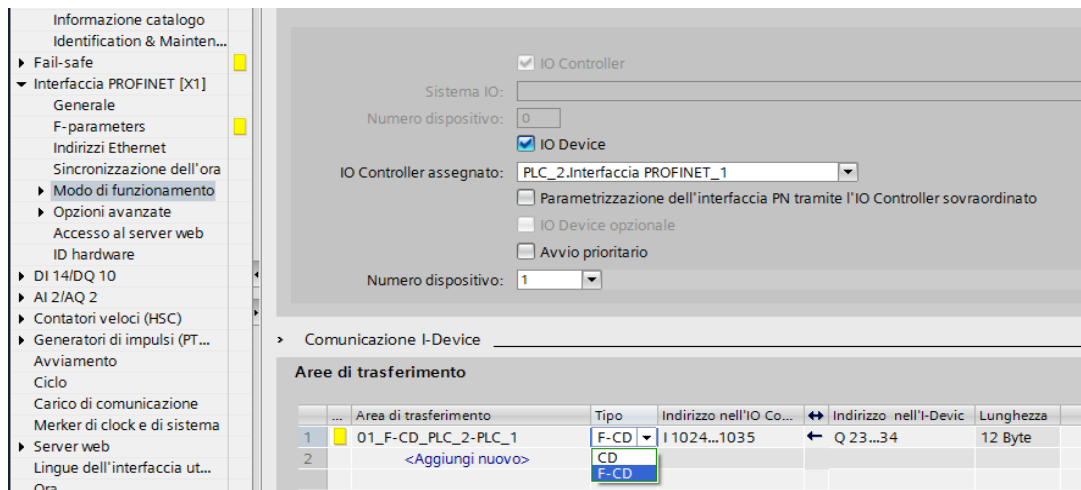
Nell'ambito della gestione safety, è possibile anche instaurare tra dispositivi una comunicazione che consente lo scambio di dati rispettando le regole della safety. Sono disponibili due tipologie di comunicazione safety: una basata su PROFINET tramite il protocollo sicuro PROFISafe che utilizza il concetto di scambio dati tra Controller e Device ed un'altra basata su TCP/IP che prende il nome di *Flexible F-Link*. Di seguito verranno presi in esame entrambi gli scenari. In questo documento quando si parla di dati "sicuri" o di "sicurezza" non ci si riferisce a concetti relativi alla cyber security, ma bensì, alla sicurezza del dato relativo al mantenimento del grado safety.

3.1. Comunicazione safety tramite PROFISafe

Tramite il protocollo PROFISafe ed i blocchi funzione SENDDP e RCVDP è possibile scambiare una serie di dati sicuri mantenendone le caratteristiche di sicurezza e quindi ad esempio **un fungo letto da una CPU può controllare un arresto sicuro di un'altra CPU con cui la prima sta comunicando**. Queste funzioni hanno bisogno di un'area di scambio dati consistente che è possibile creare nelle seguenti situazioni:

- interfaccia Controller/I-Device in Profinet
- interfaccia Master/Slave in Profibus
- appoggiando questi dati su un PN/PN coupler
- appoggiando questi dati su un DP/DP coupler.

Per la creazione di queste aree di scambio, indipendentemente dalle situazioni descritte prima, una volta dichiarato chi assume il ruolo di Device/Slave, nel configurare le aree di scambio come *tipo* si deve scegliere **F-CD**. In questo modo, il software in automatico, imposta a 12 byte la lunghezza dell'area di scambio (non modificabile in quanto numero massimo di dati che si possono scambiare in una comunicazione sicura tra due partner).



Nel caso di utilizzo con funzioni safety, per la configurazione di queste aree non è la necessità di implementare ulteriori metodi, ma devono solo essere rispettati due requisiti:

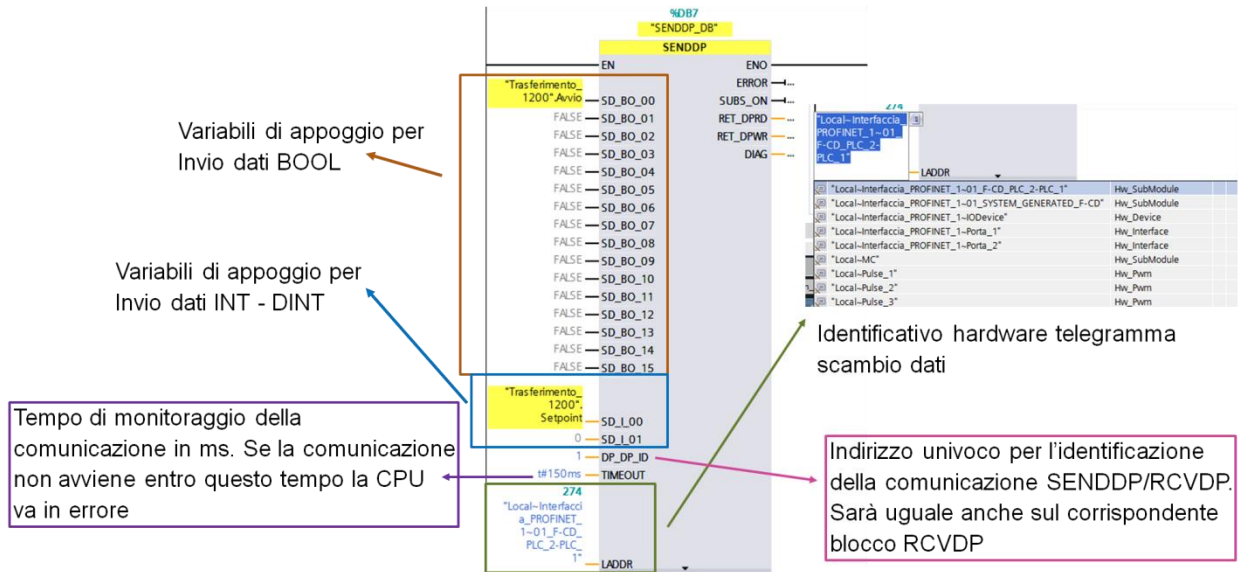
1. Dimensione fissa: per la Send deve essere 12 byte output e 6 byte input;
per la Receive deve essere 6 byte output e 12 byte input
2. Gli indirizzi dell'area di processo di Input e Output della singola funzione (Send o Receive) devono iniziare dallo stesso numero.

Le funzioni devono essere richiamate all'interno del Safety-Main (non è possibile in sottofunzioni e non è possibile usare multiistanze per queste FB) in posizioni specifiche:

- RCVDP: all'inizio del codice del Safety-Main
- SENDDP: alla fine del codice del Safety-Main.

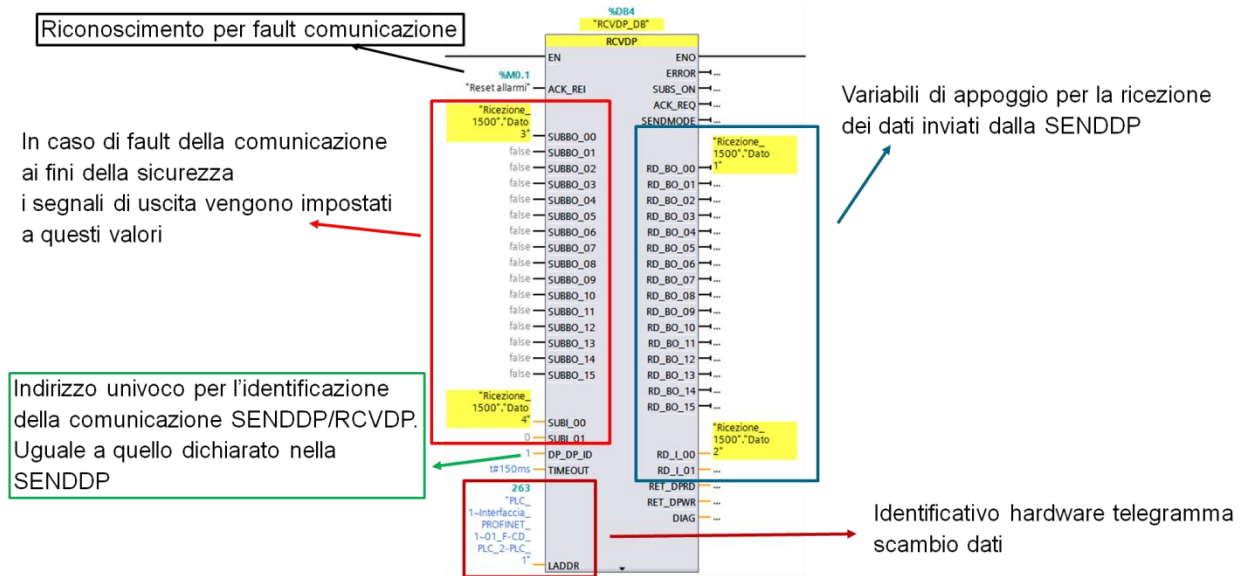
Tramite la **SENDDP** è possibile inviare **16 bit e 2 interi sicuri** in formato INT o DINT. Se ci fosse la necessità di inviare più dati è possibile richiamare più SENDDP, ognuna appoggiata ad aree diverse e con diverse RCVDP nella CPU partner.

I parametri di configurazione della **FB SENDDP** sono:



Tramite la **RCVDP** è possibile ricevere 16 bit e 2 interi sicuri in formato INT o DINT. Se ci fosse la necessità di inviare più dati si possono richiamare più RCVDP, ognuna appoggiata ad aree diverse e con diverse SENDDP nella CPU partner.

I parametri di configurazione della **FB RCVDP** sono:



Una volta configurati i parametri correttamente, la **SENDDP** continuerà ad inviare i dati, restituendo sul bit d'uscita ERROR lo stato della comunicazione.

La **RCVDP** invece ha un comportamento che richiede un approfondimento. In uscita si trovano 16 bit e 2 interi che, a seconda dello stato del bit SUBS_ON, trasmetteranno valori come segue:

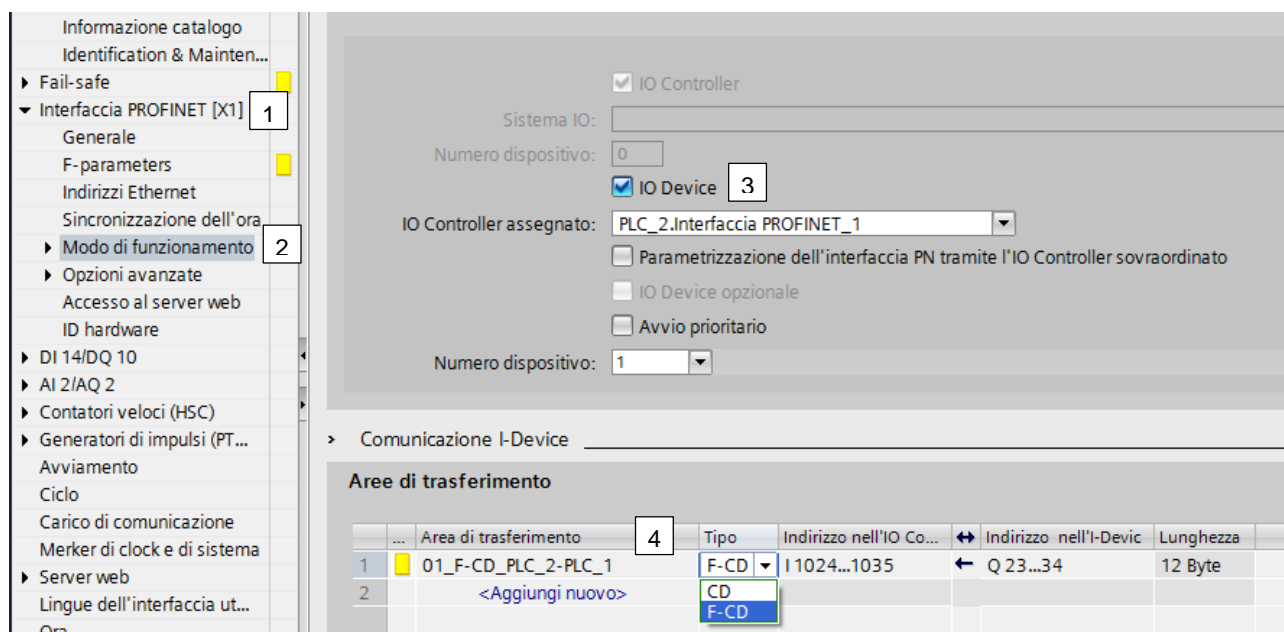
- SUBS_ON=0, i valori sono quelli ricevuti dall'altra CPU, quindi la comunicazione sta funzionando correttamente
- SUBS_ON=1, i valori riportati in uscita sono quelli passati in ingresso alla funzione, appunto i valori sostitutivi. Questi valori di default si trovano a 0 (zero), condizione di sicurezza, valori che possono però essere modificati a seconda delle esigenze.

Il valore SUBS_ON=1 è presente allo start up del PLC e in presenza di errore (ERROR=1). Per uscire dallo stato di “valori sostitutivi” e per ripristinare la comunicazione a seguito di errori, è necessario passare un acknowledge alla funzione tramite il bit ACK_REI in ingresso alla stessa. **La richiesta di ACK è segnalata dal bit ACK_REQ.**

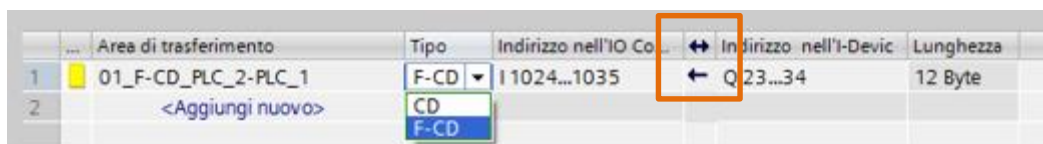
I bit in uscita dalla Receive sono bit sicuri e come tali possono controllare arresti sicuri della macchina! Possono essere appoggiati a variabili o usati direttamente accedendo al DB d'istanza della funzione.

3.1.1. Esempio applicativo SENDDP & RCVDP in un progetto integrato

In un progetto integrato è possibile instaurare una comunicazione safety sfruttando la comunicazione Controller/I-Device senza l'utilizzo di GSD esterni. Supponiamo che si voglia instaurare una comunicazione safety tra una CPU S7-1512F (I-Device PLC_1) e una CPU S7-1214F (Controller PLC_2). Nelle proprietà dell'interfaccia PROFINET della CPU S7-1512F, si deve parametrizzare la funzionalità di I-Device. Come mostrato in figura, sotto la voce *Interfaccia PROFINET* (1) → *Modo di funzionamento* (2) → selezionare la funzionalità di *IO Device* ed assegnare il rispettivo controller (3). Sotto la voce *Aree di trasferimento*, inserire l'area di trasferimento safety scegliendo come *Tipo* → *F-CD* (4).



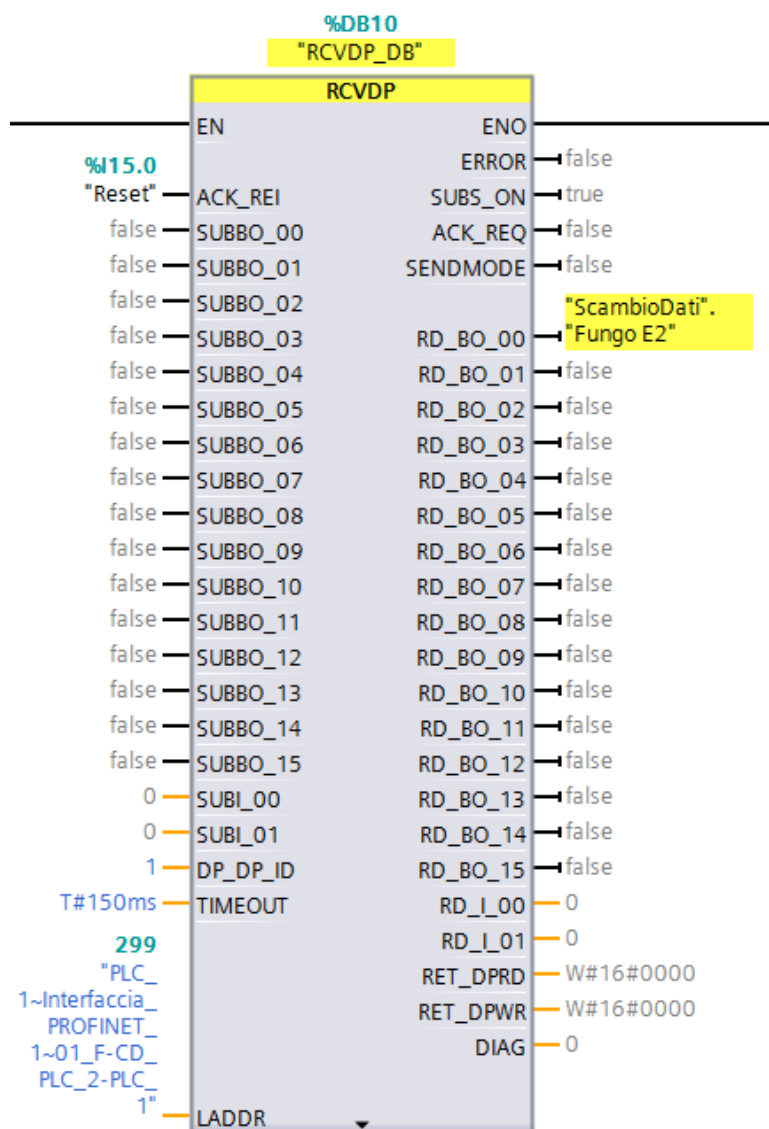
Una volta parametrizzato l'I-Device, bisogna programmare a livello software i blocchi funzione SENDDP e RCVDP descritti in precedenza. Come mostra la direzione della freccia nella definizione dell'area di trasferimento (riquadro arancio in figura), l'I-Device fungerà da “sender” e il Controller da “receiver”.



- Programmazione “receiver” CPU S7-1214F (Controller): nel primo segmento del Main Safety inserire il blocco funzione RCVDP dove in DP_DP_ID si deve inserire un numero identificativo della comunicazione che deve essere uguale anche nella programmazione della SENDDP lato I-Device;

LADDR corrisponde all'ID hardware dell'area di trasferimento safety, che è specificato nella tabella delle *Costanti di sistema* dell'I-Device come mostrato dal riquadro arancio in figura

PLC_1 [CPU 1512SP F-1 PN]				
Generale		Variabile IO	Costanti di sistema	Testi
Visualizza costanti di sistema ha				
Nome	Tipo	ID hardware	Utilizzato da	
PLC_1~Interfaccia_PROFINET_1~IODevice	Hw_Device	290	PLC_2	
PLC_1~Interfaccia_PROFINET_1~01_F-CD_PLC_2-PLC_1	Hw_SubModule	299	PLC_2	
PLC_1~Interfaccia_PROFINET_1~01_SYSTEM_GENERATED_F-CD	Hw_SubModule	300	PLC_2	
Local~Interfaccia_PROFINET_1~IODevice	Hw_Device	282	PLC_1	
Local~Interfaccia_PROFINET_1~01_F-CD_PLC_2-PLC_1	Hw_SubModule	284	PLC_1	
Local~Interfaccia_PROFINET_1~01_SYSTEM_GENERATED_F-CD	Hw_SubModule	285	PLC_1	
Local_MC	Hw_SubModule	51	PLC_1	

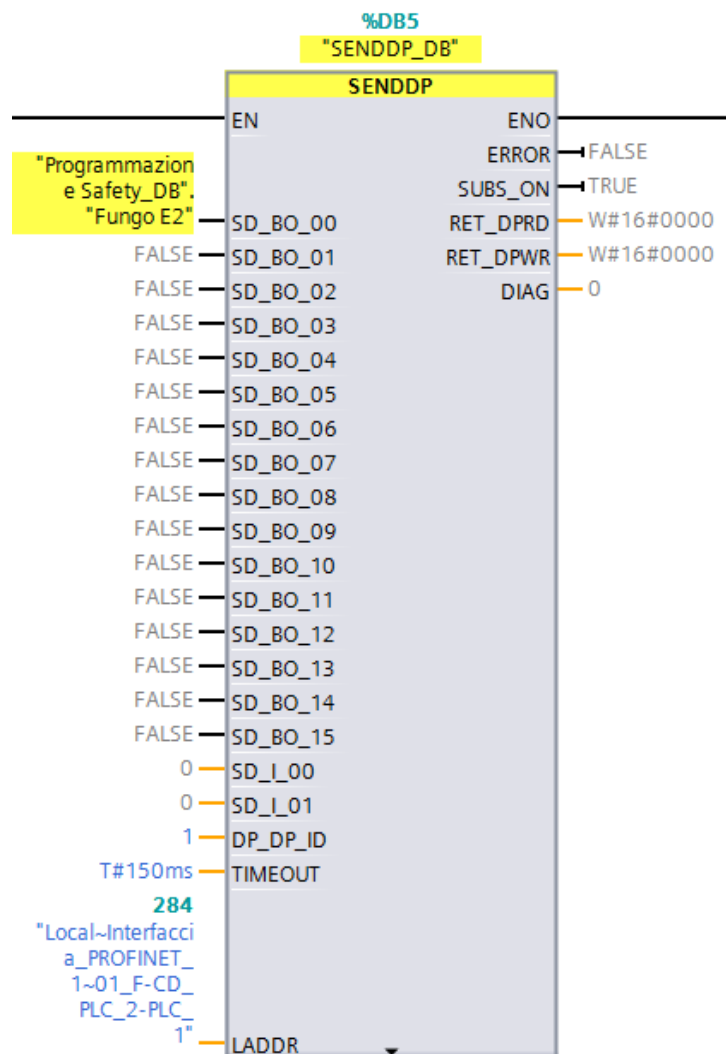


N.B. nel caso dovessero verificarsi errori nella comunicazione, dopo averli risolti, per la reintegrazione del fault di comunicazione, dare un fronte di salita sul bit di *ACK REI*.

- Programmazione “sender” CPU S7-1512F (I-Device): nell'ultimo segmento del Main Safety inserire il blocco funzione *SENDDP* dove in *DP_DP_ID* si deve inserire un numero identificativo della comunicazione che deve essere uguale a quello inserito nella programmazione della *RCVDP*

corrispondente lato Controller; *LADDR* corrisponde all'ID hardware locale dell'area di trasferimento safety, che è specificato nella tabella delle *Costanti di sistema* dell'I-Device come mostrato dal riquadro arancio in figura.

PLC_1 [CPU 1512SP F-1 PN]				
Generale		Variabile IO	Costanti di sistema	Testi
Visualizza costanti di sistema ha ▾				
Nome	Tipo	ID hardware	Utilizzato da	
PLC_1~Interfaccia_PROFINET_1~IODevice	Hw_Device	296	PLC_2	
PLC_1~Interfaccia_PROFINET_1~01_F-CD_PLC_2-PLC_1	Hw_SubModule	299	PLC_2	
PLC_1~Interfaccia_PROFINET_1~01_SYSTEM_GENERATED_F-CD	Hw_SubModule	300	PLC_2	
Local~Interfaccia_PROFINET_1~IODevice	Hw_Device	282	PLC_1	
Local~Interfaccia_PROFINET_1~01_F-CD_PLC_2-PLC_1	Hw_SubModule	284	PLC_1	
Local~Interfaccia_PROFINET_1~01_SYSTEM_GENERATED_F-CD	Hw_SubModule	285	PLC_1	
Local~MC	Hw_SubModule	51	PLC_1	

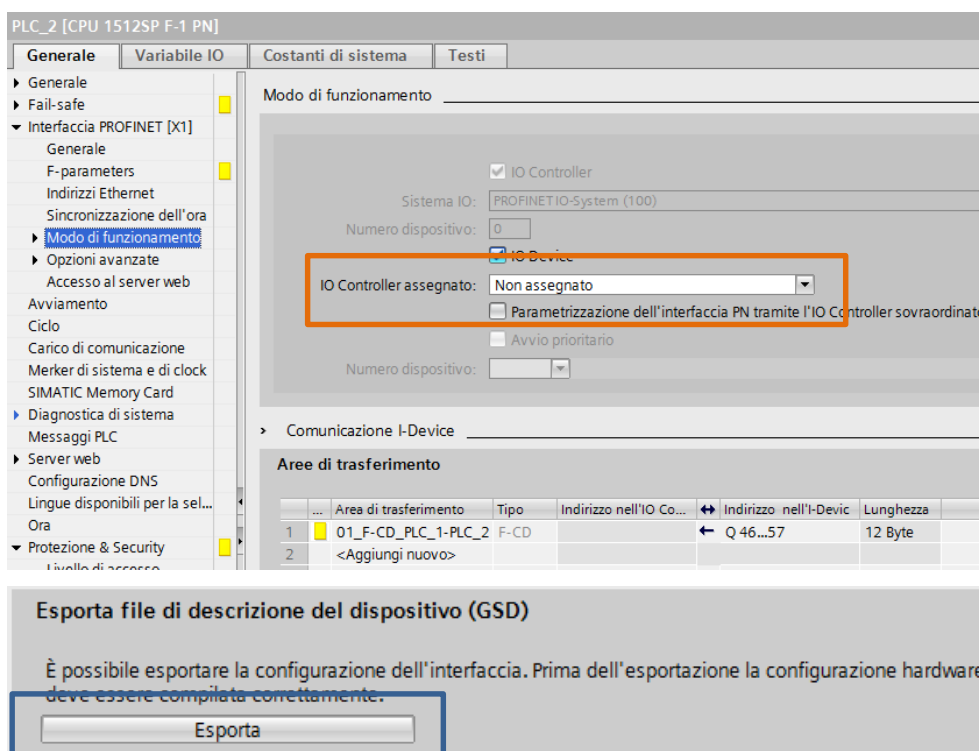


Dopo questi passaggi, il bit safety specificato nella SENDDP in corrispondenza di *SD_BO_00* (lato I-Device), verrà ricevuto sul bit *RD_BO_00* sulla RVDP (lato Controller). Questa programmazione è valida nel caso di un progetto integrato.

3.1.2. Esempio applicativo SENDDP & RCVDP in un progetto NON integrato

In un progetto NON integrato è possibile instaurare la comunicazione tramite l'utilizzo dei file GSD per i dispositivi che fungono da I-Device. Per poter generare il file GSD dell'I-Device con le aree di trasferimento comprensive dei dati safety, in TIA PORTAL è necessario seguire la seguente procedura:

- in una istanza di TIA PORTAL inserire il PLC che fungerà da I-Device
- inserire una CPU fail-safe "fittizia" che sarà il controller locale per l'I-Device
- creare la comunicazione Controller/I-Device specificando le aree di trasferimento safety (e anche standard se necessario)
- cancellare il controller "fittizio" inserito e sotto la voce *IO Controller assegnato* → *Non assegnato* (riquadro arancio in figura)
- compilare il progetto del PLC I-Device
- generare il file GSD cliccando sul tasto *Esporta* (riquadro blu in figura).



A questo punto il file GSD dell'I-Device contenente i dati safety da scambiare è pronto per essere importato nel progetto del Controller. Ovviamente, per lo scambio dati, resta valido il fatto che bisogna sempre programmare sia nel progetto del Controller che in quello dell'I-Device, i blocchi funzione SENDDP/RCVDP a seconda della direzione della freccia nella definizione dell'area di trasferimento.

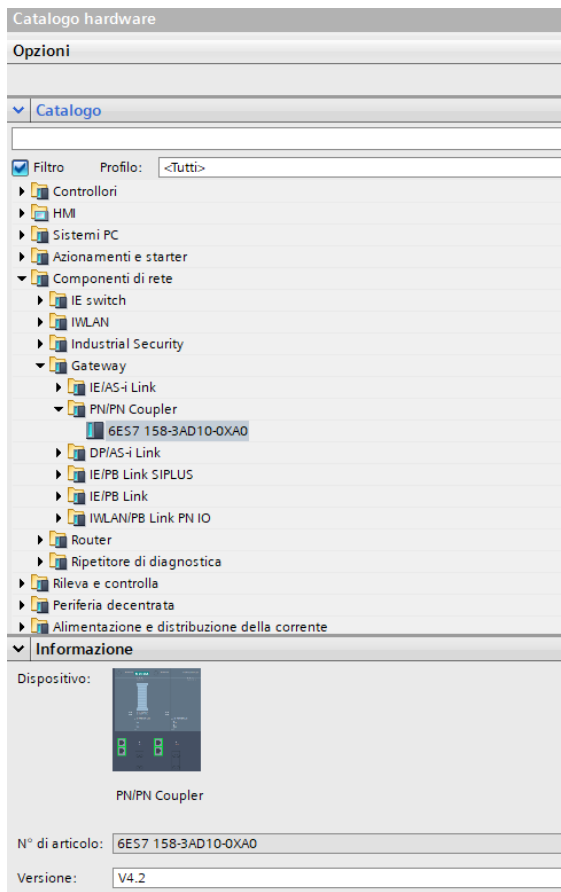
3.1.3. Esempio applicativo scambio dati safety tramite PN-PN coupler

Tramite il gateway profinet PN-PN coupler (6ES7158-3AD10-0XA0) è possibile scambiare dati, anche safety, tra due controller. Un altro aspetto fondamentale dell'utilizzo del PN-PN coupler, sta nel fatto che, tale dispositivo consente lo scambio dati anche se i due controller profinet appartengono a due sotto reti diverse;

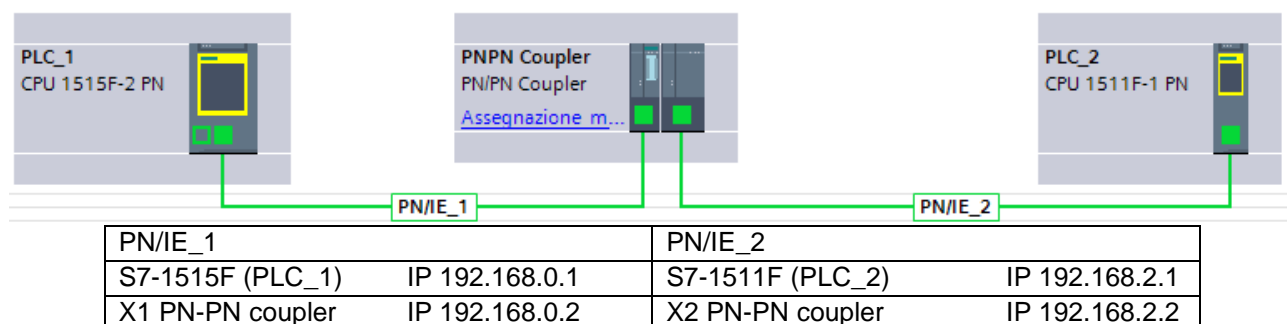
ricordiamo che il profinet, non supportando le vlan, non consente di per sé la comunicazione tra sotto reti appartenenti a classi diverse.

3.1.3.1. Utilizzo di PN-PN coupler in un progetto integrato

In un progetto integrato STEP7 in TIA Portal, il dispositivo PN-PN coupler si trova sotto la voce del catalogo hardware relativo ai *Componenti di rete* → *Gateway*.



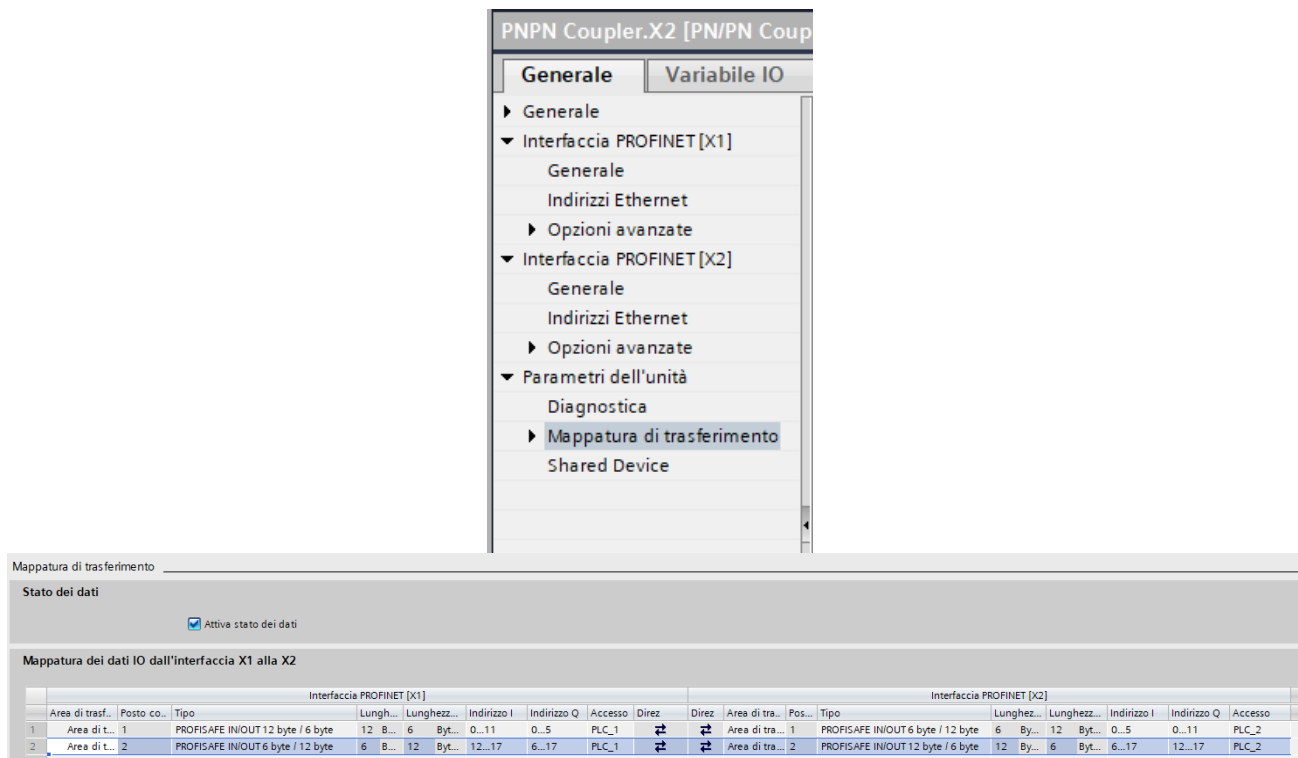
Inserire nella vista di rete il gateway e collegare all'interfaccia X1 il controller 1 ed all'interfaccia X2 il controller 2



Dato che il PN-PN coupler viene visto come un device sulla rete profinet, è necessario assegnare il nome dispositivo sia all'interfaccia X1 che all'interfaccia X2.

Nelle proprietà del PN-PN coupler dichiarare le aree di trasferimento safety da scambiare tra i due controller. Come mostrato in figura, sotto la voce *Parametri dell'unità* → *Mappatura di trasferimento* inserire una nuova

area di trasferimento e, con il menu a tendina nella proprietà *Tipo* dell'*Interfaccia PROFINET [X1]*, scegliere le aree *PROFISAFE IN/OUT* desiderate; in automatico, lato *Interfaccia PROFINET [X2]* verrà inserita un'area di trasferimento speculare a quella scelta per la X1.



A questo punto da entrambi i lati sono state dichiarate le aree di scambio dati safety che i due controller utilizzeranno per comunicare tra di loro. Nell'esempio descritto prendiamo il caso della prima area di trasferimento dichiarata:

- **PLC_1 (X1 PN-PN coupler) → RECEIVER**

Interfaccia PROFINET [X1]						
Tipo	Lunghezza I	Lunghezza Q	Indirizzo I	Indirizzo Q	Accesso	
PROFISAFE IN/OUT 12 byte / 6 byte	12	Byte(s)	0...11	0...5	PLC_1	

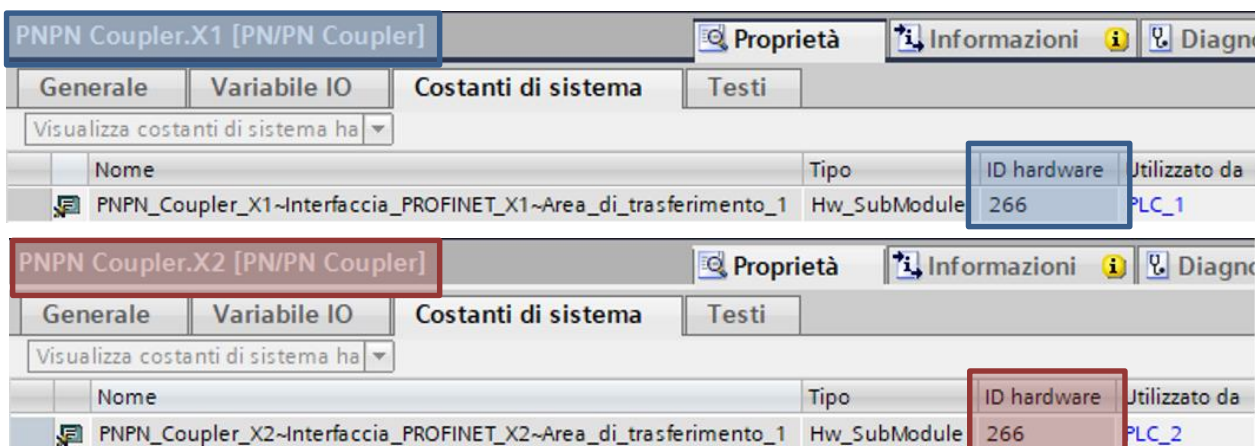
- **PLC_2 (X2 PN-PN coupler) → SENDER**

Interfaccia PROFINET [X2]					
Tipo	Lunghezza I	Lunghezza Q	Indirizzo I	Indirizzo Q	Accesso
PROFISAFE IN/OUT 6 byte / 12 byte	6	Byte(s)	0...5	0...11	PLC_2

Come mostrato per gli altri casi di comunicazione, anche per il PN-PN coupler c'è la necessità di utilizzare i blocchi funzione **RCVDP** e **SENDDP** opportunamente programmati. Di seguito uno schema di programmazione dei blocchi, in funzione dell'immagine di processo delle aree di trasferimento definite.



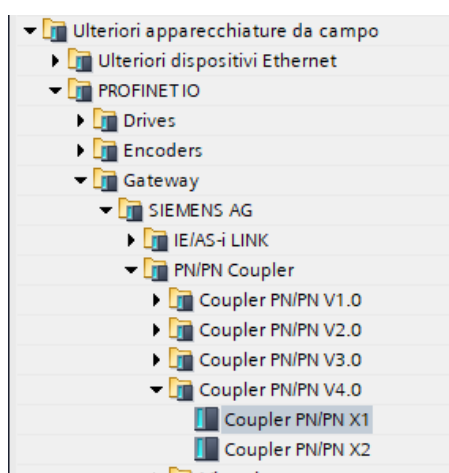
Come mostrato dalle figure seguenti, gli *LADDR* rappresentano gli ID HW delle aree di trasferimento in riferimento al plc utilizzato e *ID 2* rappresenta l'identificativo della relazione di comunicazione tra SENDDP e RCVDP.



A seguito di questi passaggi, il PLC_2, tramite il PN-PN coupler, invierà le informazioni safety salvate nell'immagine di processo Q 0..11 al PLC_1, che le riceverà sull'immagine di processo I 0..11.

3.1.3.2. Utilizzo di PN-PN coupler in un progetto NON integrato

Quando si parla di progetto NON integrato, si intendono tutti quei casi in cui i due controller che devono parlare tra di loro non sono stati progettati e programmati nello stesso ambiente di sviluppo. In tali situazioni, per la gestione del PN-PN coupler, bisogna ricorrere all'utilizzo dei file GSD. Nello specifico, si inserirà nel progetto 1 relativo al PLC_1, il GSD per il lato X1 del PN-PN coupler; per il progetto 2 relativo al PLC_2, si inserirà il GSD per lato X2 del PN-PN coupler. Se si utilizza Step7 in TIA Portal oppure Step7 classico, i GSD si trovano nel catalogo hardware sotto la cartella *Ulteriori apparecchiature da campo* → *PROFINET IO* → *Gateway* → *PN/PN Coupler* → scegliere la versione del dispositivo e quali GSD inserire per lato X1 o X2.

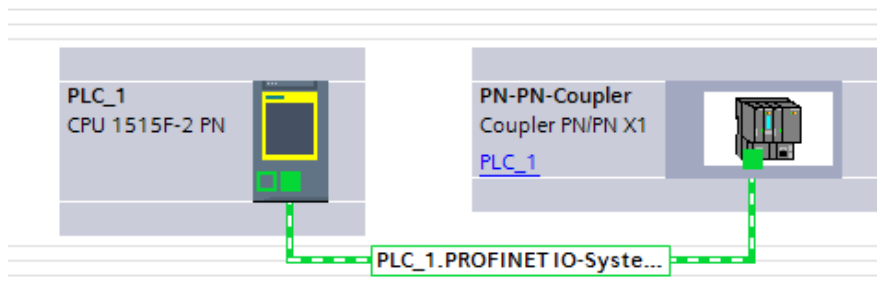


Per altri software di programmazione è possibile scaricare dal support Siemens i GSD per lato X1 oppure X2 al seguente link: <https://support.industry.siemens.com/cs/us/en/view/23742537>.

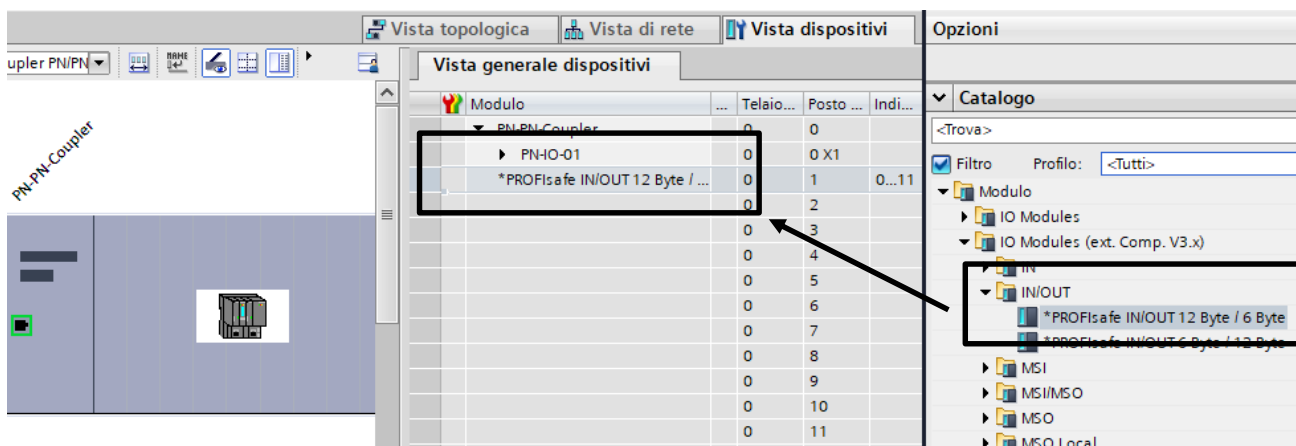
Vediamo adesso come implementare su entrambi i progetti la comunicazione safety utilizzando il PN-PN coupler.

Progetto 1 – PLC 1 – Lato X1 PN-PN coupler -- RECEVIER

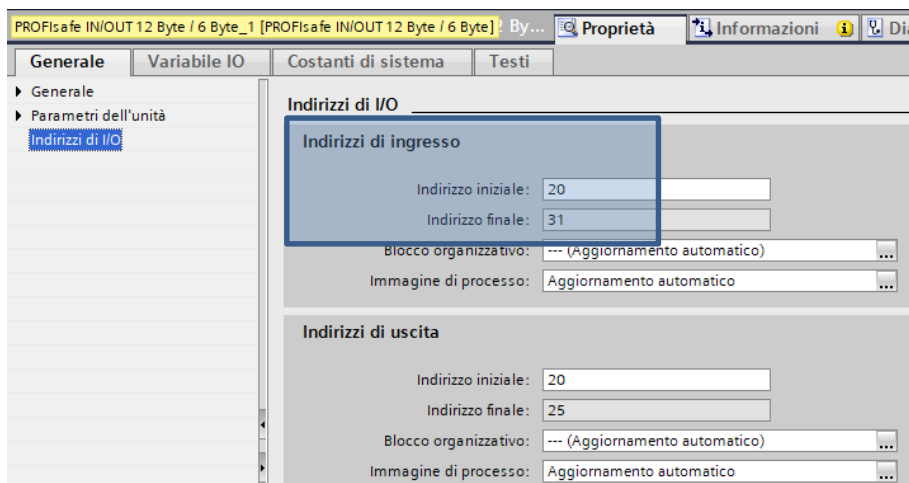
Nel progetto 1 inserire il PLC_1 ed il GSD del lato X1 del PN-PN coupler e collegarli tra di loro; assegnare il nome profinet al GSD del lato X1.



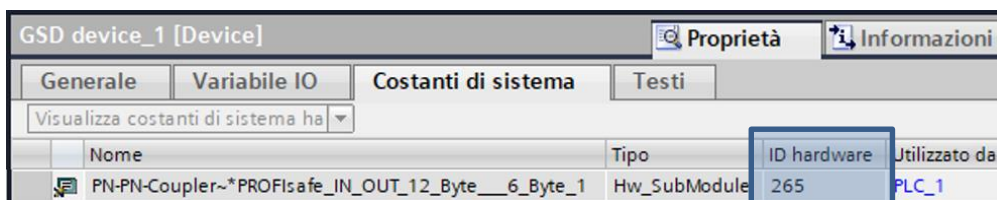
Entrare nella vista dispositivi del GSD lato X1 e, come mostrato in figura, inserire il modulo *IN/OUT* → **PROFISafe IN/OUT 12 byte / 6 byte* trascinandolo sotto la *Vista generale dispositivi*.



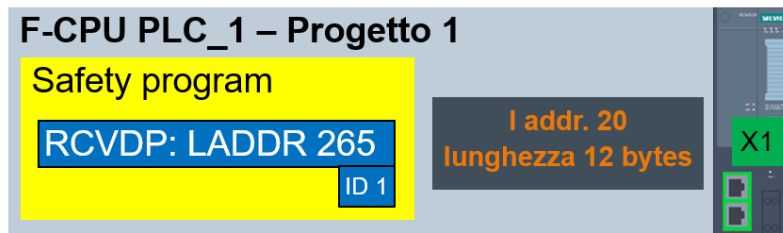
A questo punto, a tale modulo, verrà assegnata l'immagine di processo dove appoggiare i dati,



il cui ID HW si trova sempre nelle costanti di sistema come mostrato in figura.

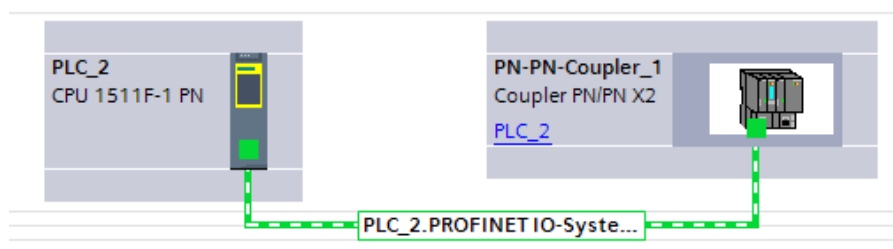


Essendo il PLC_1 il receiver della comunicazione, andrà programmato opportunamente il blocco funzione RCVDP in base agli indirizzi dell'immagine di processo assegnata ed all'ID HW associato al modulo inserito.

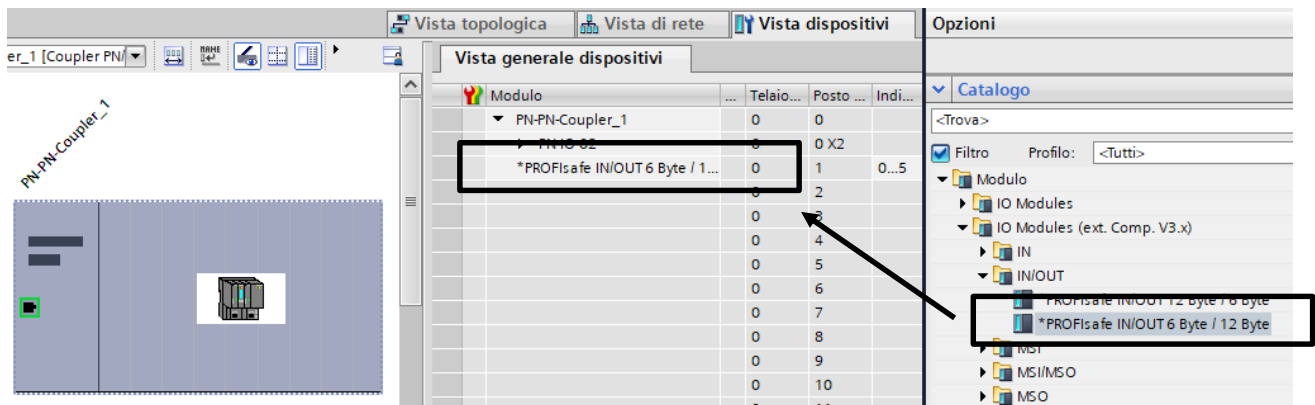


Progetto 2 – PLC_2 – Lato X2 PN-PN coupler -- SENDER

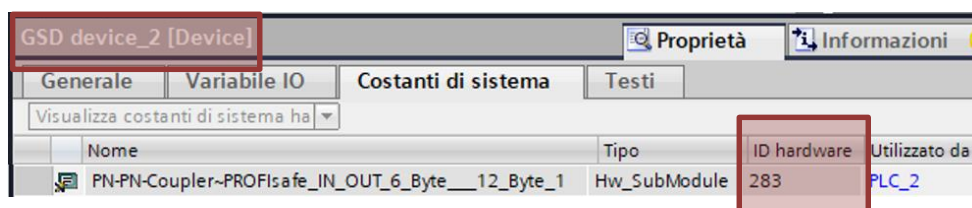
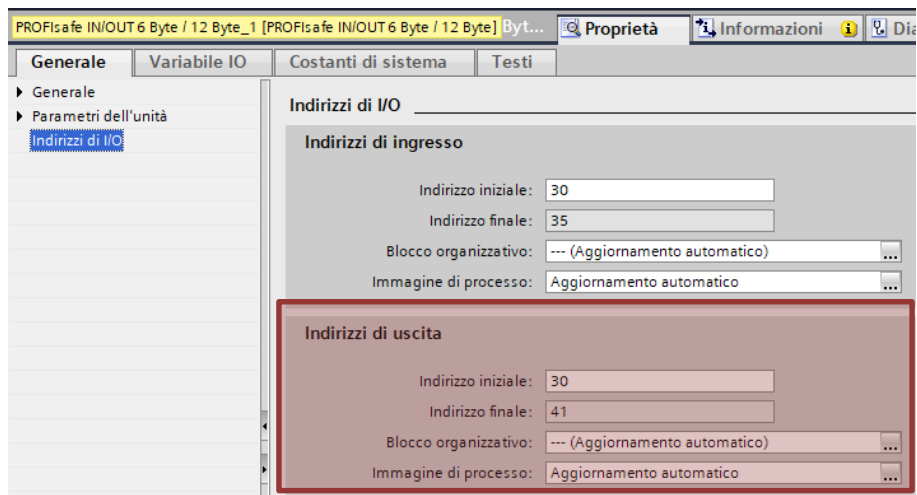
Nel progetto 2 inserire il PLC_2 ed il GSD del lato X2 del PN-PN coupler e collegarli tra di loro; assegnare il nome profinet al GSD del lato X2 (diverso da quello assegnato all'interfaccia X1 al punto precedente).



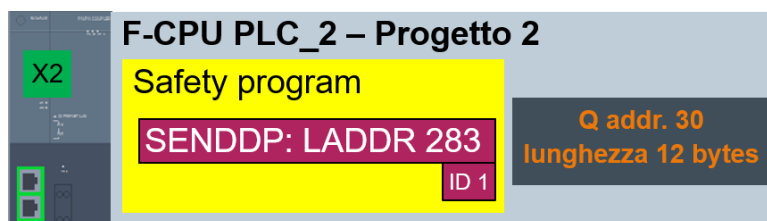
Entrare nella vista dispositivi del GSD lato X1 e, come mostrato in figura, inserire il modulo *IN/OUT* → **PROFISafe IN/OUT 6 byte / 12 byte* trascinandolo sotto la *Vista generale dispositivi*.



A questo punto, a tale modulo, verrà assegnata l'immagine di processo dove appoggiare i dati a cui verrà associato anche un ID HW



Essendo il PLC_2 il sender della comunicazione, andrà programmato opportunamente il blocco funzione SENDDP in base agli indirizzi dell'immagine di processo assegnata, all'ID HW associato al modulo inserito ed all'ID inserito nella parametrizzazione del blocco RCVDP nel progetto 1.



A questo punto, i dati safety del PLC_2 associati all'immagine di processo Q 30..41 verranno spediti attraverso il PN-PN coupler al PLC_1 sull'immagine di processo I 30..41, anche se i due controller sono stati progettati e programmati in ambienti di sviluppo differenti.

3.2. Comunicazione safety tramite TCP/IP – Flexible F-Link

Tramite la comunicazione Flexible F-Link è possibile scambiare dati fail-safe tra due F-CPU attraverso i meccanismi di comunicazione standard (quindi tramite blocchi funzione tipo TSEND, TRCV, GET, PUT, BSEND, BRCV, ecc..), senza l'utilizzo di PROFINET/PROFIsafe, Controller/Device e SENDDP/RCVDP. I vantaggi di questa soluzione sono:

- utilizzo di F-UDTs per lo scambio dati
- fino a 100bytes di dati fail-safe per F-UDTs
- generazione automatica di DBs di comunicazione fail-safe
- F-communication address signature separato per identificare facilmente cambiamenti nella comunicazione Flexible F-Link

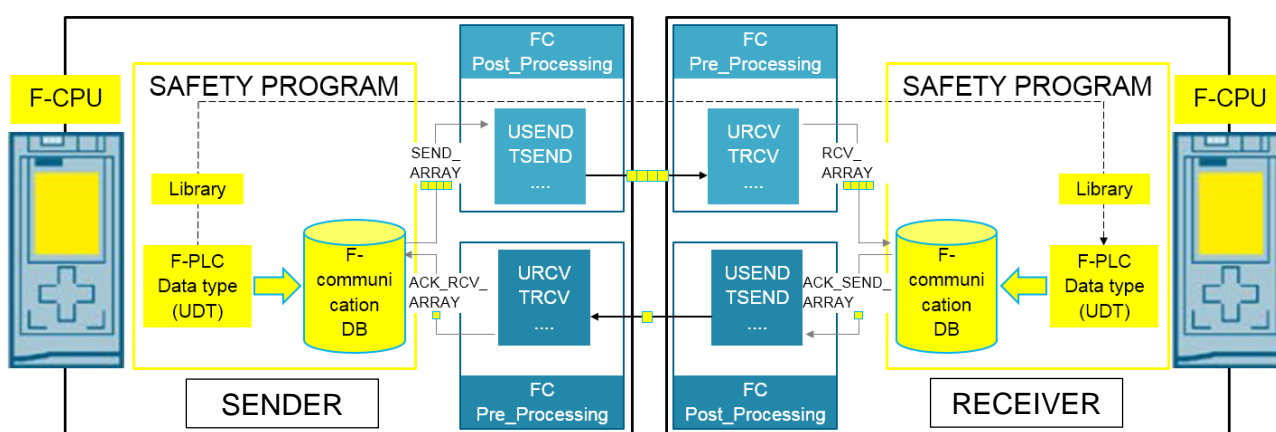
- possibilità di scambiare dati fail-safe anche tra VLAN diverse
- comunicazione tra due F-Runtime Group diversi

Requisiti di sistema richiesti:

- S7-1500/ET200SP CPU a partire dal firmware 2.0
- S7-1200 a partire dal firmware V4.2
- Safety system version V2.2
- Simatic Safety Basic/Advanced V15.1

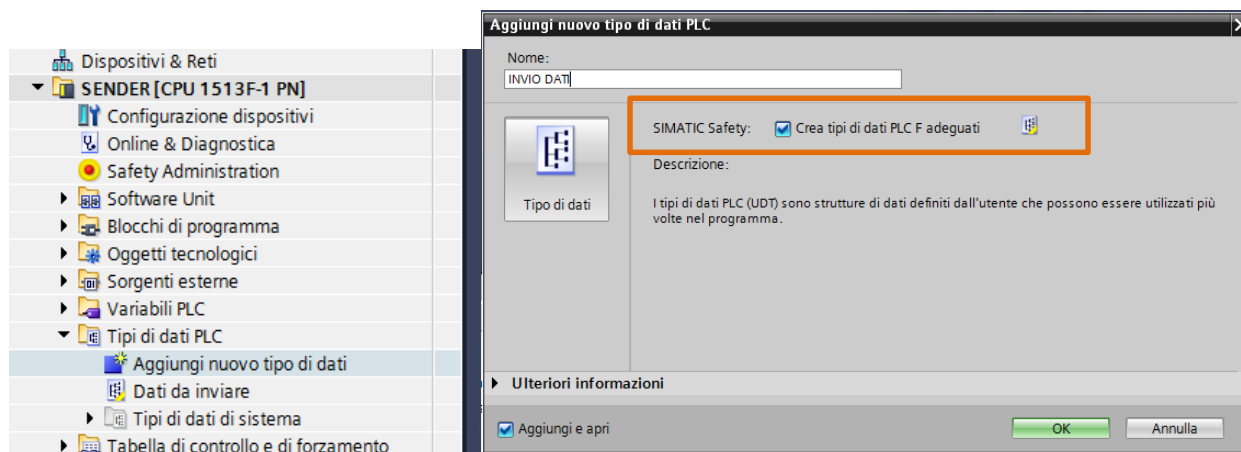
3.2.1. Flexible F-Link comunicazione F-CPU/F-CPU

Si descriverà ora brevemente il principio di funzionamento del Flexible F-Link. Come si può notare dall'immagine seguente, bisogna instaurare tra due F-CPU un meccanismo di comunicazione dove, il Sender trasferisce al Receiver i dati della sua F-UDT attraverso i blocchi funzioni relativi alle open user communication (*Send array e Rcv array*) ed il Receiver gli risponde con la conferma di ricezione (*ACK*).

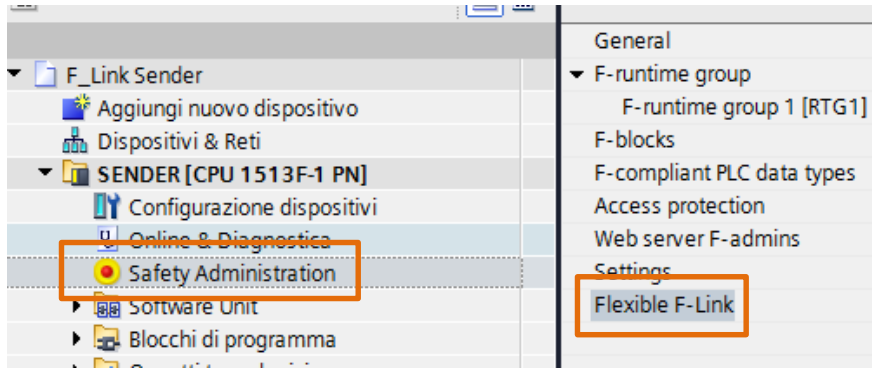


3.2.1.1. Programmazione lato SENDER

1. Creare una F-UDT: nella cartella dei *Tipi di dati PLC* (del PLC che sarà utilizzato come Sender), aggiungere un nuovo tipo di dati facendo attenzione a selezionare la proprietà *Crea tipi di dati PLC F adeguati*. Nella UDT appena aggiunta, inserire tutti i dati fail-safe che si vogliono scambiare, fino ad un massimo di 100 bytes



2. Creare la F-Communication: all'interno del Safety Administration nella cartella relativa al Flexible F-Link inserire la comunicazione sicura come mostrato in figura



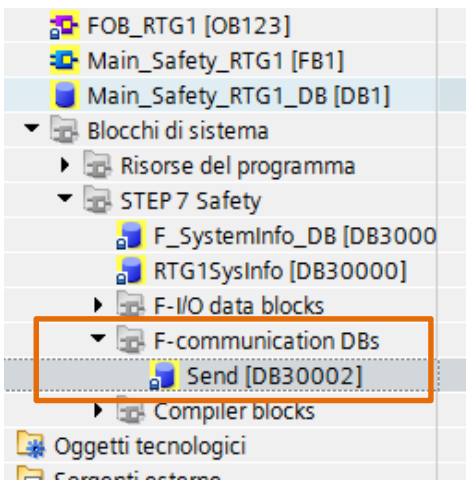
Variabili per l'invio dei dati
N.B. viene creato in automatico

Flexible F-Link settings

Name	PLC Data Type	Direction	F-monitoring time: (ms)	F-communication UUID	Output data variable	Input data tag
1 Send	Dati da inviare	Send	1000	fa226902-02ee-4832-8886-10c222e66b51	*Send*.SEND_ARRAY[]	*Send*_ACK_RCV_ARRAY[]
2 <Add new>						

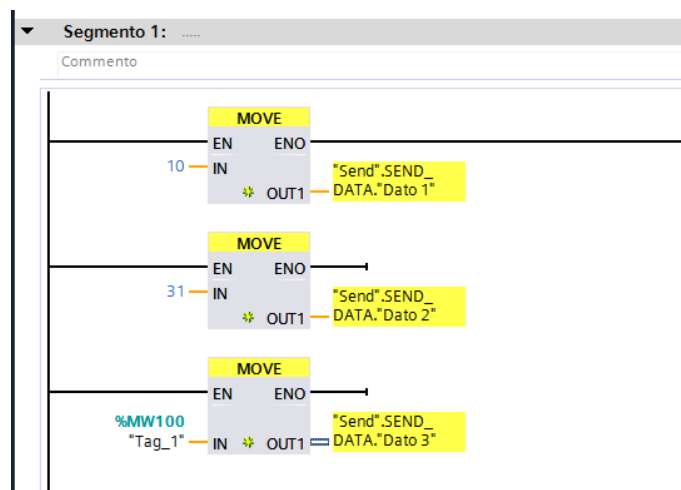
Inserire il nome del collegamento
 Richiamare la F-UDT
 Scegliere la direzione di comunicazione
 Selezionare F-monitoring time per la comunicazione
 Codice univoco che identifica la comunicazione fail-safe verso il receiver
N.B. bisogna copiarlo anche lato RECEIVER
 Variabili per l'acknowledgment
N.B. viene creato in automatico

3. Dichiarare le variabili da inviare: all'interno del Main Safety appoggiare il valore delle variabili ai dati inseriti nella F-UDT. Una volta creata la F-Communication come al punto 2, in automatico in TIA PORTAL viene generata sotto la cartella *Blocchi di sistema* → *Step7 Safety* → *F-communication DBs* una DB dove sono contenuti tutti i dati relativi alla comunicazione creata; in questa DB, come si può notare dall'immagine, sono presenti i dati della F-UDT

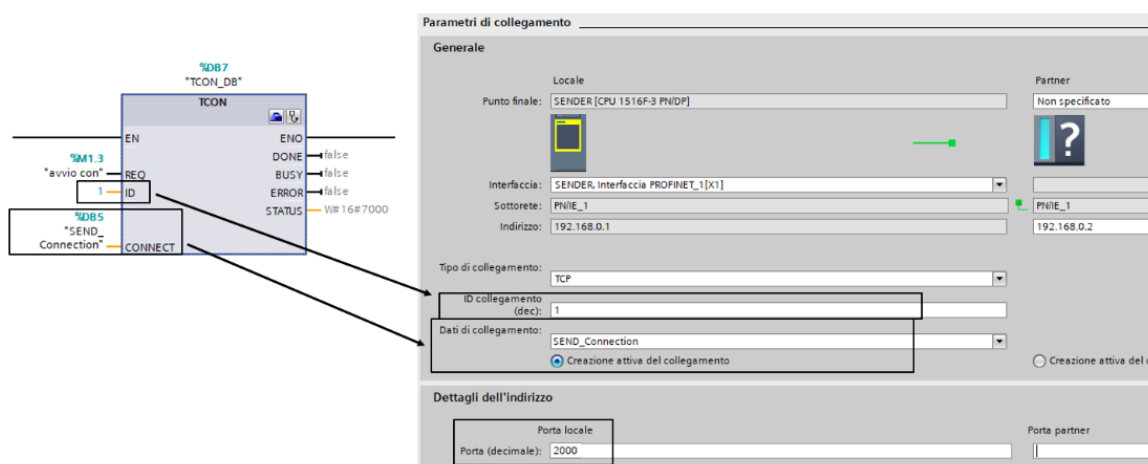


	Nome	Tipo di dati	Valore di avvio
1	Input		
2	SEND_DATA	*Safety send*	
3	ACK_RCV_ARRAY	Array[0..31] of Byte	
4	Output		
5	ERROR	Bool	false
6	ACTIVATE_FV	Bool	true
7	DIAG	Byte	16#0
8	SEND_ARRAY	Array[0..29] of Byte	
9	ACK_RCV_LENGTH	UInt	22
10	SEND_LENGTH	UInt	30
11	InOut		
12	Static		

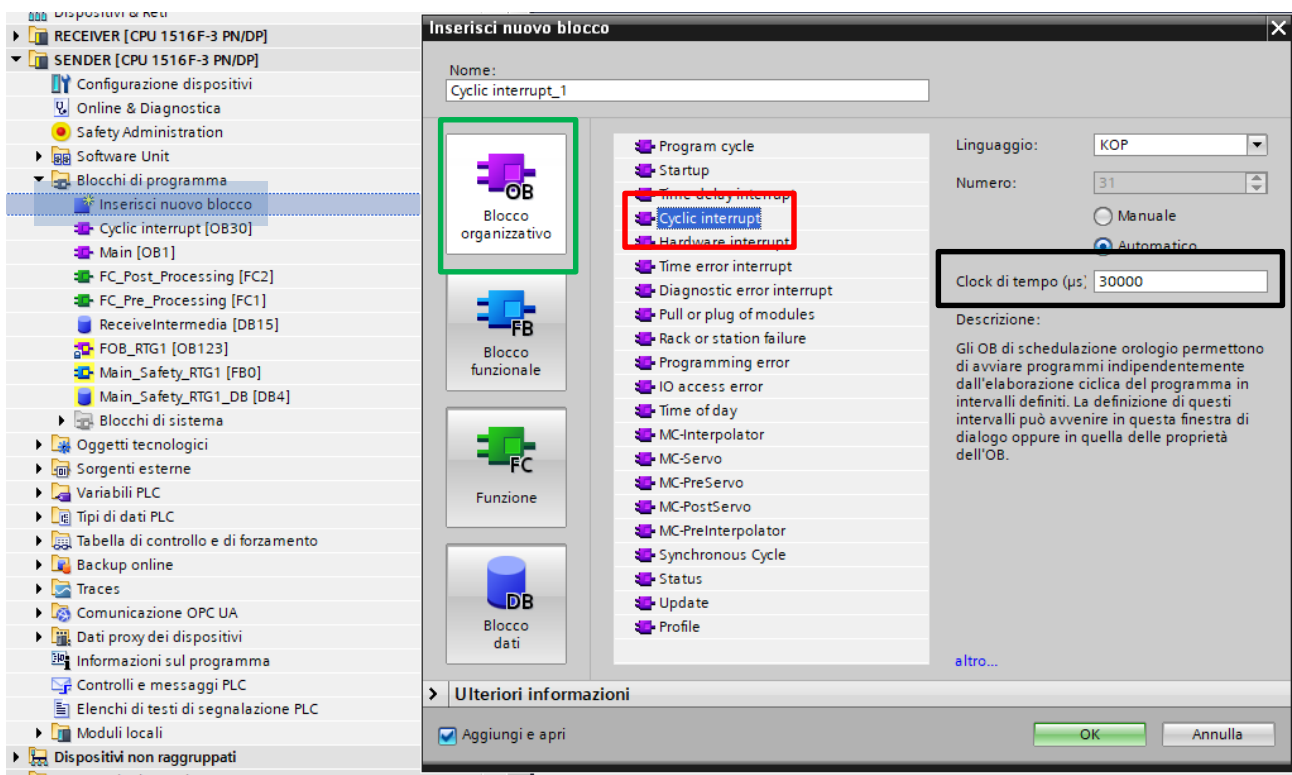
Per appoggiare i dati di questa DB nel Main Safety basta fare un semplice MOVE come riportato in figura



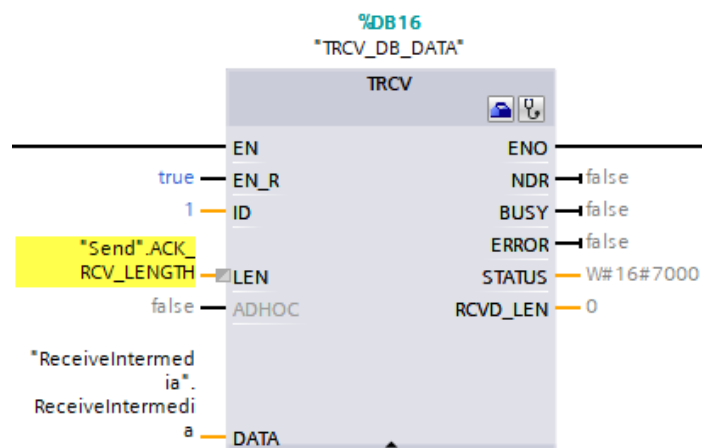
4. Creare la parte di programmazione per la comunicazione: all'interno del Main "standard" richiamare il blocco funzione *TCON* per inizializzare la comunicazione. Inserire il blocco *TCON* specificando con il parametro *ID* l'identificativo della comunicazione per lo scambio del Sender verso il Receiver ed in *Connect* i parametri di connessione; specificare la porta del partner come da regole per la comunicazione tramite le open user communication in TCP/IP;



5. Creare la parte di programmazione per la ricezione dei feedback di comunicazione (ACK): a tale scopo inserire un OB ciclico dove richiamare il blocco funzione *TRCV*; tale OB deve ciclare con un tempo molto più veloce della corrispondente *TSEND* implementata nel PLC Receiver (almeno 5 volte più veloce). Questo procedimento si rende necessario per evitare che nel PLC in ricezione si accumulino troppi dati in coda così da mandare in overflow la comunicazione (per maggiori dettagli consultare la guida presente sul support online di Siemens all'ID 109768964 capitolo 3.1). Per inserire un OB ciclico bisogna, sotto l'albero di navigazione del progetto nella cartella *Blocchi di programma*, cliccare su *Inserisci nuovo blocco* (riquadro blu nella figura seguente); nella finestra successiva cliccare su *Blocco organizzativo* (riquadro verde) e scegliere *Cyclic interrupt* (riquadro rosso); nella proprietà *Clock di tempo* inserire 30000µs (riquadro nero).
- N.B.** come spiegato in precedenza tale OB dura 30ms in quanto il corrispondente F runtime group dove verrà richiamata la *TSEND* relativa lato Receiver dura 150ms.



Una volta inserito l'OB ciclico, richiamare su di un segmento libero il blocco funzione *TRCV*, che si trova nell'albero delle Istruzioni → Comunicazione → Open user communication → Ulteriori; parametrizzarlo come mostrato nella figura seguente dove:



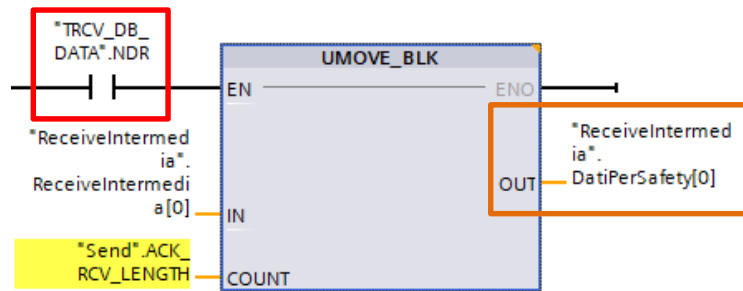
- *En_R* impostarlo a TRUE in modo da essere sempre in ascolto per la ricezione;
- *ID* deve essere lo stesso dichiarato nella TCON al punto 4;
- *LEN* rappresenta in byte la lunghezza dell'area di ricezione (nel caso specifico gli ACK), la variabile contenente la lunghezza dei dati da ricevere si trova sulla DB di istanza della comunicazione (*F-communication DBs*) come mostrato nella figura seguente;

Send			
	Nome	Tipo di dati	Valore di avvio
	▼ Input		
	▶ SEND_DATA	"Safety send"	
	▶ ACK_RCV_ARRAY	Array[0..21] of Byte	
	▼ Output		
	ERROR	Bool	false
	ACTIVATE_FV	Bool	true
	DIAG	Byte	16#0
	▶ SEND_ARRAY	Array[0..29] of Byte	
	▶ ACK_RCV_LENGTH	UInt	22
0	▶ SEND_LENGTH	UInt	30
1	InOut		
2	Static		

- *Data* inserire una variabile di appoggio per i dati ricevuti, tale variabile sarà un array di byte di 22 elementi come richiesto dalla variabile *ACK_RCV_LENGTH*;

	Nome	Tipo di dati
	▼ Static	
	▶ ReceiveIntermedia	Array[0..21] of Byte

Per concludere la programmazione dell'OB, inserire il blocco *UMOVE_BLK* (come mostrato nella figura seguente), per copiare in maniera consistente i dati ricevuti dalla TRCV su una variabile di una DB che verrà usata per scambiare i dati tra parte standard e parte safety. Come si può notare dall'immagine seguente, l'array di 22 elementi *ReceiveIntermedia.DatiPerSafety* (riquadro arancio in figura) verrà riempito nei suoi elementi ad ogni ricezione del singolo elemento nel ciclo della TRCV (riquadro rosso in figura).

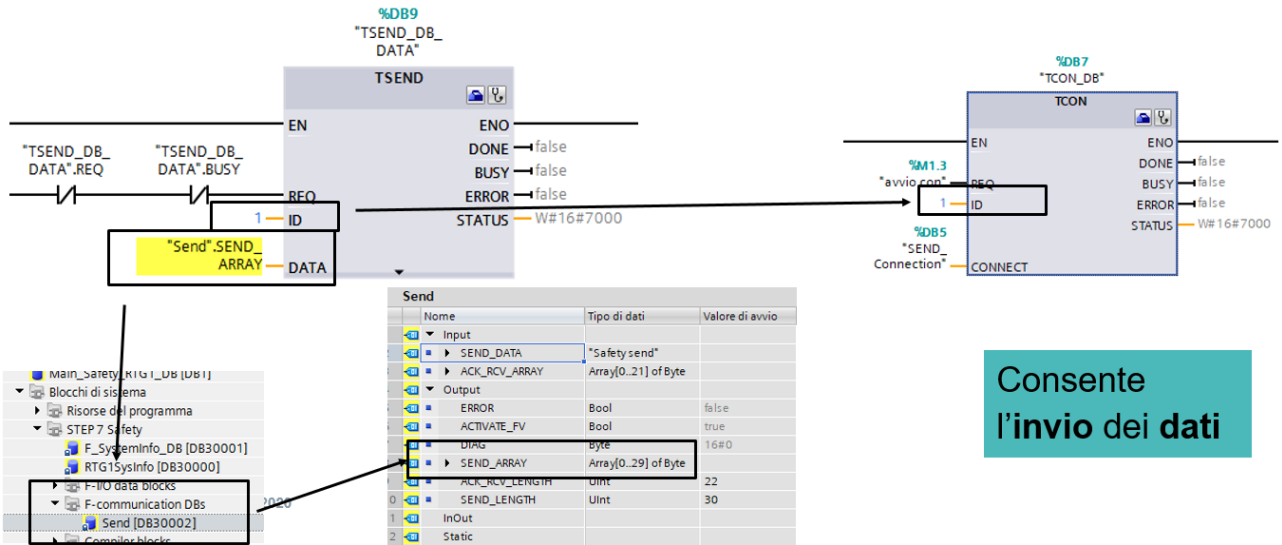


Il blocco funzione *UMOVE_BLK* è così parametrizzato:

- *IN* inserire il primo elemento dell'array dell'area sorgente da dove cominciare a copiare i dati;
- *COUNT* numero di elementi da copiare a partire dal primo elemento dell'array sorgente;
- *OUT* area di destinazione dove verranno copiati i dati.

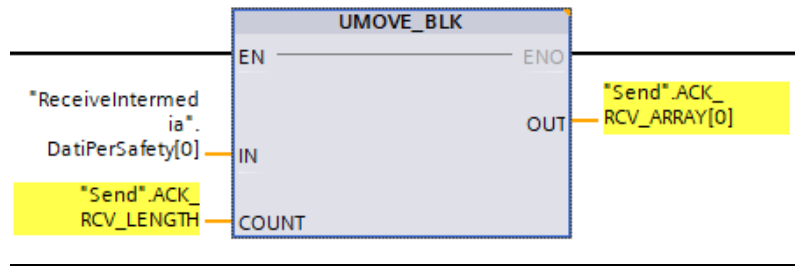
6. Creare le FC Pre-Processing e Post-Processing: al fine di richiamare dei blocchi standard direttamente nell'F-RunTime Group, bisogna creare delle FC specifiche che andranno richiamate all'interno del Safety Administration:

- FC *Post_Processing*: all'interno di questa FC inserire una TSEND relativa alla TCON per l'invio dei dati. Programmare la TSEND come mostrato in figura. Tramite la TSEND si inviano i dati safety presenti nella F-UDT al Receiver.



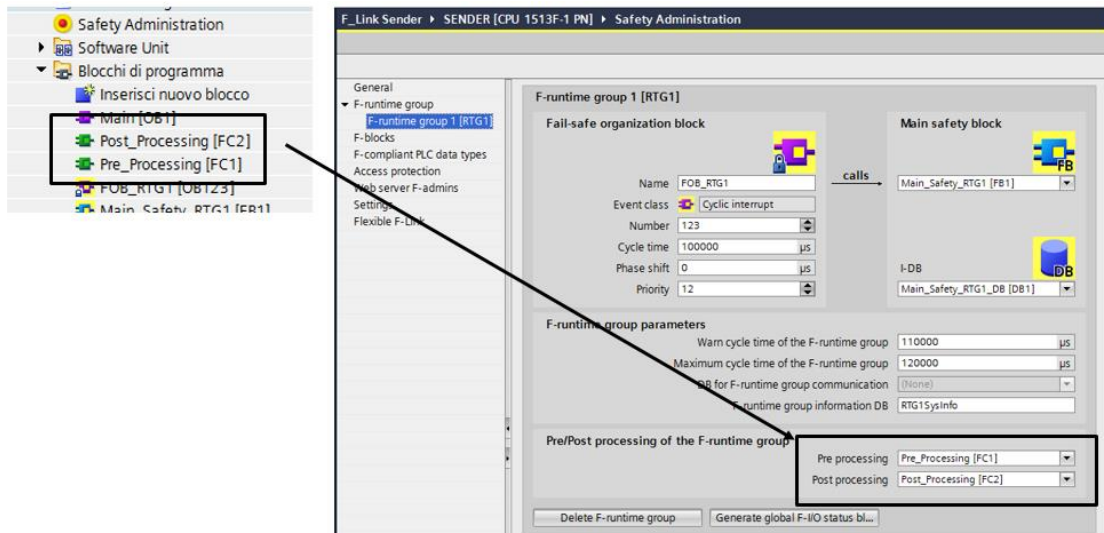
Consente l'invio dei dati

- FC Pre_Processing: all'interno di questa FC richiamare il blocco funzione *UMOVE_BLK* per copiare gli elementi dell'array *ReceiveIntermidia.DatiPerSafety* (creato al punto 5) sull'array di ricezione degli ACK presente nella *F-communication DBs* → *Send*



N.B. all'interno delle FC di Pre_Processing e Post_Processing non è consentito l'utilizzo di merker, ma solo di dati locali o di costanti

7. Associare le FC all'F-Runtime Group: come mostrato in figura, una volta create le FC come descritto nei punti precedenti, associarle all'F-Runtime Group sotto la voce *Pre/Post processing of the F-Runtime Group*.



3.2.1.2. Programmazione lato RECEIVER

1. Creare una F-UDT: come descritto per il Sender, anche per il Receiver bisogna creare una F-UDT che verrà utilizzata per ricevere i dati che vengono inviati dal Sender.
2. Creare la F-Communication: come mostrato già per il Sender, all'interno del Safety Administration → Flexible F-Link, bisogna creare la comunicazione dal Receiver verso il Sender.

Dalla figura seguente si può notare che, in questo caso la *Direction* è di tipo *Receive* e che *F-communication UUID* deve essere esattamente uguale a quello impiegato nel Sender.

Variabili per l'acknowledgment
N.B. viene creato in automatico

Name	PLC Data Type	Direction	F-monitoring ti...	F-communication UUID	Output data variable	Input data tag
1 RCV	Dati da ricevere	Receive	1000	fa226902-02ee-4832-8886-10c222e66b51	"RCV".ACK_SEND_ARRAY[]	"RCV".RCV_ARRAY[]
2 <Add new>						

Inserire il nome del collegamento

Richiamare la F-UDT

Scegliere la direzione di comunicazione

Selezionare F-monitoring time per la comunicazione

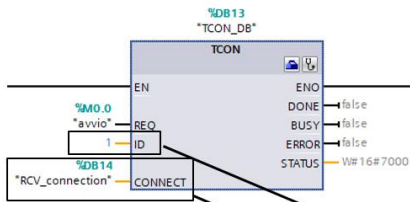
Codice univoco che identifica la comunicazione fail-safe verso il sender
N.B. deve essere identico a quello del SENDER

Variabili per la ricezione dei dati
N.B. viene creato in automatico

3. Dichiarare le variabili da inviare: anche nel caso del Receiver, una volta creata la comunicazione, viene generata una DB dove sono contenuti tutti le variabili che saranno utilizzate dal sistema per lo scambio dati. Nel Main Safety del Receiver, tramite dei MOVE è possibile leggere le variabili che saranno inviate dal Sender.

Nome	Tipo di dati	Valore di avvio
1 Input		
2 PASS_ON	Bool	false
3 ACK_REI	Bool	false
4 RCV_ARRAY	Array[0..27] of Byte	
5 Output		
6 RCV_DATA	"Dati da ricevere"	
7 Dato ricevuto 1	Int	0
8 Dato ricevuto 2	Int	0
9 Dato ricevuto 3	Int	0
10 ERROR	Bool	false
11 PASS_OUT	Bool	true
12 ACK_REQ	Bool	false
13 SENDMODE	Bool	false
14 DIAG	Byte	16#0
15 ACK_SEND_ARRAY	Array[0..21] of Byte	
16 RCV_LENGTH	UInt	28
17 ACK_SEND_LENGTH	UInt	22
18 InOut		
19 Static		

4. Creare la parte di programmazione per la comunicazione: nel Main "standard" inserire una TCON per stabilire la comunicazione con il PLC Sender. Parametrizzare la TCON come mostrato nella figura seguente: valgono le stesse regole descritte per il Sender al punto 4 del capitolo 3.2.1.1.



Parametri di collegamento

Generale

Locale: Punto finale: RECEIVER [CPU 1516F-3 PN/DP], Interfaccia: RECEIVER, Interfaccia PROFINET_1[X1], Sottorete: PN1E_1, Indirizzo: 192.168.0.2

Partner: Non specificato, 192.168.0.1

Tipo di collegamento: TCP

ID collegamento (dec): 1

Dati di collegamento: RCV_connection

Dettagli dell'indirizzo

Porta locale: 2000

5. Creare la parte di programmazione per la ricezione dei dati safety provenienti dal Sender: come già mostrato per la programmazione lato Sender, anche nel caso del Receiver, per evitare problemi di comunicazione in ricezione, inserire la TRCV in un OB ciclico con un tempo che sia almeno 5 volte più veloce di quello della relativa TSEND per l'invio dei dati implementata nel Sender (punto 6 capitolo 3.2.1.1). Valgono le stesse considerazioni fatte al punto 5 del capitolo 3.2.1.1 per la programmazione del Sender. In particolare inserire un OB ciclico che duri 30ms e su di un segmento libero dello stesso inserire una TRCV programmata come da figura seguente

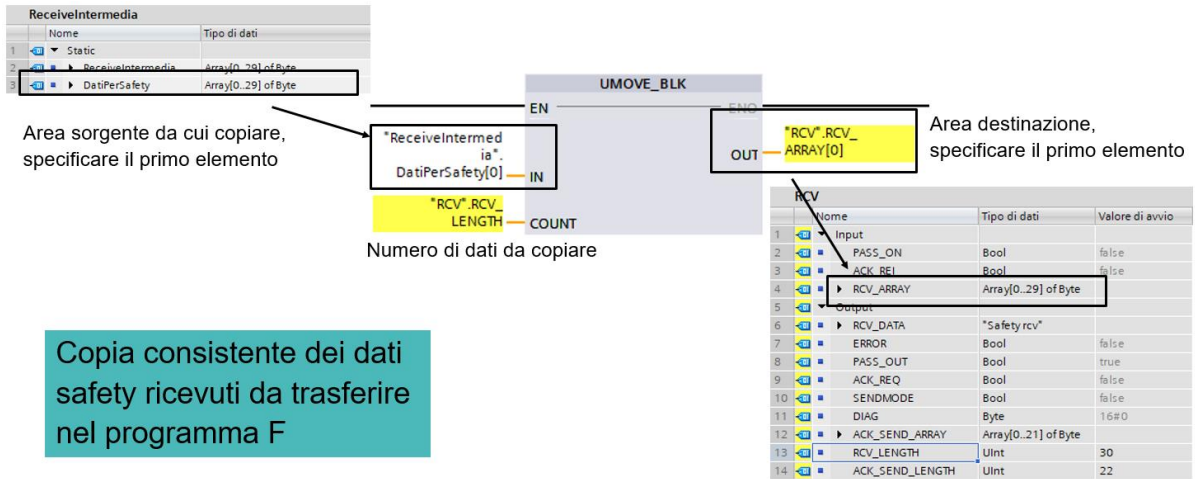
Nome	Tipo di dati	Valore di avvio
1	Input	
2	PASS_ON	Bool
3	ACK_REI	Bool
4	RCV_ARRAY	Array[0..29] of Byte
5	Output	
6	RCV_DATA	"Safety rcv"
7	ERROR	Bool
8	PASS_OUT	Bool
9	ACK_REQ	Bool
10	SENDMODE	Bool
11	DIAG	Byte
12	ACK_SEND_ARRAY	Array[0..21] of Byte
13	RCV_LENGTH	UInt
14	ACK_SEND_LENGTH	UInt

Nome	Tipo di dati
1	Static
2	ReceiveIntemdia
3	DatiPerSafety

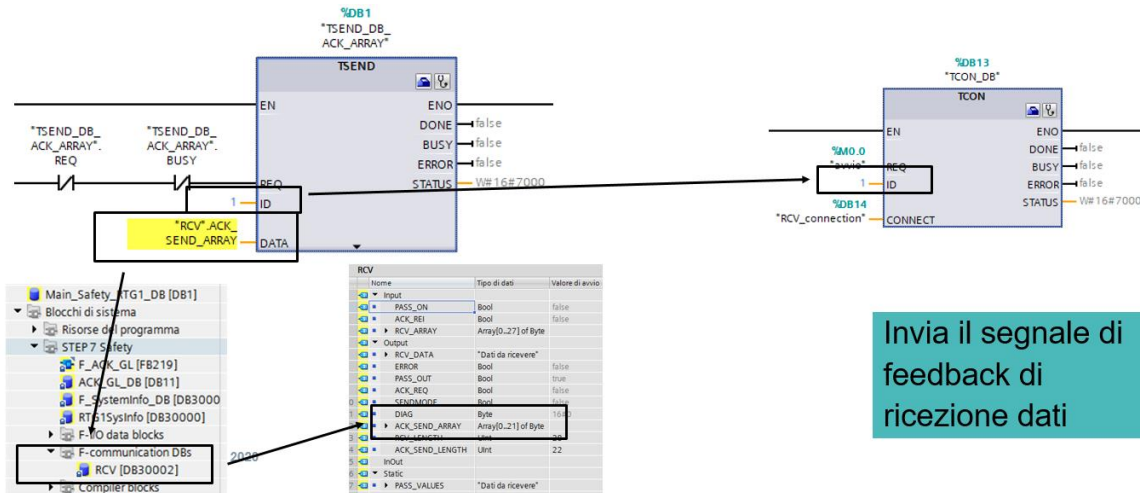
e, sempre come mostrato in figura, inserire il blocco UMOVE_BLK per copiare i dati in maniera coerente sulla DB di appoggio ReceiveIntemdia, che poi sarà utilizzata per scambiare i dati tra la parte standard e la parte safety.

6. Creare le FC Pre-Processing e Post-Processing:

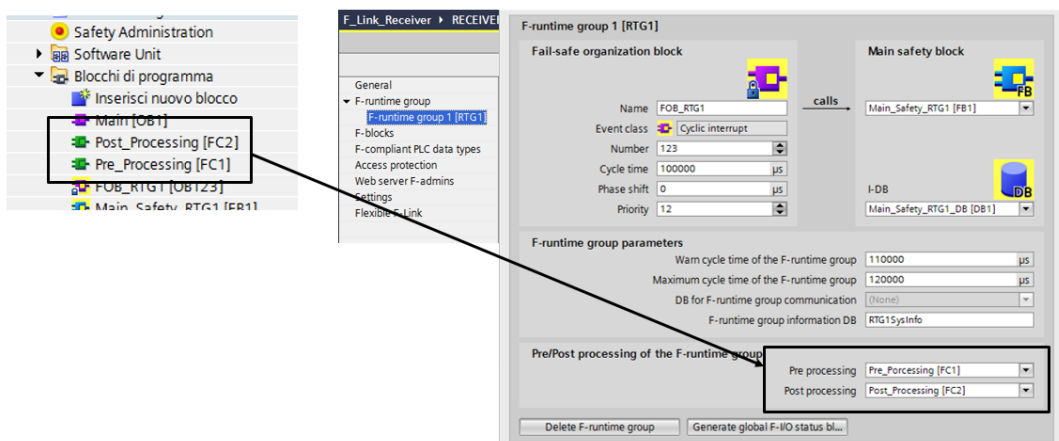
- FC Pre_Processing: all'interno di questa FC richiamare il blocco funzione UMOVE_BLK per copiare gli elementi dell'array ReceiveIntemdia.DatiPerSafety (creato al punto 5) sull'array di ricezione dei dati presente nella F-communication DBs → Send



- FC Post_Processing: all'interno di questa FC inserire una TSEND relativa alla TCON per l'invio degli ACK. Programmare la TSEND come mostrato in figura. Tramite la TSEND si inviano i feedback di avvenuta ricezione dei dati al Sender.

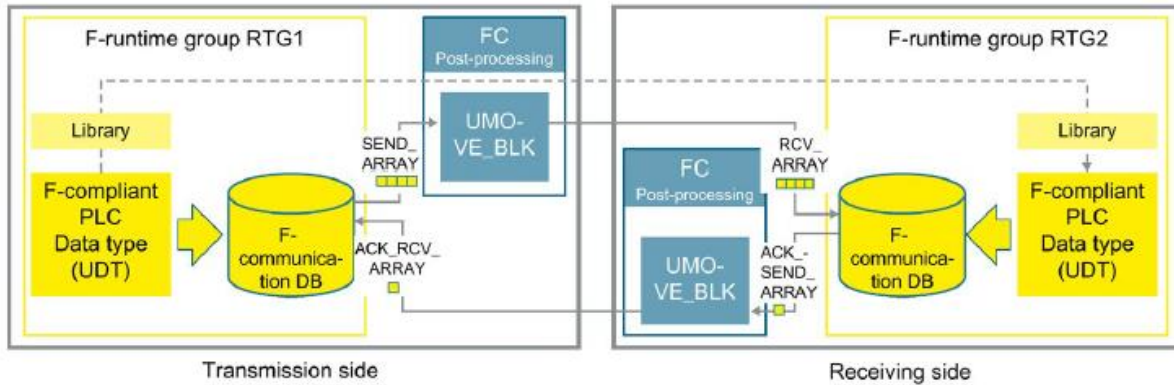


7. Associare le FC all'F-Runtime Group: come mostrato in figura, una volta create le FC come descritto nei punti precedenti, associarle all'F-Runtime Group sotto la voce *Pre/Post processing of the F-Runtime Group*.



3.2.2. Flexible F-Link per F-Runtime Group communication

Sfruttando i concetti esposti nel capitolo precedente, è possibile far comunicare tra di loro due F-Runtime Group diversi appartenenti ad una stessa F-CPU. Il principio di funzionamento si basa sul fatto che, delle F-UDT (max 100 bytes) possono essere trasferite da un F-Runtime Group ad un altro tramite istruzioni standard come *UMOVE_BLK* (come mostrato in figura)

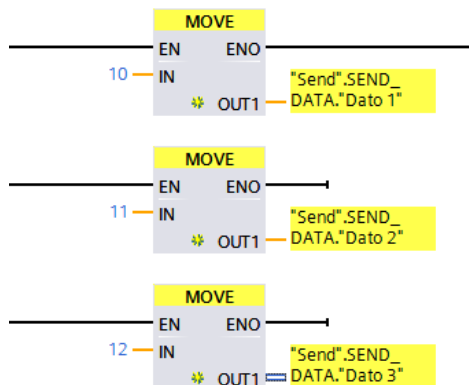


Per effettuare tale comunicazione seguire i seguenti passaggi:

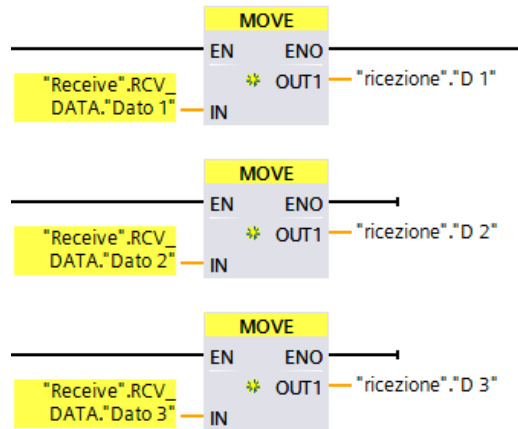
1. Creare una F-UDT che verrà usata per scambiare i dati
2. Nel Safety Administration → Flexible F-Link, creare due F-communication una per la send ed una per la receive come mostrato in figura, stando attenti ad utilizzare lo stesso F-monitoring time e F-communication UUID

General		Flexible F-Link settings					
Name	PLC Data Type	Direction	F-monitoring ti...	F-communication UUID	Output data variable	Input data tag	
1 Send	DATA	Send	500	b2c82ed2-39a6-432d-be72-fcd2e847ad1	"Send".SEND_ARRAY[]	"Send".ACK_RCV_ARRAY[]	
2 Receive	DATA	Receive	500	b2c82ed2-39a6-432d-be72-fcd2e847ad1	"Receive".ACK_SEND_ARRAY[]	"Receive".RCV_ARRAY[]	
3	<Add new>						

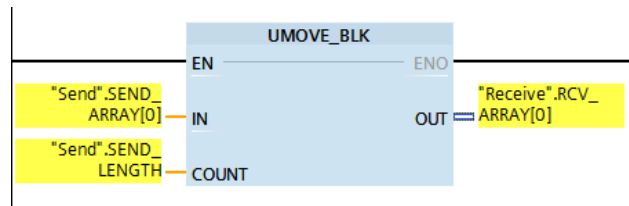
3. Lato Sender (ad esempio RGT1) inserire sulla DB di send i dati da inviare; fare questo passaggio nel Main Safety relativo al RGT1 come mostrato in figura



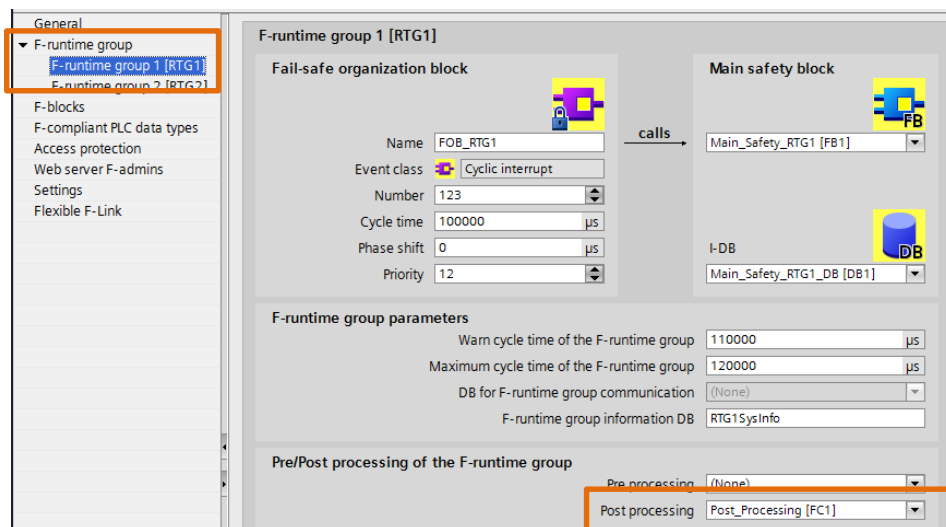
- Lato Receiver (ad esempio RGT2) leggere i dati ricevuti spostandoli con delle MOVE come mostrato in figura; fare questo passaggio nel Main Safety relativo al RGT2



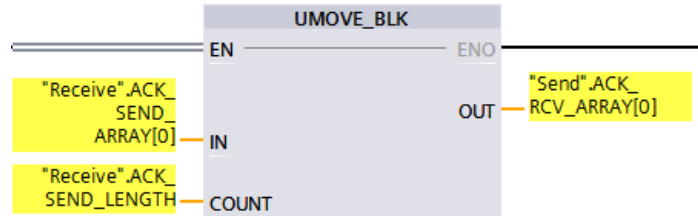
- FC Post_Processing lato Sender: creare una FC di Post_Processing lato Sender da richiamare nell'F-Runtime Group per l'invio dei dati (ad esempio RGT1) ed all'interno, programmare il blocco *UMOVE_BLK* come mostrato in figura



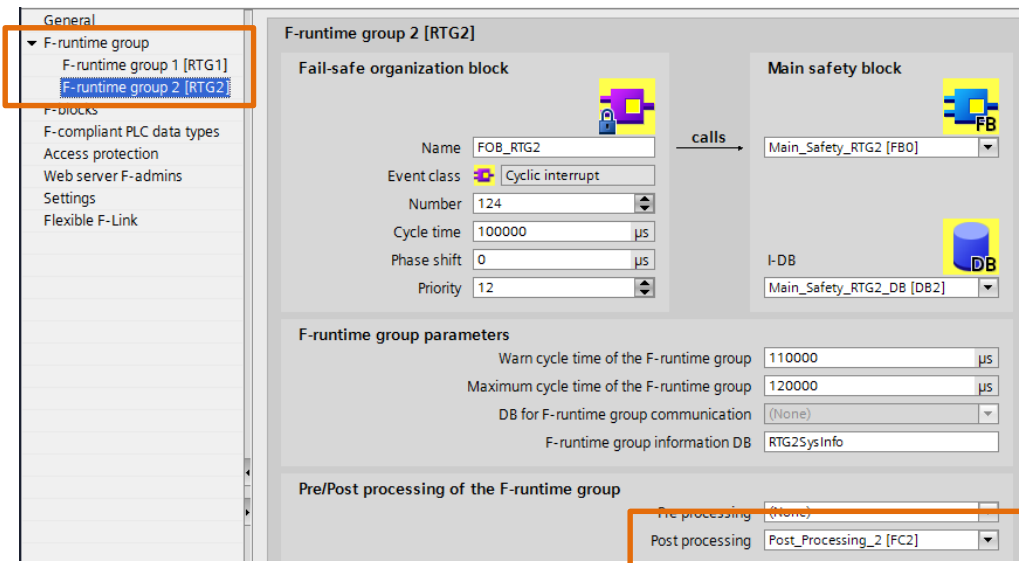
Nel Safety Administration associare tale FC all'F-Runtime Group RGT1 come mostrato in figura



- FC Post_Processing lato Receiver: creare una nuova FC di Post_Processing lato Receiver da richiamare nell'F-Runtime Group per il feedback di ricezione dati (ad esempio RGT2) ed all'interno, programmare il blocco *UMOVE_BLK* come mostrato in figura



Nel Safety Administration associare questa nuova FC all'F-Runtime Group RGT2 come da figura



3.2.3. Reintegrazione a seguito di errori di comunicazione

La gestione della reintegrazione a seguito del verificarsi di errori di comunicazione è deputata al Receiver. All'interno della DB che viene creata in automatico quando si crea la comunicazione Flexible F-Link nel Safety Administration, si trovano tutti i dati per la gestione della reintegrazione.

Come si può notare dall'immagine seguente, al verificarsi di un fault è possibile sapere lo stato della comunicazione controllando le variabili *ERROR*, *PASS_OUT*, *ACK_REQ* e *DIAG*. Per reintegrare basta porre a TRUE il bit *ACK_REI*. Quando si verifica un fault, al fine di mantenere la sicurezza, le variabili utilizzate dallo scambio dati vengono passivate ed il Receiver usa quelle che sono state impostate in *PASS_VALUES*.

Main_Safety_RTG1_DB [DB1]
Blocchi di sistema
Risorse del programma
STEP 7 Safety
F_ACK_GL [FB219]
ACK_GL_DB [DB11]
F_SystemInfo_DB [DB3000]
RTG1SysInfo [DB30000]
F-I/O data blocks
F-communication DBs
RCV [DB30002]
Compiler blocks

RCV				
	Nome	Tipo di dati	Valore di avvio	Valore di controllo
1	Input			
2	PASS_ON	Bool	false	FALSE
3	ACK_REI	Bool	false	FALSE
4	RCV_ARRAY	Array[0..29] of Byte		
5	Output			
6	RCV_DATA	*Safety rcv*		
7	ERROR	Bool	false	TRUE
8	PASS_OUT	Bool	true	TRUE
9	ACK_REQ	Bool	false	FALSE
10	SENDMODE	Bool	false	TRUE
11	DIAG	Byte	16#0	16#10
12	ACK_SEND_ARRAY	Array[0..21] of Byte		
13	RCV_LENGTH	UInt	30	30
14	ACK_SEND_LENGTH	UInt	22	22
15	InOut			
16	Static			
17	PASS_VALUES	*Safety rcv*		
18	Dato 1	Int	0	0
19	Dato 2	Int	0	0
20	Dato 3	Int	0	0
21	Dato 4	Int	0	0

Per ulteriori informazioni visita il sito:

<https://new.siemens.com/it/it/prodotti/automazione.html>

I dati tecnici presentati in questo documento si basano su un caso di utilizzo reale o su parametri progettuali, pertanto non è possibile fare affidamento a essi per qualsivoglia applicazione specifica e non costituiscono garanzia di prestazioni per qualsiasi progetto.

I risultati effettivi dipendono da una serie di condizioni variabili. Di conseguenza, Siemens non emette alcuna rappresentanza, garanzia, assicurazione in relazione all'accuratezza, vigenza o completezza dei contenuti riportati nel presente documento. Su richiesta, verranno forniti dati tecnici specifici oppure specifiche riguardanti applicazioni particolari del cliente. L'azienda lavora continuamente nell'ingegnerizzazione e nello sviluppo. Per tale ragione, si riserva il diritto di apportare modifiche in qualsiasi momento alla tecnologia e alle specifiche del prodotto contenute nel presente documento.