



POLITECNICO DI TORINO

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Meccanica

Monografia di Prova Finale

**VALUTAZIONE DI SOLUZIONI SAFETY  
PER ASSI INTERPOLATI (ROBOT)  
SUL SISTEMA D'AUTOMAZIONE MG**

Tesi presso l'azienda

**MARCHESINI GROUP S.P.A.**

**Candidato**

FABIO MANERA

**Relatore**

Prof. MASSIMO SORLI

**Correlatori**

Ing. STEFANO GUALMINI

Ing. ALBERTO MARTINI

Anno Accademico 2017/2018

*Desidero ringraziare il professore Sorli e  
gli ingegneri Gualmini e Martini  
per avermi supportato nell'elaborazione della tesi e  
per la disponibilità dimostratami.*

*Ringrazio la mia famiglia per aver contribuito  
al raggiungimento di questo traguardo,  
supportandomi sia moralmente che economicamente.*

*Ringrazio anche tutti i parenti e gli amici  
che mi sono stati vicini in questi anni.*



# INDICE

1. INTRODUZIONE .....	6
1.1 Struttura aziendale .....	6
1.2. Oggetto di studio .....	12
2. SISTEMA MARCHESINI .....	13
2.1 MPS2000.....	13
2.2 MAC2000 .....	17
2.2.1. Comunicazioni tra PC e azionamenti.....	18
2.2.2. Interfaccia MAC2000.....	19
2.3. SAFETY .....	20
2.3.1. PLC .....	21
2.3.2. Sensori.....	22
2.3.3. Attuatori .....	22
3. PIATTAFORMA DI SVILUPPO TWINCAT .....	23
3.1. Configurare un progetto TwinSAFE.....	27
3.1.1. Menù PLC .....	27
3.1.2. Menù I/O .....	29
3.1.3. Menù SAFETY .....	32
3.2. Progetto SAFETY .....	35
4. BECKHOFF E BUS DI CAMPO .....	39
4.1. Rete di trasmissione dati EtherCAT .....	39
4.2. IO LINK.....	40
4.3. Banco prova .....	40
5. BLOCCHI FUNZIONALI.....	42
5.1. AND.....	47
5.2. OR.....	48
5.3. OPMODE.....	49
5.4. ESTOP .....	51
5.5. MON .....	53
5.6. DECOUPLE.....	55
5.7. TWOHAND .....	56
5.8. MUTING.....	58
5.9. EDM.....	61
5.10. RS.....	63

5.11.	SR.....	64
5.12.	TON .....	65
5.13.	TOF .....	66
5.14.	CONNECTION SHUTDOWN.....	67
5.15.	ADD.....	68
5.16.	SUB.....	70
5.17.	MUL.....	72
5.18.	DIV .....	74
5.19.	COMPARE .....	76
5.20.	LIMIT .....	78
5.21.	COUNTER.....	80
5.22.	SCALE .....	82
5.23.	SPEED .....	84
5.24.	LOADSENSING.....	86
5.25.	CAMMONITOR.....	89
5.25.1.	CAMMONITOR (Pendulum Mode) .....	89
5.25.2.	CAMMONITOR (Excentric Mode) .....	93
5.25.3.	Descrizione del processo .....	97
5.26.	SLI.....	98
5.27.	ENVELOPE .....	100
5.28.	VIOLATIONCNT.....	103
6.	ESEMPI APPLICATIVI.....	105
6.1.	MUTING.....	105
6.2.	LOADSENSING.....	111
6.3.	CAMMONITOR (Excentric Mode) .....	113
6.4.	COMPARE, LIMIT, MON.....	117
7.	CONCLUSIONI .....	118
8.	Bibliografia .....	119

# 1. INTRODUZIONE

La tesi è stata svolta presso Marchesini Group S.p.A.

L'azienda è stata fondata nel 1974 da un'idea di Massimo Marchesini che ha proposto al settore manifatturiero l'automatizzazione di operazioni di packaging fino ad allora svolte manualmente. Il mercato ha accolto fin da subito le proposte innovative dell'imprenditore e questo ha contribuito ad una rapida crescita ed espansione dell'azienda. Prima di tutte ad esser prodotta è stata un'astucciatrice intermittente; da questa successivamente sono nate e si sono sviluppate nuove macchine adatte a soddisfare le esigenze di una clientela sempre in espansione che spazia dal mondo farmaceutico a quello della cosmesi.



Figura 1: ingresso principale dello stabilimento Marchesini Group di Pianoro

Agli inizi degli anni '80 le macchine erano fortemente improntate sulla meccanica mentre la parte di gestione e controllo realizzata dagli elementi elettronici erano pressoché assenti, in quanto la tecnologia dell'epoca non lo permetteva. Con il passare del tempo, grazie allo sviluppo dei PC, la tecnologia è entrata prepotentemente nella vita di tutti i giorni così come all'interno delle realtà aziendali, in particolare in quella della Marchesini Group, permettendo ai propri tecnici di sviluppare macchine automatiche sempre più prestazionali.

Per rispondere alle esigenze di un mercato in continua evoluzione si sono sviluppate macchine e linee produttive sempre più veloci e performanti, in grado di rispondere alle più svariate richieste di personalizzazione del cliente, sempre però senza trascurare i concetti fondamentali di qualità e di sicurezza del prodotto. Questo ha portato l'azienda ad esser presente ed operare in oltre 60 Paesi nel mondo, sia appartenenti alla Comunità Europea che extra UE.

## 1.1. Struttura aziendale

L'azienda con il passare degli anni si è ampliata sempre di più, acquisendo diverse aziende con le quali ha istaurato rapporti di stretta collaborazione; alcune di queste

sono state pienamente inglobate altre, invece, sono entrate a far parte di Marchesini Group S.p.A. mantenendo una certa autonomia sia economica che fiscale. Così facendo, si è riusciti a formare un gruppo funzionale di aziende che lavorano in sinergia tra loro, pronto a soddisfare al meglio e sempre più rapidamente le esigenze della clientela.

Oggigiorno, infatti, il gruppo è costituito dalle seguenti divisioni (che in passato erano aziende autonome):

- **Blister Division** (Carpi, MO) specializzati in macchine Blisteratrici;
- **Farcon Division** (Carpi, MO) specializzati in macchine Termoformatrici;
- **Packservice Division** (Latina, LT) specializzati in macchine Cartonatrici;
- **Tonazzi-Vasquali Division** (Latina, LT) specializzati in macchine Tubettatrici;
- **Neri S.p.a.** (Barberino del Mugello, FI) specializzata in macchine Etichettatrici.

Vi è poi anche una stretta collaborazione con un'azienda, tutt'ora autonoma dal punto di vista fiscale, ma fortemente legata al gruppo. Tale azienda è:

- **Corima S.r.l.** (Monteriggioni, SI) specializzata in macchinari per Sterili.

Le attività di stage e di tesi sono state svolte nella sede principale del gruppo, ovvero quella di Pianoro (BO): è proprio in questo piccolo paese sorto in prossimità dei colli bolognesi che la famiglia Marchesini, fondatrice del gruppo, ha posto le fondamenta per costruire quella che oggi è diventata un'affermata realtà aziendale nel mondo dell'automazione.

Lo stabilimento è composto da più capannoni che nel corso degli anni sono stati costruiti oppure acquistati da aziende limitrofe. Questi ospitano sia le linee produttive costruite in azienda, ma anche i magazzini di materie prime e di semilavorati e tutti gli uffici tecnici e commerciali.

I capannoni produttivi sono rappresentati con una lettera dell'alfabeto e sono adibiti ad ospitare differenti tipologie di macchine automatiche. Di seguito vengono riportati una suddivisione dei siti produttivi e una piantina dello stabilimento (Figura 2).

- **Capannone A:** Linee (M0);
- **Capannone B:** Linee (M0);
- **Capannone C:** Liquidi (M6), Bustinatrici (M2), Fardellatrici (M9), Cartonatrici (M8);
- **Capannone L e N:** Astucciatrici (M1 & M3), Speciali (M7), Etichettatrici (N3).

## LAY OUT DI STABILIMENTO SEDE VIA NAZIONALE



Figura 2: piantina dello stabilimento di Pianoro

Le varie tipologie di macchine che vengono progettate, costruite e commercializzate da Marchesini Group S.p.A., sono per la maggior parte costruite e assemblate nella sede di Pianoro e sono di svariate tipologie, ovvero:

Accessories	Miscellaneous
Aereosol testing	Overwrappers
Assembling Machines	Pallettizers
Blister Thermoformers	Rigid tube fillers/cappers
Cartoner, horizontal type	Sachet Fillers
Cartoner, vertical type	Soft tube fillers / cappers
Casepackers	Special Machines
Counting machines	Stickpack
Depackers/Unwrappers	Stretch/Srink wrappers
Jar fillers/capper	Strip packers
Liquid fillers/cappers	Tray former
Mascara fillers/cappers	Turntable & Feeders

Tabella 1: tipologie di macchine automatiche

Le tipologie di macchine sono identificate da un codice Mx, dove x è un numero compreso tra 1 e 9 che rappresenta la tipologia e la dislocazione all'interno dei vari capannoni dell'azienda nello stabilimento di Pianoro. Tale numero identificativo permette di riconoscerle in maniera rapida e precisa:

M1 → Astucciatrici alterne

M2 → Bustinatrici

M3 → Astucciatrici continue

M4 → Blisteratrici

M5 → Astucciatrici verticali (rare e alcune in via di sviluppo)

M6 → Liquidi

M7 → Speciali

M8 → Cartonatrici – Cartopallettizzatori

M9 → Fascettatrici – Fardellatrici



Figura 3: cartonatrice

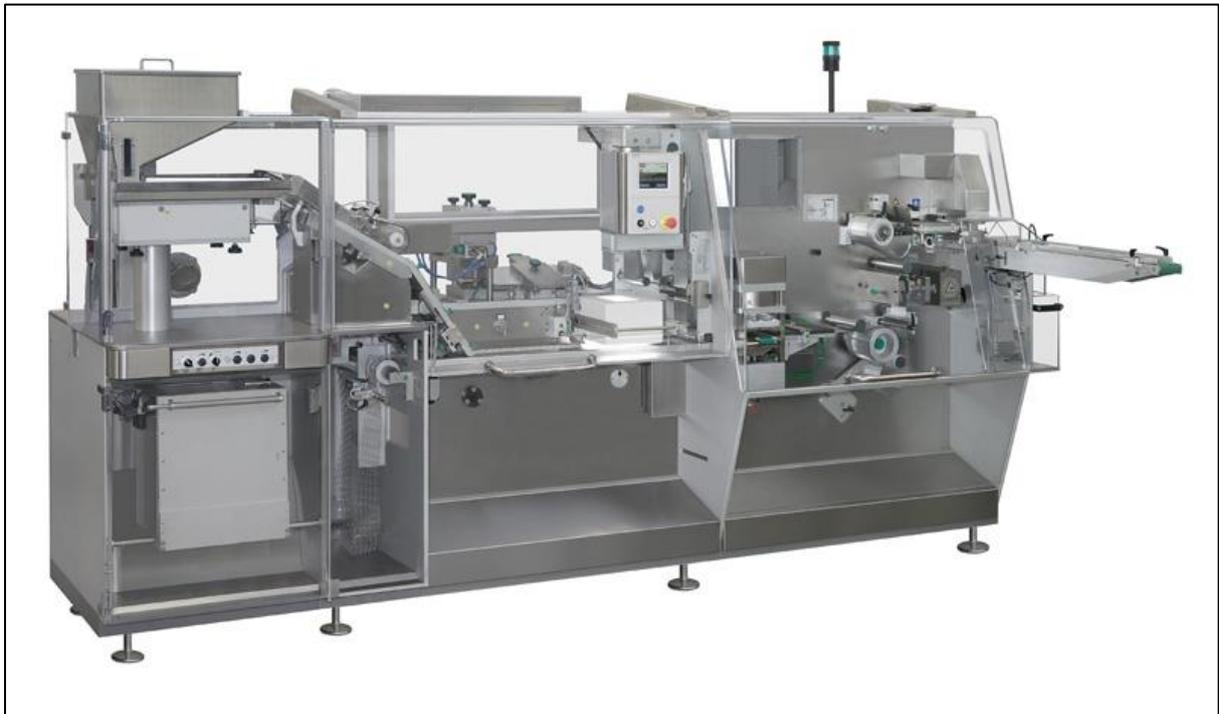


Figura 4: blisteratrice



Figura 5: dettaglio della blisteratrice



Figura 6: dettaglio della blisteratrice

## 1.2. Oggetto di studio

La tesi si propone di affrontare la realtà aziendale, presentando una panoramica generale della struttura della fabbrica. Partendo dalla programmazione dei PC, si passa in seguito alla presentazione delle tematiche riguardanti il controllo assi delle macchine automatiche e infine terminando con l'approfondimento del mondo Safety dei PLC e tutto ciò che riguarda la sicurezza.

La relazione si concentrerà in modo particolare su quest'ultimo punto, in quanto l'azienda da poco tempo ha deciso di innalzare il livello di sicurezza delle proprie macchine, passando dalla categoria di sicurezza CAT 2 alla categoria di sicurezza CAT 3, secondo la classificazione delle normative europee. La CAT 2 prevede che le informazioni dei dispositivi utilizzati a bordo macchina vengano trasmesse tramite singolo cavo, mentre la CAT 3 impone che le stesse informazioni vengano trasferite su due cavi. Questa miglioria permette di individuare nelle linee di trasmissione dati dei contatti e degli interruttori mal funzionanti. Ciò è possibile grazie alla logica del doppio contatto: una generica informazione rilevata sul campo dallo stesso sensore viene trasmessa su due canali differenti all'ingresso del PLC. Quest'ultimo verifica che i due segnali, oltre ad assumere lo stesso valore, arrivino con una certa sincronia: il programmatore deve impostare un certo tempo di discrepanza che rappresenta l'intervallo massimo che può intercorrere tra l'arrivo del primo e del secondo segnale elettrico. Se i segnali si discostano troppo tra loro oppure venisse violato il tempo di discrepanza, il PLC segnala un errore. Per capire meglio il concetto si propone un esempio.

Se per esempio si verificasse l'incollaggio di un contatto, nel caso si utilizzi una linea di trasmissione dati monocavo (CAT 2) il segnale elettrico in ingresso al modulo del PLC rimarrebbe costante ( ad esempio valore logico 0 oppure 1) e non sarebbe più rappresentativo della realtà fisica che si realizza all'interno della macchina. Inoltre, in tale situazione, non sarebbe possibile distinguere una situazione tra corretto e mal funzionamento e di conseguenza non sarebbe possibile segnalare un messaggio di errore macchina. Invece, se lo stesso problema si presentasse nel caso in cui si utilizzi una linea di trasmissione dati a doppio cavo, si avrebbe una diversa situazione: il segnale errato proveniente dal contatto incollato viene confrontato con il relativo segnale in parallelo (che si presuppone essere funzionante); non appena si rileva la discrepanza tra i due segnali, lo stato del PLC passa da quello di RUN (corretta esecuzione del programma) a quello di ERROR, bloccando quindi il processo produttivo, mettendo tutte le uscite in condizioni di sicurezza e segnalando l'errore di discrepanza.

Questo ha comportato una modifica sostanziale nella progettazione software e hardware delle macchine che l'azienda sta studiando e implementando grazie anche al proprio centro di ricerca e sviluppo R&D, all'interno del quale si studiano inoltre nuove e alternative soluzioni per le problematiche che si presentano ogni giorno all'interno della fabbrica.

## 2. SISTEMA MARCHESINI

La Marchesini Group S.p.A. è una azienda che, come detto in precedenza, produce macchine automatiche: questo implica sia la progettazione dell'hardware che quella del software. Per queste mansioni sono stati istituiti degli uffici appositi, identificati rispettivamente dalle sigle UTM (Ufficio Tecnico Meccanico) e UTE (Ufficio Tecnico Elettronico) nei quali lavora personale altamente specializzato. Questi uffici, lavorando in sinergia tra loro, provvedono allo sviluppo del prodotto: studiano approfonditamente ogni singola fase dello sviluppo, partendo dall'idea iniziale fino alla certificazione e all'installazione in loco della linea completa, per poi continuare anche dopo la vendita con i servizi di assistenza al cliente.

Nel corso degli anni, la Marchesini Group ha fatto la scelta di gestire al proprio interno sia la programmazione dei PC situati a bordo macchina che il controllo assi delle macchine automatiche. È stato scelto di intraprendere questo cammino al fine di massimizzare la qualità e la flessibilità del proprio prodotto. Così facendo, l'azienda possiede uno strumento di cui ha il pieno controllo: può modificarlo intervenendo laddove vengano riscontrate carenze e aggiornarlo costantemente a seconda delle proprie esigenze. Tutti gradi di libertà che non sarebbero gestibili qualora si fosse acquistato un software da un ente terzo.

Seguendo questa filosofia, nel corso degli anni la MG ha sviluppato un proprio software di programmazione frutto dell'evoluzione del knowhow aziendale:

*“Marlin 3.0”* (Mar = Marchesini + Lin = Linux)

Il linguaggio per la programmazione PC utilizzato in Marchesini Group è *“MPS2000”* (Marchesini Programming System), mentre per la programmazione assi si usa *“MAC2000”* (Marchesini Axis Control).

### 2.1. MPS2000

Lo sviluppo di programmi richiede strumenti di lavoro adeguati alle esigenze del sistema; l'editor è parte fondamentale di un sistema di sviluppo poiché costituisce l'interfaccia con il programmatore. L'editor del sistema MPS2000 si basa su uno strumento commerciale, modificato e personalizzato secondo le esigenze del linguaggio.

Uno screenshot della schermata di lavoro è mostrato in Figura 7.

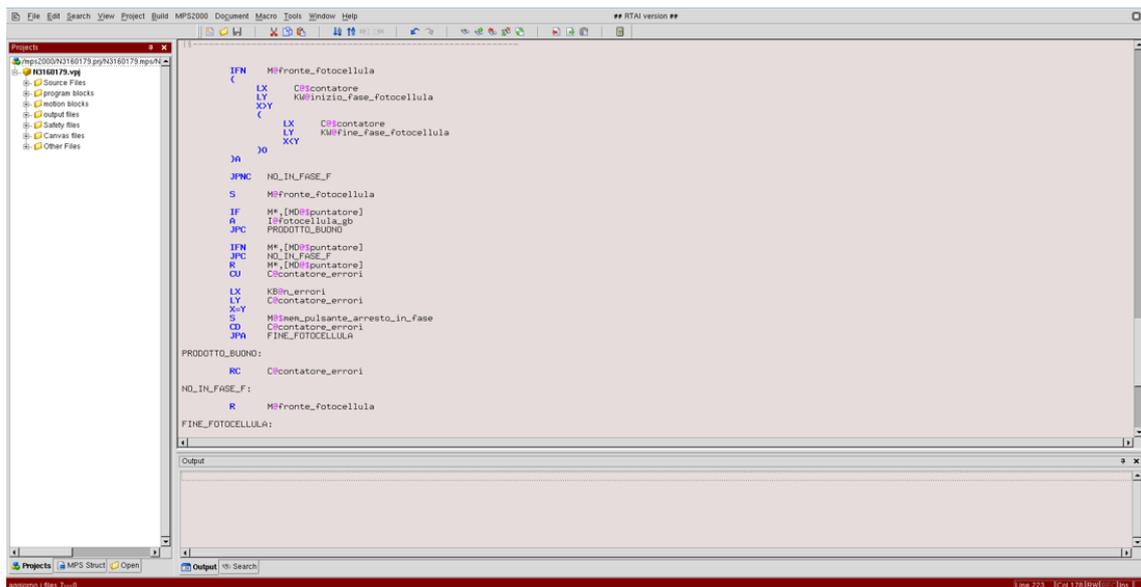


Figura 7: editor di programmazione

Grazie a questo programma è possibile scrivere le istruzioni che consentiranno successivamente alla macchina automatica di svolgere le attività per cui è stata pensata.

La programmazione avviene tramite la stesura di un codice in un linguaggio simile al C, che permette di gestire variabili e funzioni come memorie, contatori, timer, salti (jump),... Come il C, anche il linguaggio di programmazione MPS2000 ha le proprie regole di programmazione, che sono contenute nel “Manuale del programmatore”, un testo redatto dagli stessi tecnici Marchesini che hanno provveduto alla realizzazione del software e messo a disposizione di tutti i dipendenti.

Una volta scritto il programma il compilatore verifica che non ci siano errori di sintassi: qualora ve ne fossero, è necessario correggerli e rilanciare la compilazione; quando invece il programma risulta compilato correttamente in ogni suo campo, viene salvato ed è pronto per esser scaricato in macchina.

In sede di apprendimento è stata utilizzata una macchina virtuale in Linux che permette l’emulazione del pannello di controllo della macchina automatica reale, consentendo così di simulare il funzionamento della stessa e correggere eventuali errori presenti all’interno del codice. Ciò è possibile grazie ad un potente strumento sviluppato dai tecnici Marchesini, ovvero la diagnostica a bordo macchina, che consente all’operatore di osservare come evolvono gli stati degli operandi all’interno del listato mentre la macchina è in funzione.

Uno screenshot del pannello di controllo della macchina automatica è mostrato in Figura 8.

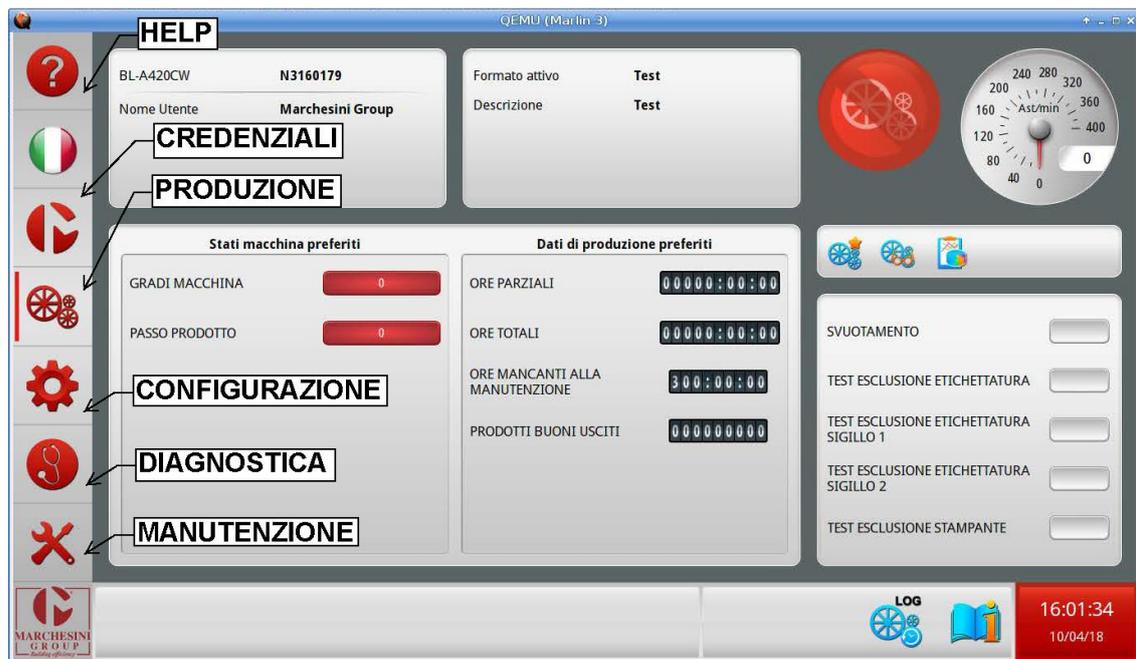


Figura 8: pannello di controllo delle macchine automatiche

Il pannello di controllo della macchina automatica presenta una schermata iniziale nella quale vengono riassunte alcune informazioni generali della stessa, come per esempio:

- matricola della macchina;
- credenziali dell'operatore che ha effettuato il login;
- formato in produzione con eventuale commento
- gradi macchina
- passo prodotto
- ore parziali dall'ultimo azzeramento del timer
- ore totali di funzionamento
- ore mancanti alla successiva manutenzione
- prodotto buoni realizzati dalla macchina

Sul lato destro dello schermo è inoltre presente un "tachimetro" che indica il numero di pezzi al minuto che la macchina sta producendo. Al di sotto di esso è presente una tendina dalla quale si può scegliere di effettuare alcune operazioni oppure test per i prodotti.

Sul lato sinistro dello schermo invece è presente l'elenco di tutti i sottomenù ai quali l'operatore può accedere.

Partendo dall'alto:

- *help*: cliccandoci sopra, la pagina entra in una particolare stato per il quale se si clicca su di una finestra qualsiasi non si entra nel relativo sottomenù ma si apre una tendina con una breve descrizione delle operazioni possibili all'interno della casella selezionata;
- *scelta lingua*: questo menù permette di selezionare la lingua del display;

- *credenziali*: è l'area che permette all'operatore di sbloccare l'accesso alla macchina. Ogni operatore che vuole apportare modifiche alla macchina o visualizzare dei dati di produzione ad essa relativi deve fare l'accesso con le proprie credenziali, quali username e password; questo è un sistema di sicurezza necessario per impedire l'accesso alle persone non autorizzate. Inoltre, ad ogni operatore registrato viene assegnato un livello di competenza: questo è un numero che varia tra 0 e 4. 0 è il più basso livello di accesso e permette un limitato numero di operazioni; 4 è il grado di accesso più elevato e si ha a disposizione il massimo numero di funzioni consentite dall'azienda ad un utente esterno; questi però non possono intervenire sul codice di testo MPS e compiere altre operazioni che l'azienda riserva solo ed esclusivamente ai propri tecnici. Il livello di accesso in questione è detto "SuperUser" e permette un accesso illimitato a qualsiasi area e funzione della macchina;
- *produzione*: è la schermata rappresentata in Figura 8, nella quale vengono riportati alcune generalità della macchina;
- *configurazione*: da questo menù è possibile accedere ai dati relativi ai vari formati del prodotto, ai parametri della sua etichetta, ai selettori della macchina, necessari per effettuare vari test; se la macchina è fornita di una bilancia, in questo menù si possono visualizzare anche i dati ad essa riferiti. Se si è fatto il login come SuperUser si dispone di un ulteriore sottomenù dal quale si possono modificare le quote della macchina, si ha accesso a selettori dedicati e ai dati che riguardano il formato base, i parametri base e i file dei messaggi della macchina;
- *diagnostica*: in quest'area si vede il listato MPS (ovvero il programma vero e proprio scritto dal programmatore), si visualizzano gli operandi con i loro corrispettivi valori, si impostano i traser e si graficano i risultati ottenuti da tali test, si possono forzare dei valori degli operandi MPS, si configura la rete EtherCAT e si scarica la diagnostica su pendrive. Se l'accesso è effettuato da un SuperUser si ha accesso ad ulteriori finestre che consentono di visualizzare informazioni aggiuntive riguardanti il listato MPS, di fare una forzatura di livello superiore dei valori degli operandi e di visualizzare i blocchi ES multipli;
- *manutenzione*: in questo menù sono presenti selettori per le operazioni di manutenzioni, sottomenù con informazioni varie su hardware e software, elenchi dei login, finestre per aggiornare oppure scaricare su pendrive i listati MPS2000 e EasyDoor, finestre per il controllo del sistema e per la gestione dello schermo touchscreen. Se si è fatto il login come SuperUser si dispone di un ulteriore sottomenù dal quale si può chiudere o ricaricare il programma.

Grazie al lavoro di simulazione svolto dalla macchina virtuale è possibile simulare il funzionamento della macchina reale presente nel capannone: ciò comporta un significativo risparmio del tempo di programmazione, la risoluzione a priori di molti problemi che altrimenti sarebbero difficili anche solo da prevedere e l'abbattimento dei costi del prodotto finale.

## 2.2. MAC2000

In parallelo alla programmazione dei PC è presente la programmazione del movimento assi. Con questo termine si indica tutta la movimentazione degli organi mobili dei robot, i quali devono essere progettati non solo per realizzare un movimento, ma per seguire una determinata traiettoria, che parte da un punto di inizio e termina in un punto di fine accuratamente individuati. Inoltre il movimento deve avvenire seguendo punti intermedi ben definiti, che ottimizzano il lavoro del robot, evitando la collisione di ogni sua parte con il mondo esterno.

Nei primi anni '80, data la scarsa diffusione dell'elettronica, tutto ciò era realizzato tramite camme meccaniche: si ottenevano macchine automatiche molto rigide e poco flessibili alle richieste del mercato e del cliente, che portavano con sé tutte le problematiche legate all'usura e al consumo meccanico degli organi in movimento, ma che al tempo stesso rappresentavano la miglior soluzione per quell'epoca.

Oggi le camme meccaniche sono state sostituite dalle cosiddette “camme elettroniche”, ovvero tabelle che riportano punto per punto la successione delle posizioni che il robot deve raggiungere per realizzare la traiettoria voluta e il tempo in cui tale movimento deve avvenire. In questo modo si controlla la posizione, la velocità, l'accelerazione e il jerk di ogni singolo motore della macchina automatica in maniera autonoma, senza che nessun motore dipenda da altri elementi. Questo è un grande vantaggio in fase di programmazione in quanto permette di considerare ogni motore in un contesto estrapolato dalla realtà nella quale dovrà poi operare. Però, quando il motore verrà montato sulla macchina automatica le cose si complicano: bisogna che tutti i motori operino in maniera coordinata. Questa problematica ha portato gli ingegneri della Marchesini ad affrontare il problema della sincronizzazione degli organi mobili presenti a bordo macchina. La difficoltà è stata superata considerando tutti i singoli motori come gli *slaves* di un unico *master*. Ovvero ogni singolo motore con il proprio azionamento viene collegato secondo la corretta fase con il master della macchina automatica in modo che per muovere l'intero robot basti comandare il solo master. Infatti le macchine automatiche che oggi vengono costruite seguono la filosofia appena citata: le traiettorie dei singoli motori vengono caricate all'interno del PC che si trova a bordo macchina e quest'ultimo, tramite un apposito sistema di comunicazione, trasferisce le informazioni agli azionamenti dei vari motori, guidandoli lungo la corretta traiettoria.

Tutto ciò è reso possibile grazie al MAC2000, un sistema di controllo assi completamente digitale, scritto nel linguaggio di programmazione C, che gestisce a bordo macchina la traiettoria dei vari motori. Per comprenderne meglio il funzionamento di questo software, vengono di seguito riportate in maniera semplificata le principali operazioni che esso compie durante la fase di esecuzione:

- elaborazione delle traiettorie con generazione di nuovi riferimenti da trasmettere agli azionamenti dei motori;
- trasferimento di tali riferimenti agli azionamenti dei motori
- ricezione della posizione attuale degli azionamenti

Quest'ultimo punto è necessario per avere istante per istante la certezza che il robot si sia spostato nella corretta posizione e non abbia raggiunto coordinate differenti da quelle indicate.

### **2.2.1. Comunicazioni tra PC e azionamenti**

Per poter realizzare la trasmissione dei dati tra PC a bordo macchina e i singoli driver dei motori è necessaria una rete di comunicazione. Il software MAC2000 consente di utilizzare sia la rete Canbus (SbcCan) che la rete EtherCAT (COE): a seconda della scelta effettuata, si hanno differenti prestazioni del sistema, diverse strutture dei controlli dei motori e un differente numero dei moduli controllabili contemporaneamente. Una tra le differenze più rilevanti sta nel fatto che la seconda è una comunicazione di tipo *real time* mentre non si può dire lo stesso della prima.

La comunicazione tra azionamenti e PC viene realizzata servendosi di appositi moduli, grazie ai quali è possibile riferirsi anche agli operandi del software MPS2000. Non tutti i moduli messi a disposizione del programmatore sono uguali: si differenziano in sincroni e asincroni. La differenza sostanziale è rappresentata dal tempo che intercorre tra due comunicazioni successive: nel primo caso tale intervallo è di 1 ms; nel secondo invece è di 2,048 ms. La linea di trasmissione dei dati deve garantire che sia la trasmissione sincrona che quella asincrona siano contemporaneamente utilizzabili per il controllo degli azionamenti. Questo permette al programmatore di scegliere quali informazioni hanno maggiore priorità e quali invece possono essere aggiornate in intervalli di tempo maggiori.

## 2.2.2. Interfaccia MAC2000

La programmazione del movimento assi di una macchina viene realizzato secondo uno schema a blocchi, di cui un esempio in Figura 9.

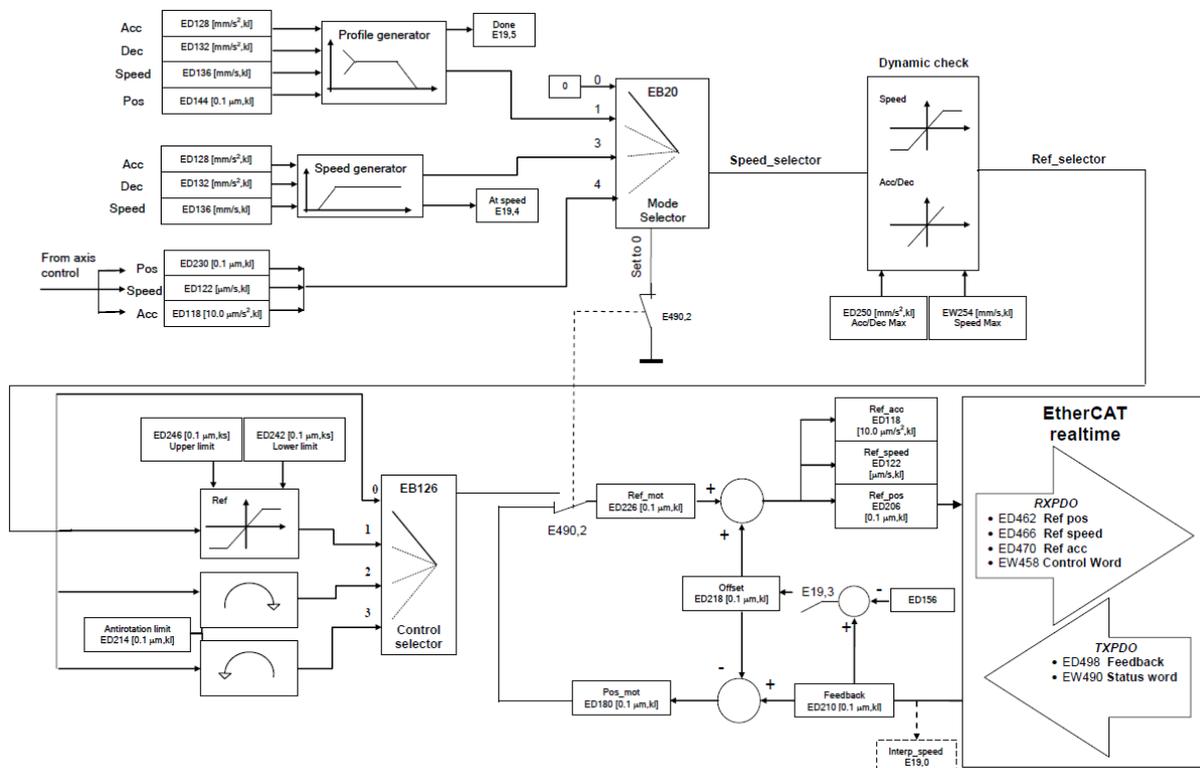


Figura 9: esempio di interfaccia MAC2000

I blocchi presenti nel diagramma rappresentano dei programmi scritti in C. Questi ricevono in ingresso sia la traiettoria di riferimento impostata dal programmatore che la retroazione dei dati rilevati sul campo, in termini di posizione, velocità e accelerazione: dal loro confronto ne esce un'informazione che, in seguito ad una precisa elaborazione e vari controlli, viene inviata al driver che aziona il relativo motore.

Dato che le istruzioni per gestire un motore non sono un numero troppo elevato, i tecnici della Marchesini hanno pensato di riunire e standardizzare almeno le funzioni più comuni sotto certe librerie, al fine di semplificare e velocizzare il lavoro del programmatore. Da ciò sono nate funzioni come "generatore rampa di accelerazione", "generatore rampa di decelerazione", "generatore di velocità costante", "generatore di profilo trapezoidale", "generatore di profilo definito dall'utente": tutti algoritmi adibiti a generare segmenti di profili che assemblati tra loro danno origine alla traiettoria che il robot deve realizzare.

Come si vede anche nell'immagine, alcune finestre di ingresso/uscita sono caratterizzate da delle sigle EDxxx, EBxxx, EDxxx in cui "xxx" rappresenta un

valore numerico. Ad esse è associato un indirizzo di memoria (un bit, una sequenza di bit, un byte oppure una sequenza di byte) nel quale bisogna leggere o scrivere una determinata informazione.

## **2.3. SAFETY**

Nel settore della produzione di macchine automatiche ha una rilevante importanza la sicurezza a bordo macchina e nelle sue immediate vicinanze. Con il concetto di sicurezza si intende il rischio di infortunio a cui è sottoposto l'operatore quando si trova in prossimità o all'interno della macchina, sia durante le ordinarie funzioni di produzione sia nelle operazioni di manutenzione ordinaria e straordinaria della macchina.

La normativa europea assegna per ogni categoria di macchine automatiche un grado di sicurezza dipendente dalla tecnologia utilizzata per la realizzazione e la messa in sicurezza della stessa. Tale valore è un numero compreso tra 1 e 4, dove 1 rappresenta il minimo valore di sicurezza mentre 4 è il massimo. Come descritto anche in precedenza, qualche anno fa la Marchesini Group ha fatto la scelta di passare dal livello di sicurezza CAT 2 al livello di sicurezza CAT 3, al fine di ridurre il più possibile il rischio infortuni collegati all'utilizzo delle macchine automatiche, mettendo così sul mercato un prodotto innovativo e più sicuro.

Questa scelta ha fortemente condizionato le direttive del gruppo, il quale è passato all'adozione di una rete EtherCAT per la gestione e la comunicazione dei dispositivi a bordo macchina, necessaria per garantire il livello di sicurezza ricercato. Fatta questa scelta, ci si è poi posti il problema se acquistare la rete da un ente terzo oppure se svilupparne una al proprio interno. In seguito ad approfondite analisi sia dal punto di vista economico che tecnico, l'azienda ha optato per quest'ultima scelta, creando al proprio interno un laboratorio di persone altamente specializzate adibite allo sviluppo dell'intera rete di comunicazione, rendendosi quindi indipendente da soggetti esterni. Partendo da una piattaforma open source Linux RTAI, lo sviluppo delle reti master/slave ha visto una crescita repentina nel giro di pochi anni in quanto le esigenze del mercato hanno richiesto macchine sempre più veloci e precise.

Il pieno controllo della rete ha permesso alla Marchesini Group di adattarsi perfettamente anche quando è stato deciso di introdurre i moduli di comunicazione Beckhoff all'interno delle proprie macchine, implementando l'utilizzo dei moderni PLC.

### 2.3.1.PLC

I PLC (o Controllori Logico Programmabili) sono dispositivi che consentono di programmare e gestire le attività svolte dai robot e dalle macchine automatiche presenti nelle linee di produzione. Ogni PLC è costituito da numerosi moduli, montati in successione su di una base fissa e cablati per eseguire operazioni logiche.

La Marchesini Group ha fatto la scelta aziendale di montare sulle proprie macchine i dispositivi TwinCAT Safety PLC, forniti dalla casa costruttrice Beckhoff, al fine di rispettare a pieno la categoria di sicurezza CAT 3 e performance level PL9 che tutte le macchine di produzione devono garantire.

I moduli possono essere classificati in base al tipo di segnale trasmesso:

- moduli digitali
- moduli analogici

Ovvero i primi operano con segnali digitali, i secondi con segnali analogici. Un'altra suddivisione può essere la seguente:

- moduli input
- moduli output

Mentre i primi sono predisposti a ricevere segnali elettrici in ingresso, i secondi, in seguito alla dovuta elaborazione, inviano i segnali di uscita. Un'ulteriore distinzione può essere fatta secondo la seguente logica:

- moduli safety
- moduli non safety

In ottica di applicare una logica a doppio contatto, i segnali cablati su moduli safety vengono classificati come sicuri (moduli Safety Singol Channel, anche Safety SC) e sono il riferimento per il controllo e il confronto del segnale non sicuro, collegato ad un modulo non safety. Per comprendere la differenza di questa classificazione si riporta un semplice esempio. Si consideri un motore rotativo che riporta in ingresso al PLC un'informazione analogica di posizione: questa informazione normalmente viene collegata ad un ingresso non safety, quindi considerato non sicuro. La posizione angolare dello stesso motore viene rilevata anche da un encoder rotativo che viene collegato a un modulo safety, che quindi genera un'informazione considerata sicura. Il confronto tra i due permette di individuare eventuali mal funzionamenti dei contatti, oltre ovviamente a garantire la correttezza dell'informazione sulla posizione del motore.

### **2.3.2.Sensori**

I sensori sono i dispositivi di interfaccia tra il mondo reale e quello virtuale: sono in grado di rilevare una certa grandezza sul campo e di convertire il dato rilevato in un segnale elettrico in corrente oppure in tensione e di inviarlo poi al PLC, il quale lo elabora.

I sensori possono essere classificati in due macro categorie:

- sensori semplici (digitali), i quali trasmettono un segnale 0/1: un esempio può essere una fotocellula utilizzata per la visione di un oggetto. Se questo è presente, la fotocellula fornisce un segnale di tipo logico 1, viceversa, il segnale logico è 0.
- sensori “intelligenti” (analogici), che sono in grado di rilevare un segnale che può assumere certi valori in un determinato range di variazione. Può esserne un esempio un sensore che rileva la posizione di un asse lineare piuttosto che la posizione angolare di un motore rotativo. Questa tipologia di sensori è in grado quindi di rilevare direttamente una grandezza fisica reale.

### **2.3.3.Attuatori**

Gli attuatori sono i dispositivi che provvedono alla movimentazione delle macchine automatiche; i più diffusi nell’ambito delle macchine automatiche sono senz’altro i motori elettrici. Questi vengono azionati direttamente dai driver e consentono alle macchine e ai robot di compiere le azioni e i movimenti per le quali sono stati costruiti.

Ne esistono di diverse tipologie:

- motori in corrente continua
- motori trifase in corrente alternata
- motori brushless
- motori passo-passo

Mentre i primi stanno andando lentamente in disuso causa le numerose problematiche correlate soprattutto all’usura, i motori brushless e i passo-passo sono certamente i più diffusi nell’ambito delle macchine automatiche. Di piccole dimensioni, compatti e relativamente poco costosi, questi motori sono in grado di raggiungere velocità e accelerazioni anche molte elevate, sviluppando importanti coppie pur garantendo un’ottima ripetibilità e precisione di posizionamento.

Di rilevante importanza nell’ambito dell’automazione sono anche gli attuatori pneumatici, sia i rotativi a palette che i cilindri pneumatici, utilizzati per realizzare movimenti più semplici e in ambiti in cui non è richiesta una precisione troppo elevata.

### 3. PIATTAFORMA DI SVILUPPO TWINCAT

Il funzionamento dei PLC viene programmato utilizzando un software commerciale gratuito di nome TwinCAT. Tale software permette al programmatore di scrivere le istruzioni sia in modalità “offline”, ovvero potendo programmare da remoto il PLC senza averlo fisicamente a disposizione, che in modalità “online”, cioè quando il PLC è nella disponibilità del programmatore ed è collegato fisicamente al PC attraverso un cavo EtherCAT. Per passare da una modalità all'altra si dispone di due icone: partendo da destra della Figura 10, se si clicca la prima icona (quella blu) ci si porta nella Config Mode, ovvero la modalità offline; se si clicca quella a lato (icona verde) si entra nella modalità online. È presente una terza icona (Activate Configuration) che permette di caricare la configurazione offline nella modalità online.



Figura 10: in ordine, le icone di Activate Configuration, Online Mode e Config Mode

Quando il programma viene aperto, come prima cosa viene richiesto al programmatore di creare un nuovo progetto, il quale è necessario per scrivere tutte le istruzioni che andranno successivamente scaricate nel PLC. In seguito, dopo averlo nominato e creato, si aprono nella finestra Solution Explorer dei menù dal quale bisogna scegliere cosa e come si vuole programmare. Le opzioni a disposizione sono le seguenti:

- System
- Motion
- PLC
- Safety
- C++
- I/O

Il menù “System” contiene tutte le informazioni relative al programma e al progetto, quali licenze del software utilizzato, aree di memoria dedicate e parametri del PLC forniti dalla casa costruttrice.

Il menù “Motion” permette di configurare e gestire tutti gli elementi che realizzano il movimento di un asse, come per esempio un motore elettrico con il relativo driver.

Nel menù “PLC” si trovano numerosi sottomenù nei quali è presente la voce per scrivere il programma vero e proprio in un linguaggio di programmazione simile al C, si dichiarano le variabili globali e locali, si discriminano gli input dagli output e, durante le prove sul PLC, si forzano i valori delle variabili.

Nel menù “Safety” si programma la logica di tutti i blocchi funzionali di cui si parlerà in seguito, necessari per garantire il funzionamento in sicurezza del

dispositivo e della macchina ad esso associata. Questi vengono interfacciati tra loro collegando le uscite dei blocchi a monte con gli ingressi di quelli a valle, al fine di realizzare la data sequenza degli stati della macchina.

Il menù “C++” permette di realizzare un’interfaccia grafica.

Nel modulo “I/O” vengono visualizzati tutti i moduli del PLC, con i relativi ingressi e uscite hardware: di ciascuno di essi si possono configurare e visualizzare numerosi parametri.

I menù appena elencati sono strettamente correlati gli uni agli altri: per esempio una variabile che viene considerata d’uscita lato PLC sarà considerata come un ingresso lato Safety e viceversa. Perché tutto funzioni correttamente è necessaria un’attenta e scrupolosa fase di programmazione.

Prima di esporre la questione relativa alla programmazione, è importante presentare alcuni concetti fondamentali che stanno alla base dell’utilizzo di questo software.

Il PLC è un elemento hardware la cui componente software è in continua evoluzione nel tempo: infatti il suo stato di esecuzione è fortemente collegato allo stato nel quale si trovano i singoli blocchi funzionali che compongono il programma. Si analizzano di seguito i vari stati nel quale si può trovare un PLC.

Durante la corretta fase di elaborazione il programma si trova nello stato di RUN: se si attiva la modalità di visione dei valori online, il riquadro si colora di verde, come in Figura 11.

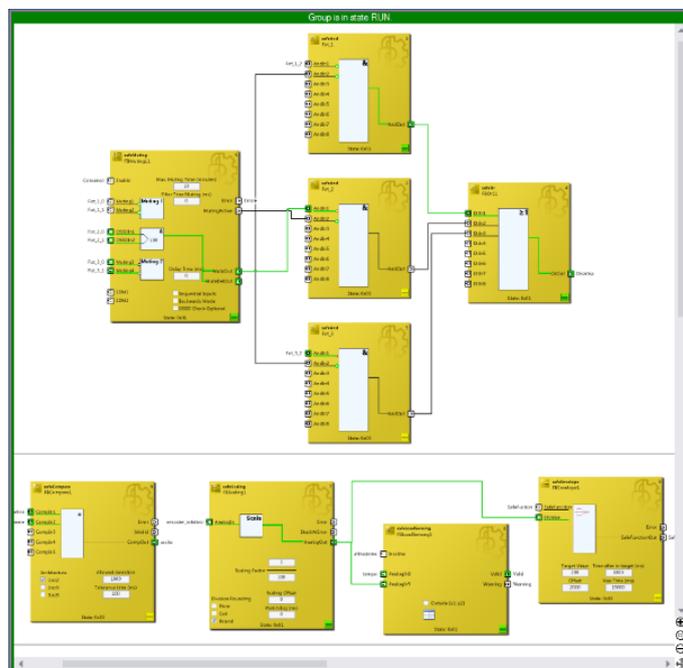


Figura 11: RUN mode

Se si verifica un errore, ovvero un blocco funzionale attiva l’uscita Error, l’intero programma entra nello stato di ERROR: se si attiva la modalità di visione dei valori online, il riquadro si colora di rosso, come in Figura 12.

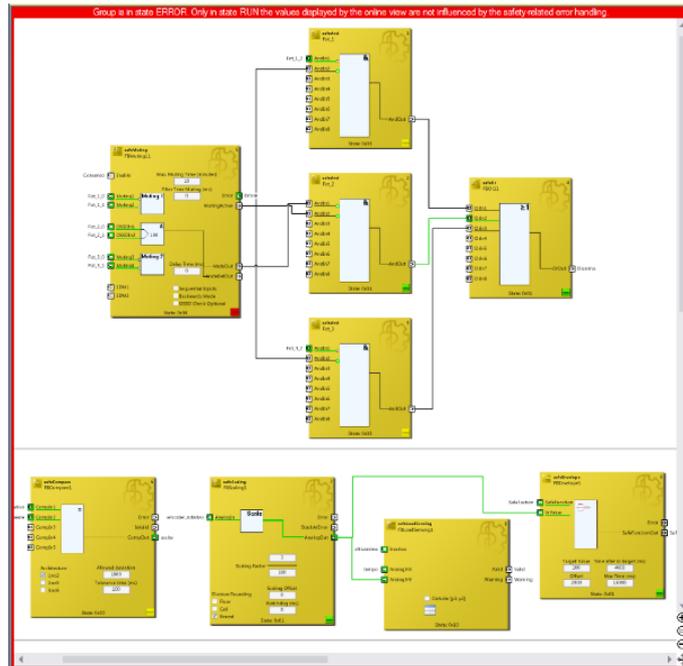


Figura 12: ERROR mode

Mentre il passaggio dalla fase di RUN a quella di ERROR è immediata non appena un blocco funzionale del programma entra in errore, non si può dire lo stesso per il passaggio opposto. Per uscire dalla fase di ERROR bisogna anzitutto liberare la macchina dalla condizione per la quale si è verificato l'errore, ovvero bisogna che ogni singolo blocco del programma abbandoni lo stato di errore. Ristabilita la corretta situazione bisogna dare il consenso per la ripartenza, impostando inizialmente nello stato logico TRUE l'entità presente di default in tutti i progetti con il nome di ErrorAcknowledgment. Il programma si porta quindi nello stato di RESET: se si attiva la modalità di visione dei valori online, il riquadro si colora di blu, come in Figura 13. Solo in seguito, per riprendere effettivamente l'esecuzione del programma, questa uscita deve tornare allo stato logico FALSE.

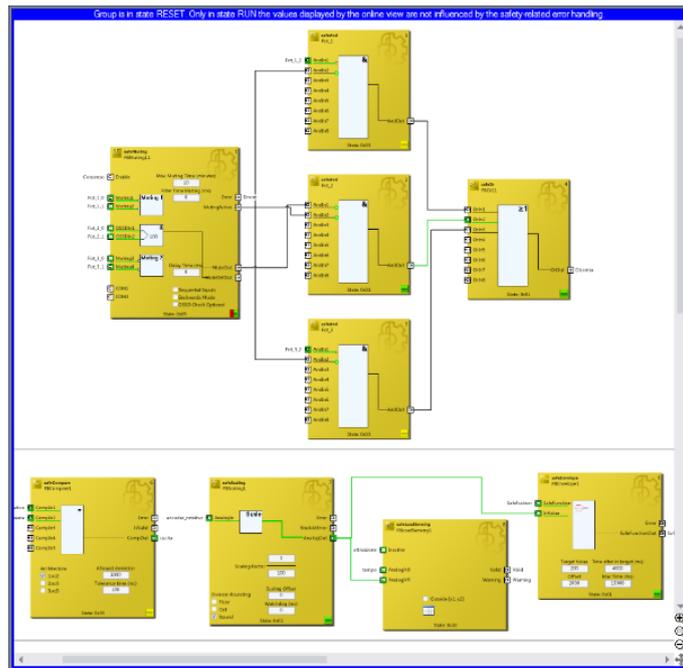


Figura 13: RESET mode

Come per il programma, anche i singoli blocchi funzionali possono trovarsi in differenti stati di esecuzione. Per riconoscere lo stato del singolo blocco è stato introdotto un piccolo riquadro nella parte inferiore del blocco stesso, che assume diverse colorazioni:

- verde: RUN, non ci sono errori e il blocco funzionale ha almeno un'uscita attiva
- giallo: SAFE, non ci sono errori e il blocco funzionale non ha nessuna uscita attiva
- rosso: ERROR/STOP, nel blocco funzionale è presente almeno un errore oppure è ancora in stato di STOP; in questa condizione tutte le uscite vengono impostate allo stato logico FALSE indipendentemente dalla logica

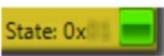
FB Icon	Description
	FB State: RUN In RUN state no error is present, and the output of the FB is set.
	FB State: SAFE In SAFE state no error is present, and the output of the FB is NOT set.
	FB State: ERROR/STOP In ERROR/STOP state an FB error is present or the FB is still in STOP state. This is the case if the group has not yet been started.

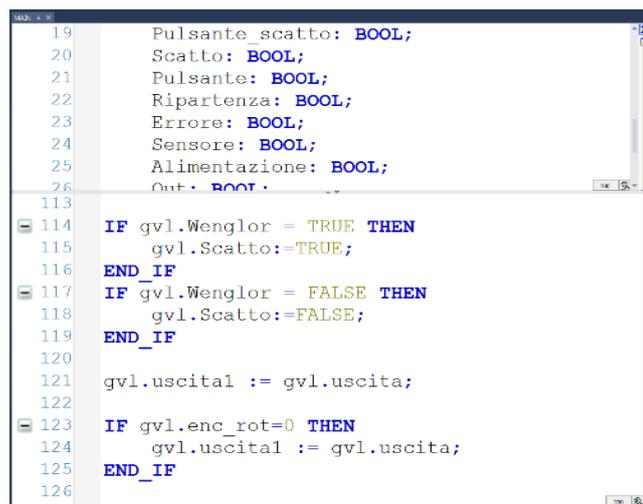
Figura 14: descrizione degli stati dei blocchi funzionali

## 3.1. Configurare un progetto TwinSAFE

Per gestire una macchina automatica è necessario programmare il suo funzionamento sotto ogni punto di vista, sia MPS2000 che Safety; questo è possibile anche grazie al software TwinCAT e ai menù che questo mette a disposizione del programmatore, già descritti in precedenza.

### 3.1.1. Menù PLC

Per scrivere le istruzioni che regolano il funzionamento della macchina lato PLC è necessario accedere all'omonimo menù: da qui, si ha accesso a numerose sottocartelle, inizialmente tutte vuote. Facendo tasto destro su una qualsiasi di queste e selezionando la voce Add, il programma propone una serie di elementi che si possono aggiungere al proprio progetto. Per iniziare si seleziona la dicitura POU (Program Organization Unit) che, una volta compilata correttamente, crea una nuova voce MAIN(PRG). Cliccandoci sopra si apre una nuova finestra divisa in due aree: quella inferiore è destinata alla stesura delle righe di codice (cicli FOR, WHILE, scelte condizionate IF...). Ogni volta che si introduce una nuova variabile, il programma automaticamente la riconosce e chiede al programmatore di definirne il nome e la tipologia (BOOL, INT, UINT, REAL...). Riempiti questi campi, non appena si conferma la scelta, il programma compila in automatico anche l'area superiore, destinata appunto alla dichiarazione delle variabili. Vengono qui riportati nome e estensione di tutte le variabili presenti all'interno del listato; quest'area è direttamente accessibile dal programmatore in qualsiasi istante per realizzare eventuali modifiche (vedi Figura 15).



```
19 Pulsante_scatto: BOOL;
20 Scatto: BOOL;
21 Pulsante: BOOL;
22 Ripartenza: BOOL;
23 Errore: BOOL;
24 Sensore: BOOL;
25 Alimentazione: BOOL;
26 Out: BOOL;

113
114 IF gvl.Wenglor = TRUE THEN
115     gvl.Scatto:=TRUE;
116 END_IF
117 IF gvl.Wenglor = FALSE THEN
118     gvl.Scatto:=FALSE;
119 END_IF
120
121 gvl.uscital := gvl.uscita;
122
123 IF gvl.enc_rot=0 THEN
124     gvl.uscital := gvl.uscita;
125 END_IF
126
```

Figura 15: MAIN

Se all'interno del listato si utilizzano segnali o valori provenienti direttamente dalla sensoristica, è necessario seguire una determinata procedura che metta in relazione i segnali provenienti dai sensori con le variabili software. In questo caso

bisogna cliccare la voce Global Variable List che si trova sotto la stessa Add a cui si faceva riferimento in precedenza. Come prima, compilati i campi necessari e confermata la scelta, si crea una nuova voce nella sottocartella GVLs nella quale è possibile dichiarare tutte le variabili che riportano le informazioni rilevate direttamente sul campo. Per queste variabili deve essere specificato, oltre al nome, se si tratta di ingressi oppure di uscite e la tipologia a cui appartengono. Se vogliono essere richiamate nel MAIN bisogna utilizzare la seguente dicitura: gvl.nome\_variabile.

```
8 Power_low AT %Q*:BOOL;
9
10 Run_safety AT %Q*: BOOL;
11 Err_Ack_safety AT %Q*: BOOL;
12
13 Fermo_macchina AT %I*: BOOL;
14 Scatto AT %Q*: BOOL;
15 Errore AT %I*: BOOL;
16 sensore AT %I*: BOOL;
17 Wenglor AT %I*: BOOL;
18 contatti AT %I*: BOOL;
19 Restart AT %Q*: BOOL;
20 alimentazione AT %Q*: BOOL;
21 valore AT %I*: UINT;
22
23 Uscita AT %I*: BOOL;
24 uscital AT %Q*: BOOL;
25
26 tempo AT %Q*: UINT;
27 enc_rot AT %I*: INT;
28 onoff AT %I*: BOOL;
29 enc_lin AT %I*: INT;
```

Figura 16: GVL

Quando il programma viene compilato, per tutte le variabili appartenenti a questo menù vengono create le relative entità software, opportunamente suddivise tra ingressi ed uscite: queste istanze permettono successivamente di collegare queste grandezze alle loro corrispettive entità presenti nel sottomenù Safety. Questo è il principale anello di congiunzione tra la programmazione lato MPS e lato Safety.

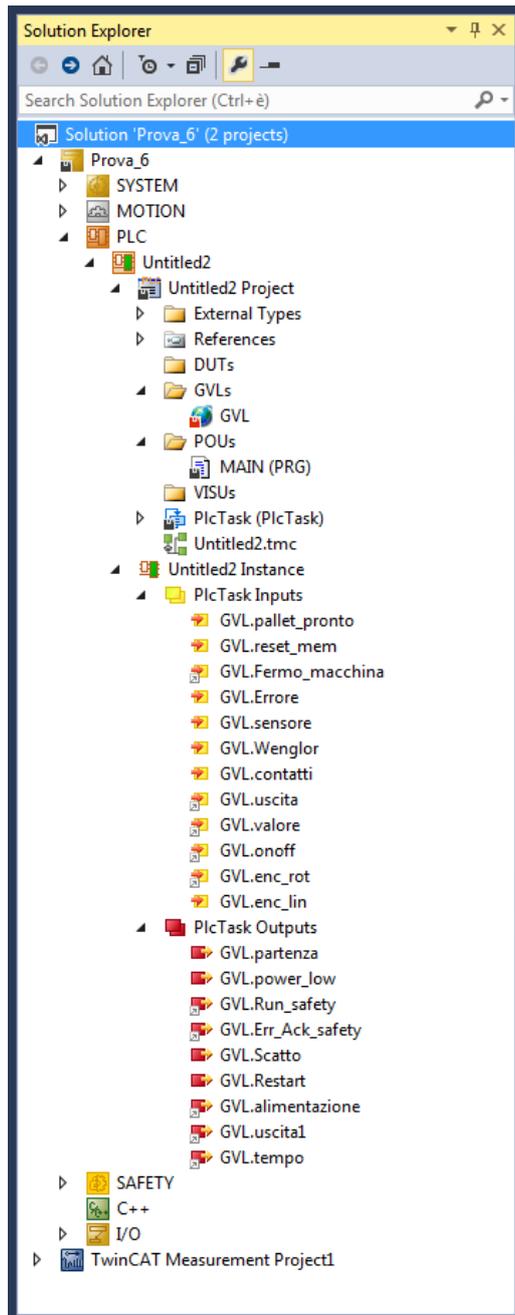


Figura 17: variabili input e output GVL

### 3.1.2. Menù I/O

Nel sottomenù Safety viene realizzata la programmazione a blocchi della logica presente all'interno della macchina. Quest'area di TwinCAT è fortemente correlata alla sezione successiva I/O, all'interno della quale vengono rilevati, organizzati e gestiti tutti i moduli montati sul PLC. Per fare ciò, è necessario anzitutto creare una rete di comunicazione facendo tasto destro sulla voce Devices e selezionando l'alternativa Add New Item. Nella finestra che si apre bisogna specificare la tipologia di rete che si desidera utilizzare, oltre al nome che gli si vuole assegnare;

nell'esercizio sviluppato si è utilizzato il master di una rete EtherCAT. Completata questa operazione, si fa nuovamente tasto destro sulla voce Devices e si seleziona l'alternativa Scan. Se questo non fosse possibile, assicurarsi che il PLC sia correttamente collegato al PC e di aver attivato la Config Mode, unica modalità nella quale è possibile realizzare questa operazione. La finestra che si presenta, propone due elenchi: quello di sinistra riporta tutti i moduli che sono stati rilevati sul PLC, mentre quello di destra i moduli che si ha intenzione di configurare e quindi che si vogliono utilizzare. Selezionando i moduli dall'elenco di sinistra bisogna spostarli in quello di destra prestando particolare attenzione all'ordine che questi assumono sul PLC reale (vedi Figura 18).

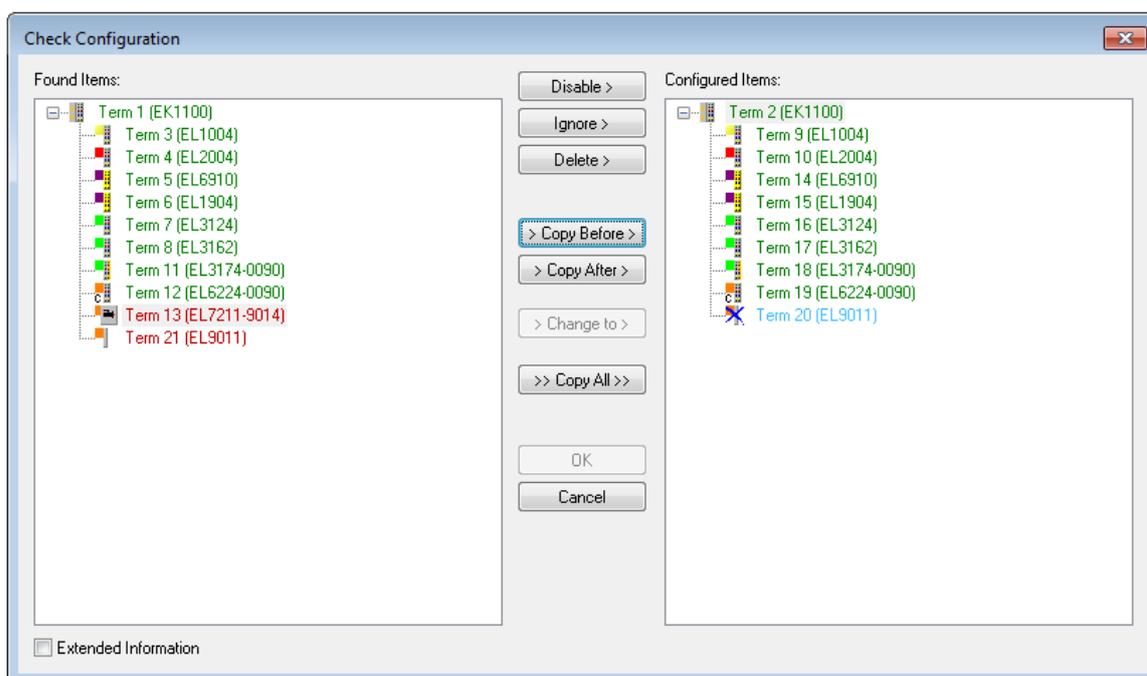


Figura 18: configurazione dei moduli del PLC

Come si vede dall'immagine, le voci assumono colorazioni differenti in base a come vengono configurati i vari moduli:

- Verde: il modulo è presente ed è correttamente configurato
- Blu: il modulo è presente ed è correttamente configurato, ma la revisione del modulo non coincide. Se la revisione è successiva rispetto a quella configurata, il modulo può essere utilizzato in seguito ad un corretto settaggio delle impostazioni; se la revisione è precedente rispetto a quella configurata, il modulo non può essere utilizzato
- Azzurro: il modulo è ignorato
- Rosso: il modulo non è presente nell'elenco "oggetti configurati" oppure è presente ma ha differenti proprietà da quelle specificate

Solamente quando le voci di tutti i moduli sono colorate di verde si attiva il tasto Ok, che permette di confermare la scelta; quando ciò avviene, il programma provvede a configurare secondo impostazioni standard tutti i moduli selezionati: nel sottomenù I/O sono quindi comparse le voci che descrivono ciascun modulo.

Cliccando su uno qualsiasi di essi si apre una finestra con numerose tendine: da queste è possibile avere informazioni relative ai canali di cui esso dispone, allo stato in cui si trovano, al valore che assumono istante per istante e molte altre informazioni utili per essere configurati e personalizzati. Ad esempio, per configurare un dispositivo TwinSAFE SC che si collega ad un modulo, bisogna entrare nella tendina Slots e selezionarlo dalla finestra in cui viene elencato (quella di destra) e spostarlo nella finestra a lato, mettendolo così nella disponibilità del modulo, come si vede in Figura 19, in cui tutti i dispositivi sono già stati configurati.

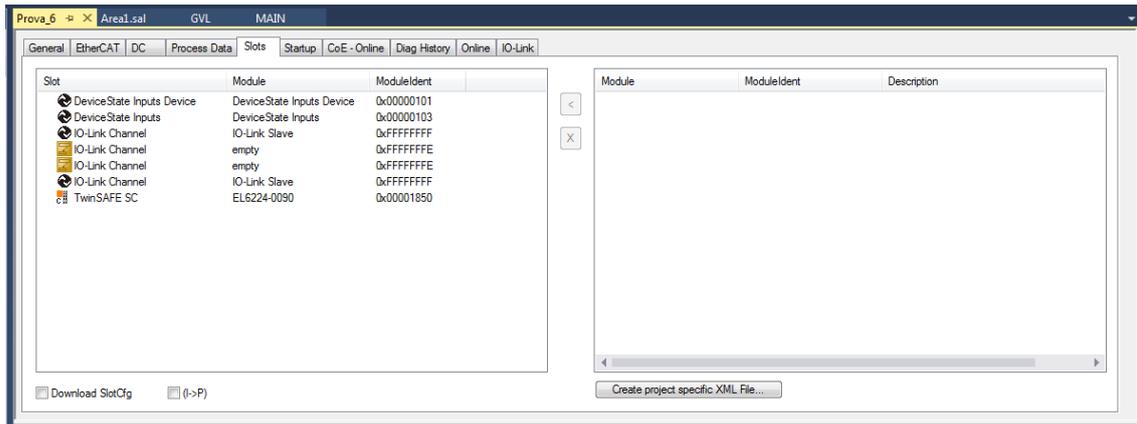


Figura 19: tendina Slots per configurazione di un dispositivo collegato ad un modulo del PLC

Se il modulo è adibito a gestire segnali I/O link provenienti da sensori, è presente anche una tendina nella quale tali dispositivi devono essere configurati. Cliccando sulla scritta I/O-Link si apre una finestra all'interno della quale vengono visualizzate tutte le porte d'ingresso del modulo in questione; inizialmente vuote, queste devono riportare per ogni ingresso le informazioni relative al dispositivo ad esse associato. Prima di realizzare questa configurazione, è necessario importare in TwinCAT i file contenenti tali informazioni: questi sono facilmente reperibili online dalla pagina dei costruttori dei sensori. Ottenuti tali file e importati correttamente in TwinCat attraverso l'opzione Import Devicedescription, si procede cliccando sulla voce Scan devices e, se i sensori sono stati correttamente cablati sul PLC, il sistema associa automaticamente ad ogni ingresso il relativo dispositivo collegato, come mostrato in Figura 20.

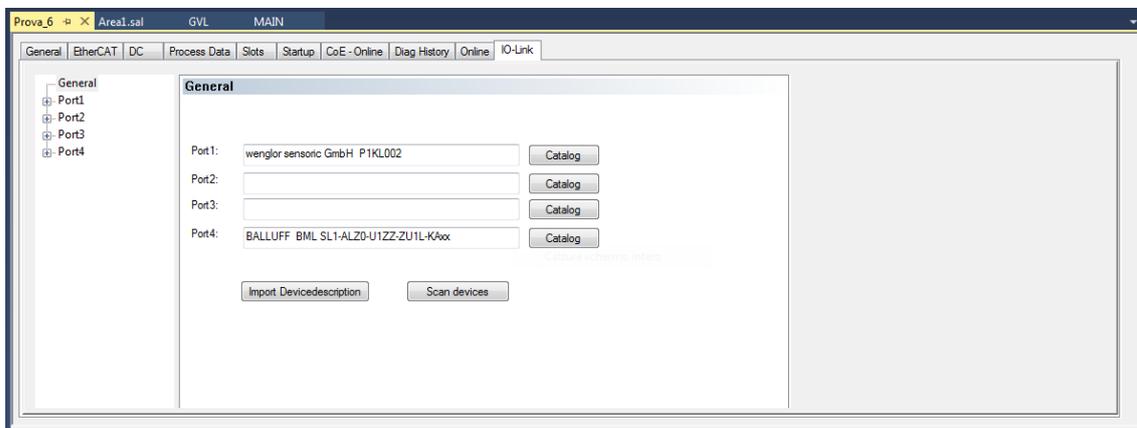


Figura 20: tendina IO-Link per configurazione di dispositivi I/O Link collegati ad un modulo del PLC

Tra le altre tendine presenti in questa finestra, è particolarmente importante quella nominata CoE - Online: questa riporta tutte le memorie visibili e accessibili dal programmatore, di cui sono specificati sia gli indirizzi che i valori da esse assunti.

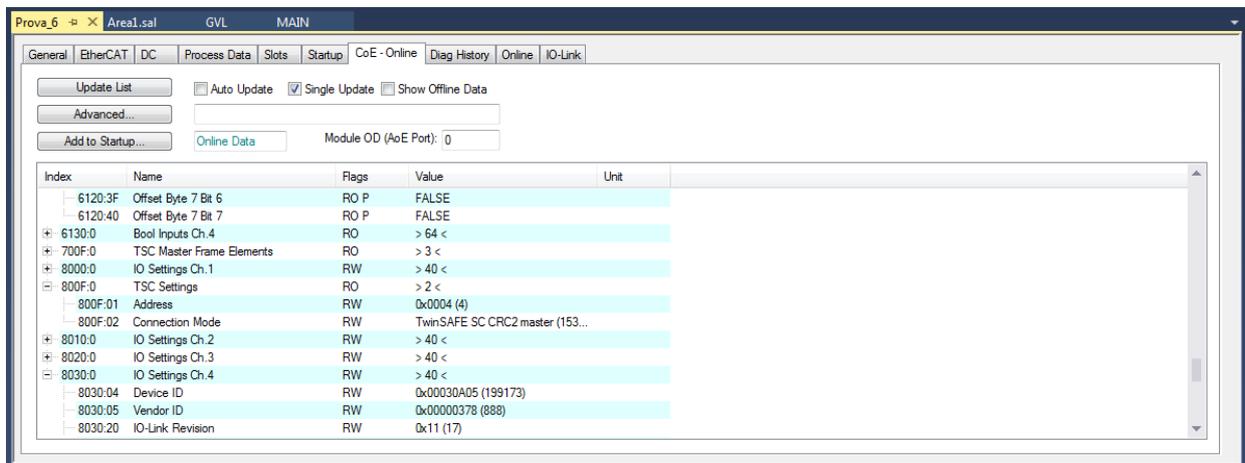


Figura 21: tendina CoE - Online

Realizzata la configurazione degli elementi hardware collegati alle porte dei moduli del PLC, si procede alla realizzazione della parte Safety della macchina.

### 3.1.3. Menù SAFETY

Per iniziare, è necessario creare il progetto TwinCAT Safety Project: tasto destro sul sottomenù SAFETY, quindi si seleziona Add New Item; tra i vari progetti proposti di default, si sceglie quello in cui è già stato configurato l'ErrorAcknowledgment. Così facendo, oltre al progetto viene creata la voce Target System: accedendo a questa finestra si seleziona il modulo safety di comunicazione che funge da PLC di sicurezza (in questo caso è EL6910); in automatico la scheda dati del modulo viene compilata e viene verificato che il Dip Switch presente sul modulo fisico coincida con quello impostato nel software di programmazione. In questa finestra si trovano anche altre informazioni relative agli User Administrator che possono modificare il PLC, riguardanti i backup e i restore effettuati e impostazioni varie di settaggio del progetto. Definito anche il modulo safety di riferimento del progetto, bisogna creare i gruppi TwinSafeGroup (gruppo operativo di sicurezza): tasto destro sul nome del progetto, Add e poi si seleziona New Item. Si stabilisce se aprire un gruppo standard (TwinSafeGroup) oppure uno con gli input già preconfigurati (TwinSafeGroup Preconfigured Inputs): nel caso in esame è stato scelto di utilizzare un gruppo standard. Confermando la scelta cliccando su Add, sotto la voce del progetto si aggiunge il riferimento al gruppo appena definito, all'interno del quale sono presenti due nuove voci: Alias Devices e il file .sal con lo stesso nome del gruppo che è stato creato, come mostrato in Figura 22.

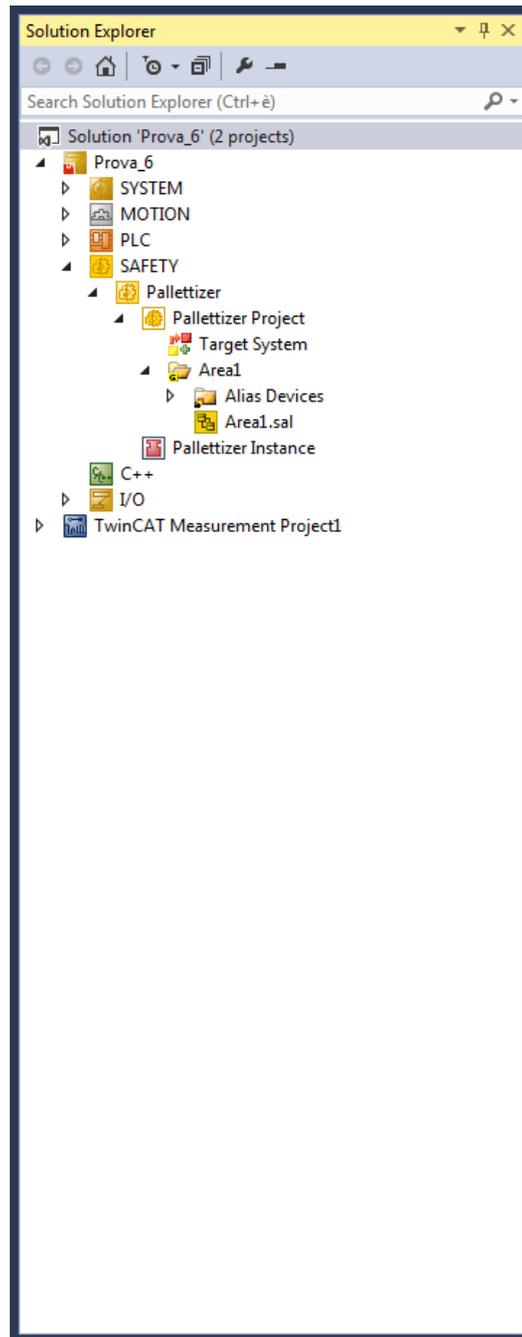


Figura 22: Solution Explorer

Quest'ultimo file è quello che bisogna aprire per poter organizzare i blocchi funzionali. Alias Devices consente invece di mettere in comunicazione il mondo safety con quello PLC: facendo tasto destro sulla dicitura, scegliendo prima Add e poi New Item si accede ad un menù dal quale si possono creare gli elementi di interfaccia che devono poi essere correttamente configurati. Sono proposti ingressi e uscite standard, sia digitali che analogici; si possono anche selezionare le corrispettive entità Safety, configurate di default: selezionato quello desiderato, lo si rinomina e premendo su Add si crea la nuova identità che viene poi utilizzata per la gestione dei blocchi. Facendo doppio click su tale identità si apre la finestra che consente di realizzare il collegamento con la variabile MPS desiderata. In particolare, ciò che lato Safety risulta essere un'uscita, lato MPS è un'entrata e viceversa. I

collegamenti realizzati tra questi due mondi sono univoci, ovvero un ingresso/uscita lato MPS può essere collegato al massimo ad una uscita/ingresso lato Safety. La gestione della comunicazione risulta un po' più complessa se si introducono i moduli I/O Safety: tasto destro sulla voce Alias Devices e scegliere l'opzione Import Alias-Device(s); si apre una finestra in cui vengono elencati tutti i moduli riconosciuti dal sistema. Si selezionano quelli che si desidera utilizzare per il gruppo in sviluppo e si conferma la scelta premendo Ok (vedi Figura 23).

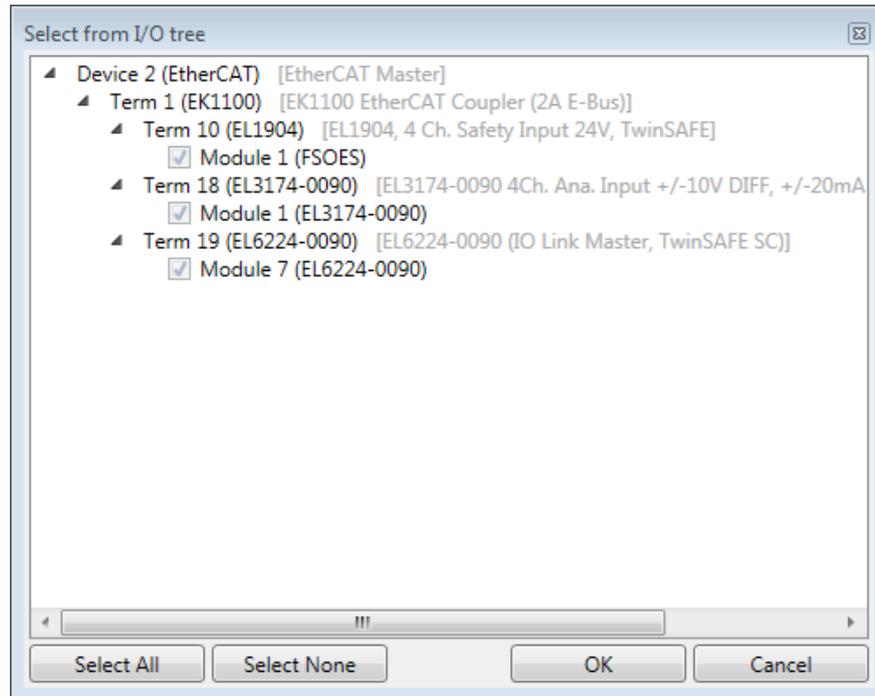


Figura 23: importazione degli alias dei moduli

Per ogni modulo selezionato, sotto la cartella Alias Devices si è generata una voce con lo stesso nome del modulo che rappresenta: cliccandoci sopra si apre una finestra con alcune tendine che permettono di configurarlo.

La prima tendina della finestra, denominata Linking, deve essere già in parte compilata con i collegamenti effettuati automaticamente dal software e comunque non modificabili. L'unico parametro da inserire è FSoE Address, ovvero l'indirizzamento FSoE (Fail Safe Over EtherCAT): qualora sul modulo fisico sia presente il Dip Switch (vedi Figura 24), questo campo va compilato riportando lo stesso valore impostato sul modulo; se ne fosse sprovvisto, si può inserire un valore a piacimento, accertandosi che sia univoco per l'intero progetto.

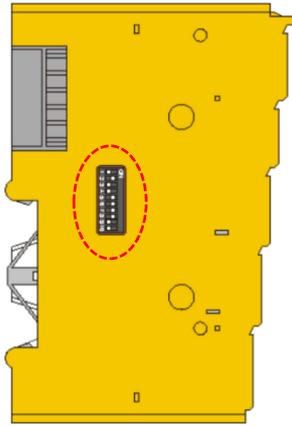


Figura 24: modulo Safety della Beckhoff visto di profilo sul quale è stato evidenziato il Dip Switch

La tendina a lato (Connection) permette di selezionare il canale della rete per la trasmissione dati: in particolare, il parametro Mode deve essere scelto differente per ogni modulo. Per tutti i moduli così configurati, bisogna riportare i valori appena inseriti in FSoE Address e in Mode nelle relative diciture dello stesso modulo del sottomenù I/O, in particolare nella voce TSC Settings della tendina CoE - Online, cosicché si instauri un rapporto biunivoco tra le varie entità create. È poi presente un'altra tendina (Safety Parameters) nella quale vengono riportati indici, nomi e valori di alcuni parametri Safety. Nell'ultima tendina (Process Image) si selezionano gli indirizzi di memoria del modulo che vengono messi a disposizione del programmatore. Per vedere il valore che questi assumono, se la modalità online è attiva, si può fare riferimento alla tendina CoE - Online.

## 3.2. Progetto SAFETY

La configurazione del progetto di sicurezza necessita di variabili di comando che fungono da interfaccia tra il sistema di controllo macchina MPS ed il PLC di sicurezza. Per ogni TwinSafeGroup presente nel progetto sono richieste normalmente 2 variabili di ingresso e 3 variabili di uscita lato PLC Safety.

Variabili d'ingresso:      1)Error Acknowledgment  
                                  2)Run/Stop

Variabili d'uscita:        1)Comunication Error  
                                  2)Function Block Error  
                                  3)Out Error

Le entità appena descritte devono anche essere realizzate lato MPS e successivamente collegate correttamente: in questo caso però si devono introdurre tre variabili d'ingresso (Comm\_Err, FB\_Err, Out\_Err) e due di uscita (Err\_Ack\_Safety, Run\_Safety)

Lo svolgimento del programma Safety prevede di realizzare una logica che mantenga costantemente in sicurezza tutte le aree della macchina. Se una delle condizioni di sicurezza venisse a mancare è necessario che la relativa area oppure l'intera macchina automatica venga messa in sicurezza, ovvero venga sospesa l'alimentazione ai driver e agli azionamenti di potenza.

Per poter realizzare un programma in grado di gestire quanto appena descritto, il software TwinCAT mette a disposizione del programmatore una serie di blocchi funzionali disponibili nella libreria Toolbox.

I blocchi funzionali sono entità software che ricevono in ingresso segnali digitali o analogici e in seguito ad una elaborazione restituiscono una o più uscite. Il valore assunto dalle uscite dipende direttamente dai valori dei segnali di ingresso e da eventuali altri parametri che alcuni blocchi mettono a disposizione del programmatore.

Gli ingressi e le uscite dei vari blocchi funzionali possono essere realizzati secondo due modalità differenti:

- 1) Collegamento con un altro blocco funzionale;
- 2) Collegamento con una variabile Alias.

Se si vuole collegare l'uscita di un blocco funzionale a monte con l'ingresso di un altro a valle bisogna cliccare sul simbolo dell'uscita e, tenendo premuto il tasto sinistro del mouse, spostare il puntatore in corrispondenza dell'ingresso prescelto; per collegare una porta di ingresso/uscita con un alias è sufficiente cliccare a lato di questa e inserire il nome dell'etichetta. Questa viene automaticamente riportata nella finestra Variable Mapping, necessaria per realizzare i collegamenti; grazie a quest'ultima si stabilisce per ogni etichetta l'entità della sezione Alias Devices con la quale bisogna effettuare il collegamento logico.

Durante la fase di programmazione è importante ricordare alcuni limiti e alcune regole legati al collegamento di ingressi e uscite.

È possibile che dall'uscita di un modulo funzionale partano più collegamenti diretti a ingressi differenti, mentre è possibile collegare un solo segnale ad ogni ingresso. Ogni etichetta all'interno del progetto deve essere univoca: perciò se si vuole assegnare lo stesso segnale di ingresso a blocchi differenti, non si possono chiamare i due alias con lo stesso nome ma bisogna aggirare il problema: si riportano di seguito due esempi.

Il primo esempio può essere applicato solamente se il segnale è di tipo booleano: si collega l'ingresso ad un blocco safeAnd, che necessita di almeno due ingressi, al quale si collega poi un altro ingresso fittizio il cui valore viene forzato come sempre attivo. Alternativamente, allo stesso blocco safeAnd si collega il segnale sia alla prima che alla seconda porta di ingresso. L'uscita di tale blocco assume quindi gli stessi valori che assume il segnale di ingresso, negli stessi istanti di tempo; in questo modo però è possibile collegare l'uscita del blocco a più porte di ingresso.

Il secondo esempio è più generale e può essere utilizzato sia per segnali analogici che digitali: per ogni porta di ingresso si utilizza un differente alias, anche se ci si

riferisce allo stesso segnale d'ingresso; quando si compila la Variable Mapping, alle due etichette si collega la stessa porta hardware.

Completata la fase di scrittura del programma, perché questo diventi operativo deve essere scaricato nel PLC, non prima di esser stato revisionato dal software, il quale si accerta che non siano presenti errori sintattici, discordanze, dimenticanze o imprecisioni. Per fare ciò, TwinCAT mette a disposizione del programmatore il toolbox rappresentato in Figura 25.



Figura 25: toolbox di TwinCAT

La prima icona (Verify Safety Project) consente di revisionare l'intero progetto per un check di validità generale a livello software. L'icona a lato (Verify Complete Safety Project) implementa il controllo della prima, facendo anche una revisione a livello hardware. Solamente se da quest'ultima revisione non emergono errori si può procedere con il download del programma nel PLC. Se fossero presenti errori, questi vengono riportati nella Error List: in questa finestra vengono elencati tutti gli errori riscontrati nel programma affiancati da una breve spiegazione che ne specifica la causa. Per esempio il programmatore avrebbe potuto dimenticare di collegare una porta nella finestra Variable Mapping oppure aver collegato un ingresso di tipo booleano ad una porta configurata per ricevere un valore di tipo intero. Inoltre, in questa finestra vengono riportati degli dei messaggi di Warnings, avvertimenti che il software segnala: tali consigli potrebbero migliorare il funzionamento del programma ma non rappresentano alcun limite. Ovvero se in un programma sono presenti degli errori, questo non può in alcun modo esser scaricato nel PLC; se invece sono presenti solamente dei warnings, l'operazione di download è consentita.

Solamente quando tutti gli errori vengo corretti si può procedere con lo scaricamento del programma cliccando l'icona Download Safety Project. Questa apre anzitutto una finestra che chiede al progettista se gli indirizzi dei terminali hardware corrispondono con quelli configurati nel progetto Safety; in seguito ad una attenta verifica si procede dando la conferma. Si apre quindi una nuova finestra nella quale l'operatore deve effettuare il login: questo è necessario di modo che solo le persone autorizzate con Username, Password e numero seriale del dispositivo possano modificare il componente. Confermati i dati, il software permette di scegliere se scaricare l'intero programma o solo alcune parti di esso, nel caso si trattasse di un aggiornamento. Effettuata la scelta, bisogna nuovamente inserire la Password prima che il download venga effettuato.

A lato di questa icona è presente la Delete Safety Project, ovvero l'opzione che consente di cancellare il progetto presente all'interno del PLC.

L'icona raffigurante la chiave inglese è detta Customize Safety Project e serve per personalizzare le impostazioni del progetto

L'icona raffigurante gli occhiali si chiama Show Online Data of Safety Project: solo se il PLC è online e i CRC coincidono, permette di visualizzare lo stato del progetto in esecuzione (RUN, RESET, ERROR), lo stato dei blocchi, quali ingressi/uscite

sono attivi e quali disattivi e di accedere alla funzione Show Online Value, la quale consente di visualizzare il valore che assume ogni segnale. Infine è presente l'icona Simulate Safety Project che consente di simulare il funzionamento del progetto qualora si stesse programmando offline.

ICONA	NOME
	Verify Safety Project
	Verify Complete Safety Project
	Download Safety Project
	Delete Safety Project
	Customize Safety Project
	Show Online Data of Safety Project
	Simulate Safety Project

Tabella 2: icone del toolbox

A lato di questo toolbox se ne trova un altro che riporta la dicitura CRC (Codice di Ridondanza Ciclica) a cui fanno seguito tre codici alfanumerici. Ogni volta che il programmatore effettua una modifica al progetto, TwinCAT assegna un nuovo codice al progetto stesso. Questo è un dispositivo di sicurezza utilizzato per garantire che il progetto sul monitor del PC coincida in ogni suo particolare alla versione scaricata nel PLC e alla sua versione offline. Affinché venga assegnato un differente CRC all'intero progetto è sufficiente che sia modificato anche un solo carattere di un nome di un'etichetta. Il primo codice rappresenta il CRC del progetto online, il secondo rappresenta il CRC scaricato nel PLC mentre l'ultimo è associato al CRC offline. Solo quando i tre codici coincidono il progetto può entrare in modalità RUN e si può attivare l'icona Show Online Data of Safety Project. Se i tre non coincidono, bisogna effettuare un salvataggio del progetto e effettuare il download di questo nel PLC.

## 4. BECKHOFF E BUS DI CAMPO

### 4.1. Rete di trasmissione dati EtherCAT

Lo sviluppo tecnologico ha portato a realizzare macchine automatiche sempre più complesse, sia a livello hardware che soprattutto software: più la macchina è complessa e di dimensioni rilevanti, maggiore è il flusso di informazioni che bisogna gestire. In seguito ad un approfondito studio dei bus di campo, a fronte di una significativa mole di dati, la Marchesini Group ha scelto di servirsi del mondo EtherCAT: questo permette di organizzare e gestire tutta l'elettronica a bordo macchina utilizzando i terminali realizzati da Beckhoff per le reti fieldbus.

In Figura 26 viene presentato lo schema dell'infrastruttura elettronica di una macchina automatica.

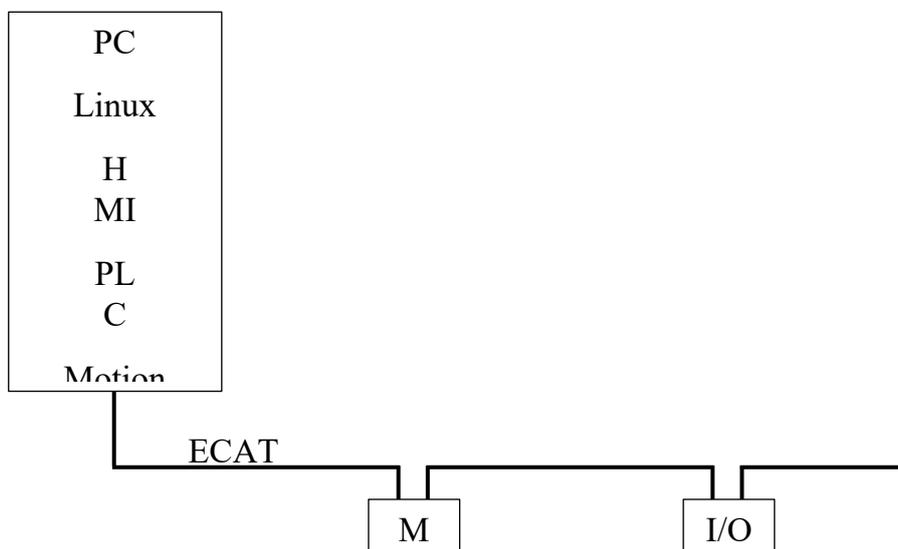


Figura 26: infrastruttura elettronica di una macchina automatica

La rete ETHERCAT è stata sviluppata secondo la filosofia già presente nel bus di campo chiamato dell'INTERBUS: il master emette un pacchetto con le istruzioni di comando per tutti gli slave; ogni slave riceve l'intero pacchetto, dal quale estrapola solo le informazioni a esso riservate e restituisce i dati richiesti dal master. Dopodiché passa il pacchetto allo slave successivo, che si comporterà come il primo. Nel pacchetto dati è inserita anche un'informazione di clock al fine di garantire la sincronizzazione e la contemporaneità delle operazioni. In caso di reti che necessitano di sincronizzazioni molto spinte, si utilizza il metodo dei clock distribuiti: il master invia i pacchetti ai vari slave, che ricevono ma non eseguono finché l'ultimo dell'anello non ha ricevuto il pacchetto. Si innesca un ritardo di esecuzione pari al tempo di trasmissione dei comandi dal master all'ultimo slave.

## 4.2. IO LINK

Il protocollo di comunicazione *IO Link* è lo strumento che regola la comunicazione tra i PLC ed i sensori/attuatori. Questo protocollo di comunicazione non sostituisce il vecchio sistema di comunicazione che si realizzava tramite i bus di campo ma lo integra, migliorandone svariati aspetti:

1. si ha un cablaggio standardizzato di tutti i componenti. Ciò si realizza grazie ad un cavo standard costituito da quattro pin, due dei quali necessari per l'alimentazione del dispositivo, uno destinato alla trasmissione del segnale ed uno lasciato libero, come mostrato in Figura 27. Si ha una semplificazione sia della fase di progettazione che nella fase di cablaggio della macchina. Inoltre, questo permette di avere cavi standard, utilizzabili per una svariata quantità di sensori anche molto differenti tra loro il che comporta la limitazione delle scorte in magazzino di questi componenti;

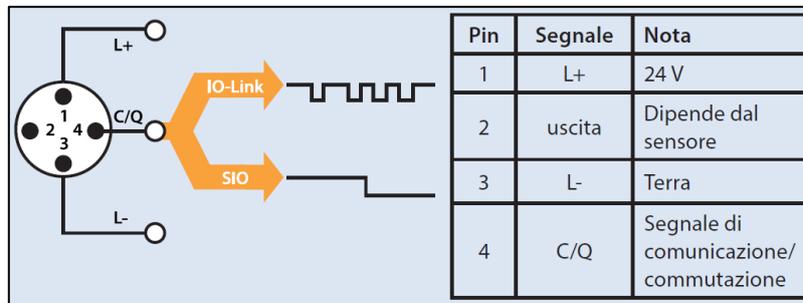


Figura 27: schematizzazione di un cavo standard per I/O Link

2. si hanno a disposizione i dati utilizzati dal sensore: questo permette di elaborarli per migliorare i processi produttivi in fase di produzione e ottimizzare la manutenzione ordinaria della macchina;
3. permette di gestire la configurazione ed il monitoraggio dei dispositivi direttamente da remoto. Ciò consente anche, in fase di sostituzione di un hardware, di scaricare l'attuale configurazione software in una memoria di appoggio per poterla poi ricaricare successivamente sul nuovo dispositivo installato;
4. consente la visibilità degli stati di ogni dispositivo che, in simbiosi con un'avanzata diagnostica, permette di individuare sensori malfunzionanti e linee di comunicazione danneggiate.

## 4.3. Banco prova

Per lo sviluppo e l'approfondimento della tesi è stato utilizzato un piccolo banco prova, rappresentato in Figura 28, dotato di:

- un PLC Beckhoff a cui sono stati collegati diversi moduli, di cui si approfondisce in seguito;
- alcuni pulsanti sia bistabili che monostabili (sono stati utilizzati sia nella configurazione normalmente aperti che in quella normalmente chiusi);
- un sensore ultrasonico e una fotocellula retro-reflex per rilevamento della presenza/assenza del prodotto;
- due encoder, uno lineare e uno rotativo;
- un motore elettrico ( $C0 = 1,2 \text{ Nm}$ ,  $\text{rpm} = 8000 \text{ rpm}$ )
- cavi elettrici per realizzare i collegamenti.



Figura 28: Banco prova utilizzato per i test

La base del PLC utilizzato è EK1100, mentre i moduli utilizzati sono i seguenti:

- EL1004, modulo per gli ingressi digitali;
- EL1904, modulo safety per gli ingressi digitali;
- EL2004, modulo per le uscite digitali;
- EL3124, modulo per gli ingressi analogici;
- EL3162, modulo per gli ingressi analogici;
- EL3174-0090, modulo per gli ingressi analogici;
- EL6224-0090, modulo di comunicazione;
- EL6910, modulo safety di comunicazione;
- EL7211-9014, modulo motion.

Questi comunicano tra loro grazie alla rete di trasmissione dati EtherCAT, configurata con il software TwinCAT come descritto nei paragrafi precedenti.

La strumentazione è servita per testare il funzionamento dei vari blocchi strumento di studio e per simulare il comportamento di semplici gruppi di macchine reali.

## 5. BLOCCHI FUNZIONALI

Durante l'attività di tesi si è studiato il funzionamento di tutti i blocchi funzionali (Function Blocks, abbreviato FB) che il software TwinCAT dispone nella propria libreria. Per poterli utilizzare è necessario aprire il file .sal che è stato generato durante la creazione del progetto. Nella finestra centrale del programma si apre l'area di programmazione con il primo Network già impostato mentre a lato di questa si predispone una ulteriore finestra con due menù differenti.

Il primo menù è il Toolbox, la libreria che contiene tutti i blocchi funzionali, dalla quale bisogna attingerli per realizzare il progetto Safety. Come si vede dall'immagine Figura 29 è suddiviso in più sottogruppi. In ognuno di questi è presente la voce Pointer che permette di selezionare il puntatore standard del PC.

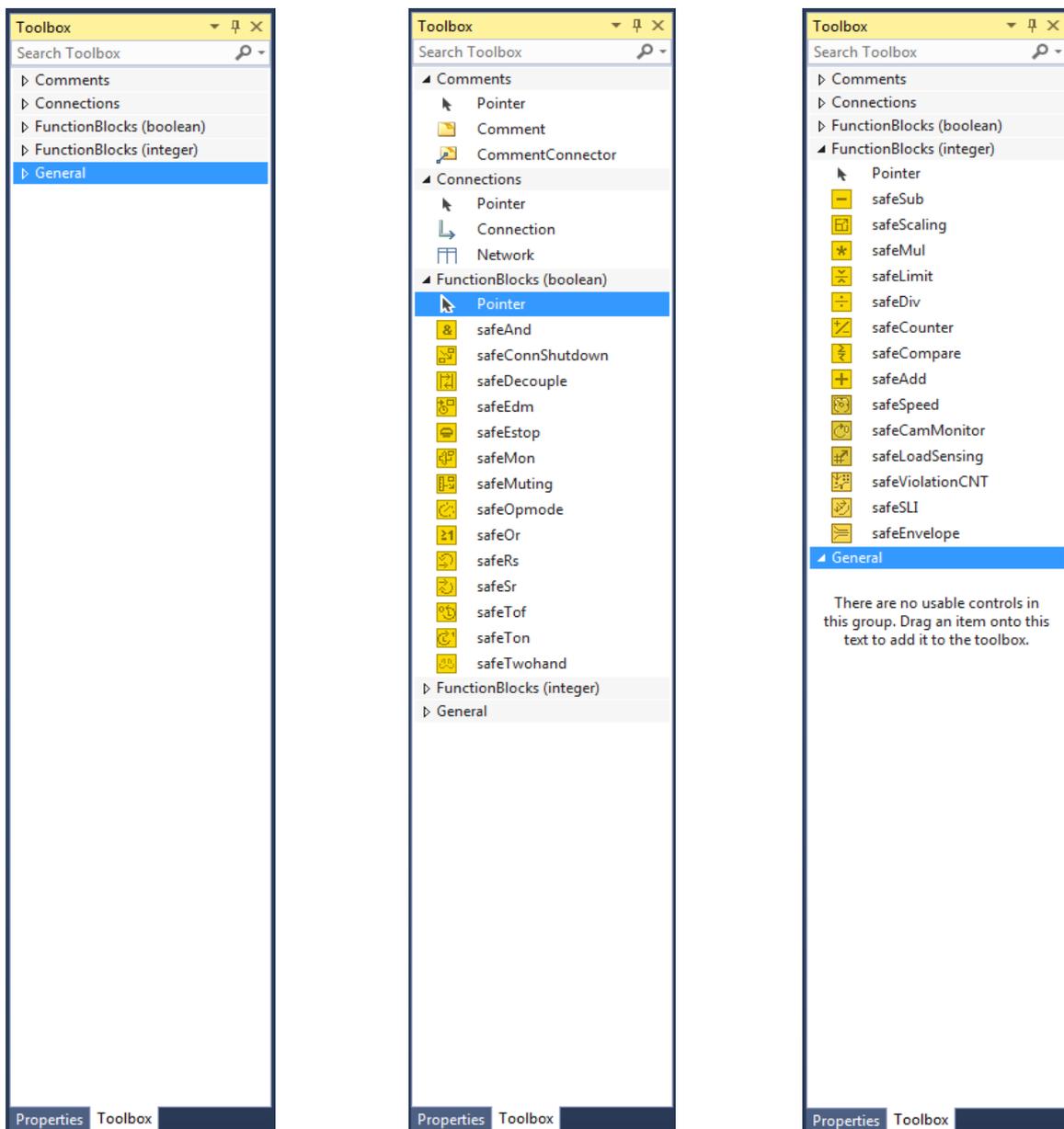


Figura 29: Toolbox

Il primo sottogruppo è il Comments e permette di gestire tutte le operazioni correlate ai commenti che possono essere apposti direttamente sui FB. Il successivo, Connections, consente di gestire i Network del progetto, ovvero quelle aree di lavoro adibite a contenere i blocchi funzionali. Il terzo sottogruppo è FunctionBlocks(boolean) e contiene tutti quei blocchi che lavorano con segnali booleani, che quindi realizzano funzioni logiche come l'And e l'Or. Nel quarto, denominato FunctionBlocks(integer), sono presenti tutti quei FB che gestiscono segnali analogici, ovvero segnali caratterizzati da un campo di variabilità e non semplici TRUE/FALSE. Infine nel sottogruppo denominato General è possibile visualizzare altri FB introdotti dal programmatore.

Per poter realizzare il progetto bisogna selezionare il FB desiderato dal sottogruppo di appartenenza e trascinarlo nella corretta area di lavoro; una volta rilasciato, si provvede a configurarlo e a realizzare i dovuti collegamenti.

Il secondo menù della finestra in esame ha il nome Properties: all'interno di questo si possono modificare tutte le impostazioni relative al blocco funzionale. Per esempio, preso un qualsiasi FB, si può impostare il nome oppure visualizzare lo stato dei suoi ingressi; se si clicca su uno di questi, si può scegliere se abilitare o disabilitare l'ingresso in questione.

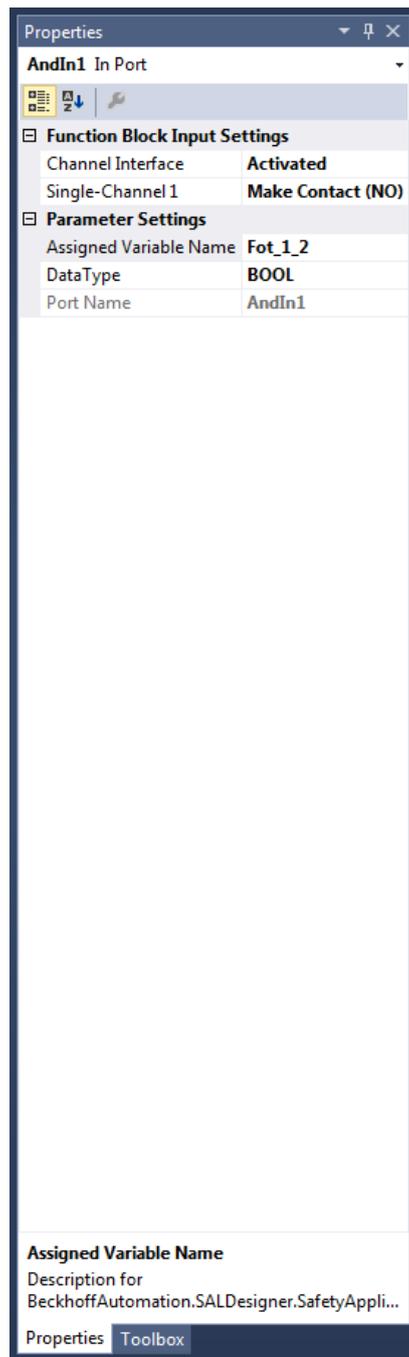


Figura 30: Properties

A seconda che il blocco operi con ingressi digitali o analogici si hanno a disposizione differenti personalizzazioni.

Per i blocchi appartenenti al sottogruppo FunctionBlocks(boolean), questo menù permette di specificare se il canale abilitato sia un normalmente aperto (NO, Make Contact) oppure un normalmente chiuso (NC, Break Contact). Quest'ultima modalità elabora il segnale d'ingresso come lo riceve, senza apportargli delle modifiche; la modalità normalmente aperto invece elabora il segnale d'ingresso solamente dopo averlo negato. Inoltre, per i blocchi appartenenti a questo sottogruppo, non è possibile modificare la tipologia del segnale d'ingresso, in quanto l'unica opzione consentita è BOOL.

Per i blocchi appartenenti all'altro sottogruppo FunctionBlocks(integer) invece, normalmente non è concessa la scelta tra NO oppure NC, in quanto si sottintende che siano tutti NC. Esistono però delle eccezioni: alcuni FB permettono al programmatore di operare questa scelta; se si imposta il canale di ingresso come NO, il blocco elabora il segnale analogico dopo averne cambiato il segno (inverte il positivo con il negativo e viceversa).

Per questa categoria di FB è importate invece specificare la tipologia del segnale in ingresso nella porta: questa deve coincidere con la tipologia dell'area di memoria da cui la variabile viene letta. Volendo essere più chiari, si procede con il seguente esempio: un encoder, tra le altre informazioni, riporta anche quella relativa alla posizione dell'asse. Questo dato viene costantemente aggiornato, salvato all'interno di uno o più byte di memoria e si supponga che ad esso venga associata la tipologia UINT; anche la porta di ingresso del blocco funzionale deve essere riportata la stessa informazione e l'ingresso deve essere impostato sulla tipologia UINT. In fase di compilazione, se il compilatore rileva una discrepanza tra la tipologia di un segnale d'ingresso e la porta adibita a riceverlo riporta un messaggio di errore. Inoltre esistono dei blocchi funzionali che riportano in uscita alcuni segnali che ricevono in ingresso: la tipologia di queste uscite non è legata a quella degli ingressi in quanto i FB sono in grado di fare conversioni. Se lo stesso ingresso citato in precedenza venisse reso disponibile anche all'uscita del blocco funzionale che lo elabora, si può decidere quale tipologia assegnare al segnale in questione.

La differenza tra le varie tipologie è da ricercare nella dimensione delle aree di memoria riservate per la variabile associata e nel fatto che sia una variabile in grado di assumere valori solo positivi piuttosto che valori sia positivi che negativi. In Tabella 3 vengono elencate le quattro tipologie di segnali consentite:

Nome	Descrizione
INT	16 bit con segno
UINT	16 bit senza segno
DINT	32 bit con segno
UDINT	32 bit senza segno

Tabella 3: tipologie di segnali analogici

Come già descritto pocanzi, i blocchi funzionali possono essere classificati in base alla tipologia di segnali che elaborano.

Nella categoria dei FunctionBlocks(boolean) si trovano:

- safeAnd
- safeOr
- safeOpmode
- safeEstop
- safeMon
- safeDecouple
- safeTwohand
- safeMuting
- safeEdm
- safeRs
- safeSr
- safeTon
- safeTof
- safeConnShutdown

Nella categoria dei FunctionBlocks(integer) sono presenti:

- safeAdd
- safeSub
- safeMul
- safeDiv
- safeCompare
- safeLimit
- safeCounter
- safeScaling
- safeSpeed
- safeLoadSensing
- safeCamMonitor
- safeSLI
- safeEnvelope
- safeViolationCNT

Di seguito vengono descritti tutti i blocchi funzionali messi a disposizione da TwinCAT: per ognuno viene approfondito il funzionamento e vengono studiate le singole porte di ingresso e di uscita.

## 5.1. AND

<b>FB AND</b>	
Principio di base	Il blocco funzionale safeAnd attiva la propria uscita solo quando tutti gli ingressi sono attivi.
Funzionamento	Tale blocco riceve in ingresso fino ad un massimo di otto segnali: solo quando tutti i segnali di input sono attivi l'uscita passa nello stato logico TRUE.
Ingressi	<p>AndIn(x): porte di ingresso dei segnali. Tutti gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul> <p>Fa eccezione la porta d'ingresso AndIn1 in cui è consentito anche l'ingresso di una variabile standard del PLC (l'output nel PLC %Q*)</p>
Uscite	<p>AndOut: porta di uscita del segnale. L'uscita del blocco funzionale può essere collegata a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>

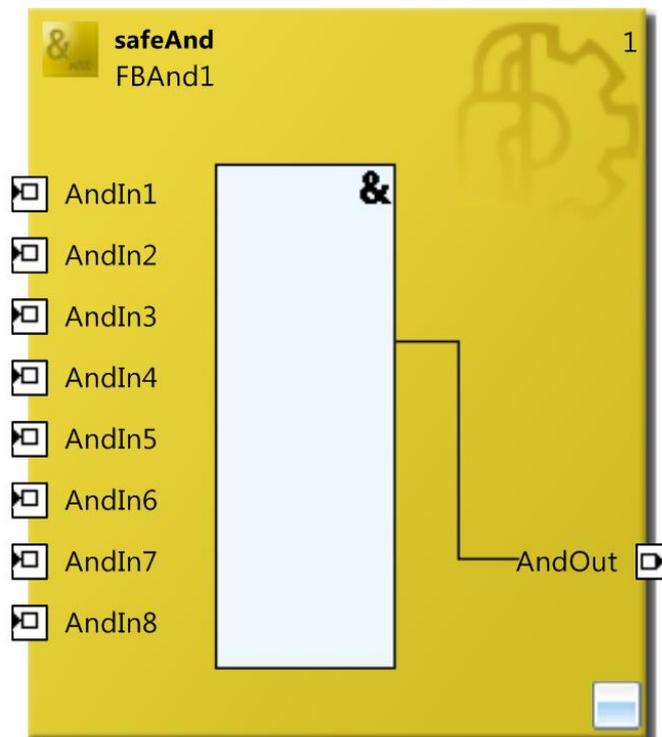


Figura 31: FB safeAnd

## 5.2. OR

<b>FB OR</b>	
Principio di base	Il blocco funzionale safeOr attiva la propria uscita quando almeno un ingresso è attivo.
Funzionamento	Tale blocco riceve in ingresso fino ad un massimo di otto segnali: fintanto che almeno un ingresso rimane attivo l'uscita rimane nello stato logico TRUE.
Ingressi	OrIn(x): porte di ingresso dei segnali. Tutti gli ingressi del blocco funzionale possono essere collegati a: <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul>
Uscite	OrOut: porta di uscita del segnale. L'uscita del blocco funzionale può essere collegata a: <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>

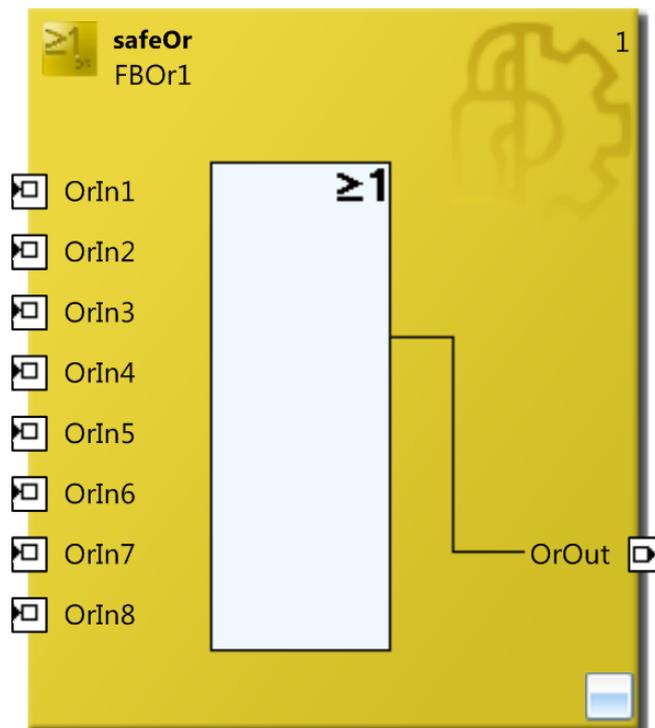


Figura 32: FB safeOr

### 5.3. OPMODE

<b>FB OPMODE</b>	
Principio di base	Il blocco funzionale safeOpmode viene utilizzato per gestire il passaggio tra differenti modalità operative.
Funzionamento	<p>Tale blocco riceve in ingresso fino ad un massimo di otto segnali e associa ad ognuno di essi una uscita. Se nessun segnale d'ingresso è attivo oppure ne arrivano due o più si disattivano tutte le uscite, tranne Error che passa nello stato logico TRUE. Se arriva un solo segnale di ingresso, in seguito ad una sequenza di FALSE→TRUE→FALSE realizzata all'ingresso Restar, si attiva la corrispondente uscita.</p> <p>Il tempo di discrepanza permette di monitorare il cambio delle configurazioni del blocco: se per un tempo inferiore a quello indicato nella finestra Discrepancy (ms) si verifica la situazione per cui non si hanno ingressi attivi o se ne hanno più di uno, il blocco non va in errore.</p>
Ingressi	<p>Restart: ingresso utilizzato per uscire dalla condizione di errore e per attivare l'uscita a fronte di un singolo segnale di ingresso. Affinché ciò si realizzi, questo ingresso deve verificare la sequenza di segnali FALSE→TRUE→FALSE. Tale ingresso può essere collegato ad una porta input di un modulo TwinSAFE (per esempio EL1904), ad un'uscita di un altro blocco funzionale oppure ad una variabile standard del PLC (l'output nel PLC %Q*).</p> <p>OpIn(x): porte di ingresso dei segnali.</p> <p>Questi ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul>
Uscite	<p>Error: tale uscita può essere attivata dalle situazioni sopra descritte.</p> <p>OrOut: porta di uscita del segnale.</p> <p>Tutte le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	<p>Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p> <p>Per questo blocco funzionale non è possibile impostare degli ingressi come moduli bicanale.</p>

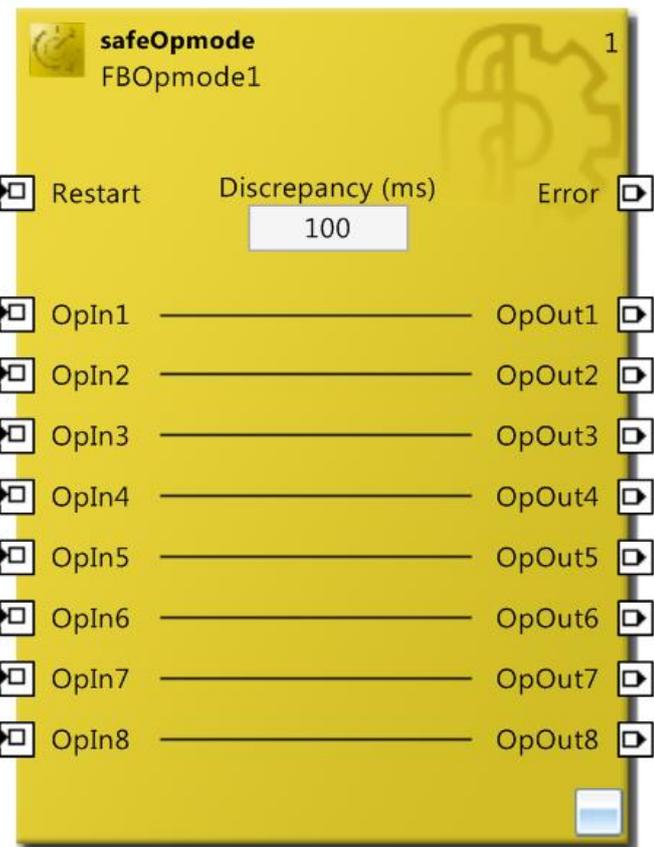


Figura 33: FB safeOpmode

## 5.4. ESTOP

<b>FB ESTOP</b>	
Principio di base	Il blocco funzionale safeEstop attiva le sue uscite solo quando tutti gli ingressi abilitati sono attivi. L'impiego classico per questo blocco funzionale è il riarmo delle macchine: agli ingressi si collegano tutte le sicurezze dei carter e solo quando sono tutte attive contemporaneamente si può attivare l'alimentazione della macchina.
Funzionamento	Tale blocco riceve in ingresso fino ad un massimo di otto segnali (EStopIn1- EStopIn8), oltre il Restart e i due segnali di retroazione EDM dei segnali di uscita. Solo quando tutti i segnali EStopIn abilitati sono attivi, la sequenza FALSE→TRUE→FALSE sull'ingresso Restart permette alle uscite EStopOut e EStopDelOut di passare nello stato logico TRUE. Queste rimangono attive fintanto che un ingresso non richiede lo stato di sicurezza: non appena ciò si verifica l'uscita EStopOut si porta immediatamente nello stato logico FALSE, mentre l'uscita EStopDelOut raggiunge lo stesso stato con un tempo di ritardo pari a quello specificato nella finestra Delay Time (ms). Gli ingressi EDM servono per gestire il circuito di retroazione dei segnali d'uscita.
Ingressi	Restart: ingresso utilizzato sia per attivare le uscite EStopOut e EStopDelOut quando tutti gli ingressi abilitati sono attivi che per uscire dalla condizione di errore. Perché ciò avvenga questo ingresso deve verificare la sequenza di segnali FALSE→TRUE→FALSE. EStopIn(x): porte di ingresso dei segnali. EDM1 e EDM2: tali ingressi sono destinati al circuito di retroazione del segnale di uscita. Tutti gli ingressi del blocco funzionale possono essere collegati a: <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul> Inoltre, gli ingressi Restart e EDM possono anche essere collegati a una variabile standard del PLC (l'output nel PLC %Q*)
Uscite	Error: uscita che si attiva se il blocco va nello stato di errore. EStopOut: uscita che si disattiva non appena un ingresso richiede lo stato di sicurezza. EStopDelOut: uscita che si disattiva con un ritardo pari a Delay Time dopo che un ingresso richiede lo stato di sicurezza. Tutte le uscite del blocco funzionale possono essere collegate a:

	<ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	<p>Una o più coppie di ingressi (EStopIn1 e EStopIn2, EStopIn3 e EStopIn4, EStopIn5 e EStopIn6, EStopIn7 e EStopIn8) possono essere accoppiate con un tempo di discrepanza, impostabile direttamente all'interno della finestra del blocco funzionale. Qualora uno dei due segnali d'ingresso richieda lo stato di sicurezza, anche il secondo ingresso deve riportare la stessa informazione, rispettando il tempo di discrepanza: se ciò non avviene, il blocco entra nello stato di errore, non prima di aver disattivato tutte le uscite.</p> <p>Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p>

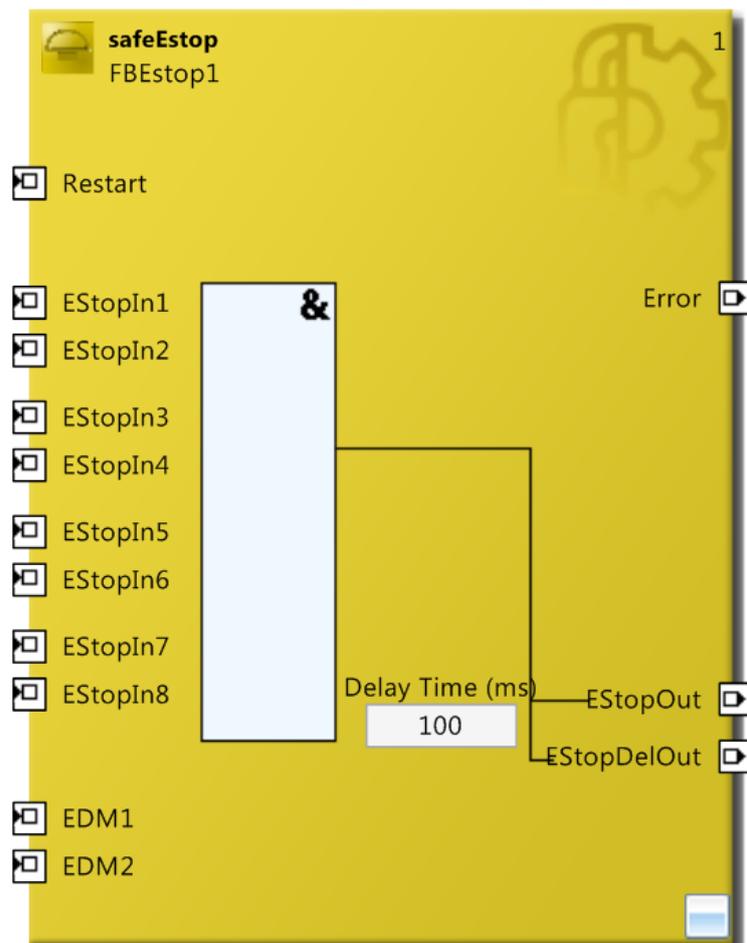


Figura 34: FB safeEstop

## 5.5. MON

<b>FB MON</b>	
Principio di base	Il blocco funzionale safeMon attiva le sue uscite solo quando tutti gli ingressi MonIn abilitati sono attivi. Inoltre sono previsti due ingressi di sicurezza che permettono di bypassare i segnali di ingresso citati pocanzi.
Funzionamento	Tale blocco riceve in ingresso fino ad un massimo di quattro segnali (MonIn1- MonIn4), oltre il Restart, i due segnali di sicurezza Secure e i due segnali di retroazione EDM dei segnali di uscita. Solo quando tutti i segnali MonIn abilitati sono attivi, la sequenza FALSE→TRUE→FALSE sull'ingresso Restart permette alle uscite MonOut e MonDelOut di passare nello stato logico TRUE. Queste rimangono attive fintanto che un ingresso non richiede lo stato di sicurezza: non appena ciò si verifica, l'uscita MonOut passa immediatamente nello stato logico FALSE, mentre l'uscita MonDelOut raggiunge lo stesso stato con un tempo di ritardo pari a quello specificato nella finestra Delay Time (ms). Sono inoltre previsti due ingressi di sicurezza Secure in grado di bypassare i segnali MonIn; si può abilitare il singolo ingresso oppure entrambi. Quando la combinazione logica tra questi uno/due segnali di ingresso è TRUE, le uscite del blocco funzionale non vengo disabilitate anche se un ingresso MonIn richiede lo stato di sicurezza. Gli ingressi EDM servono per gestire il circuito di retroazione dei segnali d'uscita.
Ingressi	Restart: ingresso utilizzato sia per attivare le uscite MonOut e MonDelOut quando tutti gli ingressi abilitati sono attivi che per uscire dalla condizione di errore. Perché ciò avvenga questo ingresso deve verificare la sequenza di segnali FALSE→TRUE→FALSE. MonIn(x): porte di ingresso dei segnali. EDM(x): tali ingressi sono destinati al circuito di retroazione del segnale di uscita. Tutti gli ingressi del blocco funzionale possono essere collegati a: <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul> Inoltre, gli ingressi Restart e EDM possono anche essere collegati a una variabile standard del PLC (l'output nel PLC %Q*).
Uscite	Error: uscita che si attiva se il blocco va nello stato di errore. EStopOut: uscita che si disattiva non appena un ingresso richiede lo stato di sicurezza. EStopDelOut: uscita che si disattiva con un ritardo pari a Delay Time dopo che un ingresso richiede lo stato di sicurezza.

	<p>Tutte le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	<p>Una o più coppie di ingressi (MonIn1 e MonIn2, MonIn3 e MonIn4, Secure1 e Secure2) possono essere accoppiate con un tempo di discrepanza, impostabile direttamente all'interno della finestra del blocco funzionale. Qualora uno dei due segnali modifichi il proprio stato logico, anche il secondo ingresso deve riportare la stessa informazione, rispettando il tempo di discrepanza: se ciò non avviene, il blocco entra nello stato di errore, non prima di aver disattivato tutte le uscite. Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p> <p>Il blocco funzionale safeMon quando ha gli ingressi Secure disattivati si comporta esattamente come il blocco funzionale safeEStop.</p>

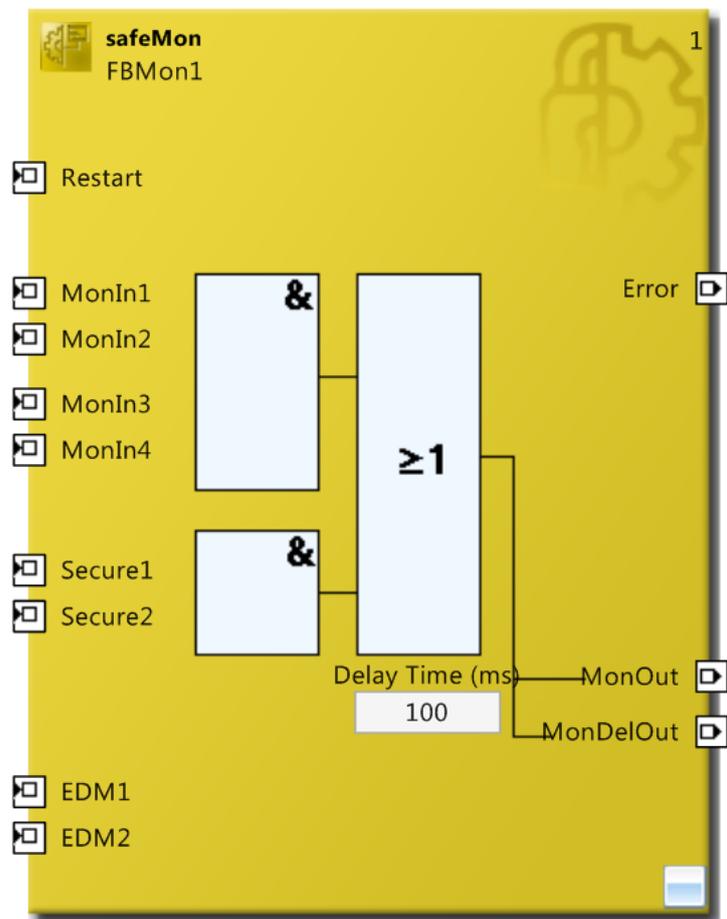


Figura 35: FB safeMon

## 5.6. DECOUPLE

<b>FB DECOUPLE</b>	
Principio di base	Il blocco funzionale safeDecouple permette di disaccoppiare i segnali di una TwinSAFE Connection, ovvero permette di collegare uno stesso ingresso a più TwinSAFE Group.
Funzionamento	Il blocco, costituito da otto ingressi e otto uscite collegati singolarmente uno ad uno, attiva l'uscita non appena il corrispondente ingresso passa allo stato logico TRUE e viceversa. Il blocco è pensato per offrire al programmatore la possibilità di portare il segnale di ingresso a più di un FB: si collega la singola uscita agli ingressi di differenti blocchi funzionali cosicché non appena il corrispondente ingresso cambia il proprio stato logico, cambiano anche tutte le uscite ad esso associate
Ingressi	DecIn(x): porte di ingresso dei segnali che si vuole collegare a diversi FB. Tutti gli ingressi del blocco funzionale possono essere collegati a: <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul>
Uscite	DecOut(x): porte di uscita dei segnali. Entrambe le uscite del blocco funzionale possono essere collegate a: <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>

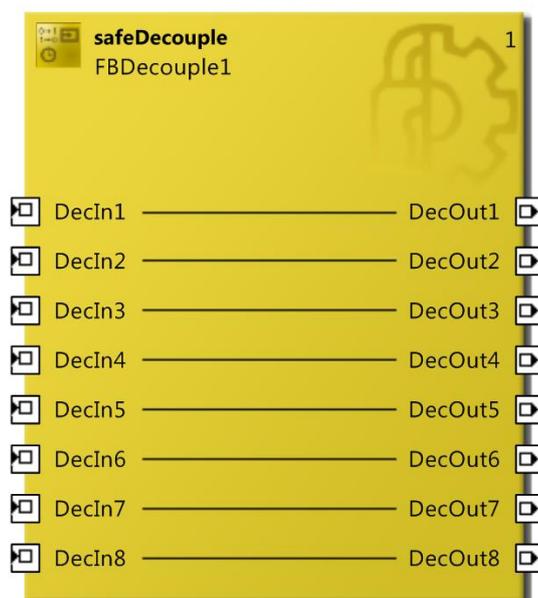


Figura 36: FB safeDecouple

## 5.7. TWOHAND

<b>FB TWOHAND</b>	
Principio di base	Il blocco funzionale safeTwohand permette di sincronizzare degli eventi.
Funzionamento	<p>Tale blocco riceve in ingresso un massimo di 4 segnali: questi realizzano una funzione logica di AND due a due. Se entrambi i sottogruppi cambiano lo stato logico a TRUE in un intervallo di tempo inferiore a quello indicato nella finestra Discrepancy between groups (ms) si attiva l'uscita TwoHandOut. Con tale uscita attiva, se un segnale d'ingresso dovesse commutare nello stato logico FALSE, anche la stessa uscita TwoHandOut viene disattivata.</p> <p>Qualora, a fronte della commutazione allo stato logico di TRUE del primo sottoblocco, il secondo sottoblocco non commuti entro il tempo utile, il blocco safeTwohand non attiva nessuna uscita e prima di poter verificare una successiva sincronizzazione, tutti gli ingressi devono trovarsi contemporaneamente nello stato logico FALSE.</p> <p>Nel momento in cui si attiva l'uscita TwoHandOut, se un segnale d'ingresso dovesse commutare nello stato logico FALSE, anche la stessa uscita viene disattivata.</p> <p>Gli ingressi di questo blocco funzionale possono essere anche accoppiati, generando un ingresso bicanale, con relativo tempo di discrepanza: qualora questo tempo venisse violato, il blocco entra nello stato di errore, mettendo in condizioni di sicurezza tutte le uscite.</p>
Ingressi	<p>TwoHand(x): segnali di ingresso.</p> <p>Tutti gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul>
Uscite	<p>Error: tale uscita può essere attivata dalla situazione sopra descritta.</p> <p>TwoHandOut: uscita che si attiva quando si realizza la sincronizzazione dei segnali di ingresso.</p> <p>Entrambe le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	Le coppie di ingressi (TwoHand1 e TwoHand2, TwoHand3 e TwoHand4) possono essere accoppiate con un tempo di discrepanza, impostabile direttamente all'interno della finestra

del blocco funzionale. Qualora uno dei due segnali modifichi il proprio stato logico, anche il secondo ingresso deve riportare la stessa informazione, rispettando il tempo di discrepanza: se ciò non avviene, il blocco entra nello stato di errore, non prima di aver disattivato tutte le uscite. Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.

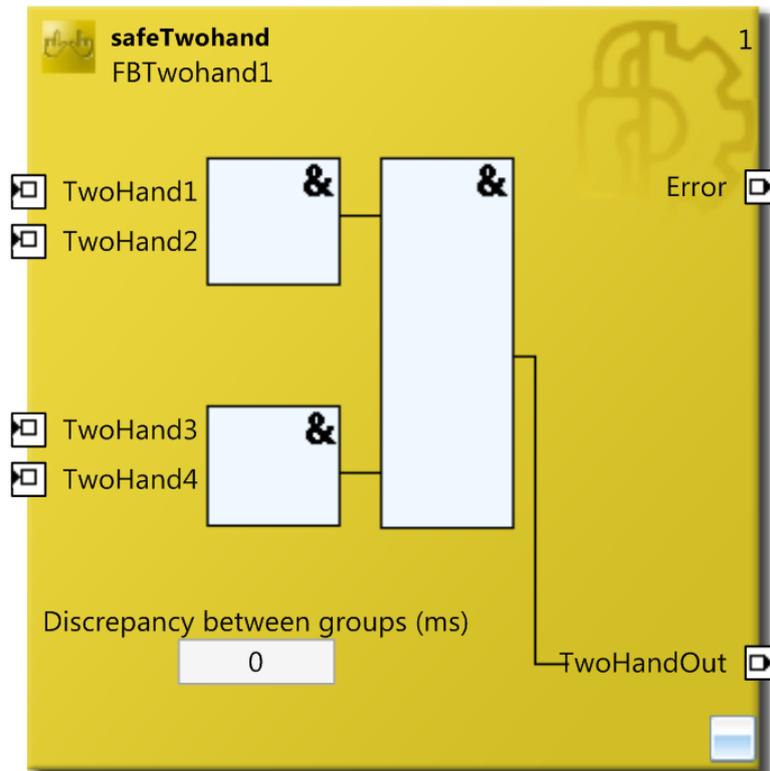


Figura 37: FB safeTwohand

## 5.8. MUTING

<b>FB MUTING</b>	
Principio di base	Il blocco funzionale safeMuting permette realizzare una soppressione conforme della funzione di protezione, ad esempio per trasportare del materiale in/da una area protetta.
Funzionamento	<p>Tale blocco si abilita quando riceve il segnale di consenso, ovvero quando l'ingresso Enable passa allo stato logico TRUE. Quando ciò avviene, gli ingressi OSSDIn devono già esser impostati allo stato logico TRUE. Non appena l'ingresso Muting1 passa allo stato logico TRUE si attiva l'uscita MutingActive: questa uscita rimane attiva se si verifica la seguente successione di eventi:</p> <ol style="list-style-type: none"> <li>1) attivazione ingresso Muting1</li> <li>2) attivazione ingresso Muting2</li> <li>3) disattivazione ingresso OSSDIn1</li> <li>4) disattivazione ingresso OSSDIn2</li> <li>5) attivazione ingresso Muting3</li> <li>6) attivazione ingresso Muting4</li> <li>7) disattivazione ingresso Muting1</li> <li>8) disattivazione ingresso Muting2</li> <li>9) riattivazione ingresso OSSDIn1</li> <li>10)riattivazione ingresso OSSDIn2</li> <li>11)disattivazione ingresso Muting3</li> <li>12)disattivazione ingresso Muting4</li> </ol> <p>Sul fronte di discesa della disattivazione dell'ingresso Muting4 si disattiva anche l'uscita MutingActive.</p> <p>Se con il segnale di Enable attivo tale ordine non viene rispettato, si disattiva l'uscita MutingActive e il blocco si porta nello stato di errore, attivando l'uscita Error e mettendo nello stato di sicurezza tutte le altre uscite.</p> <p>L'uscita MuteOut riporta lo stato logico degli ingressi OSSDIn (che solitamente vengono uniti come ingresso bicanale di un singolo elemento fisico e ai quali si assegna un tempo di discrepanza); l'uscita MuteDelOut riporta lo stesso segnale, ritardato però di un tempo equivalente a quello specificato nella finestra Output Delay Time.</p> <p>Gli ingressi EDM servono per gestire il circuito di retroazione dei segnali d'uscita.</p> <p>La finestra Max. Muting Time (minutes) prevede la possibilità di impostare il tempo massimo per realizzare la sequenza corretta di eventi: qualora questo tempo venisse superato, anche se l'ordine degli eventi è stato rispettato, il blocco si porta nella condizione di errore.</p> <p>La finestra Filter Time Muting (ms) permette di impostare un filtro di massimo 500 ms per gli ingressi Muting.</p>

	<p>Il blocco funzionale propone anche tre funzioni aggiuntive che possono essere attivate. La finestra Sequential Inputs permette di disabilitare gli ingressi Muting3 e Muting4 quando gli ingressi Muting1, Muting2, OSSDIn1 e OSSDIn2 sono abilitati e di utilizzare questi ultimi come quattro segnali provenienti da dispositivi differenti.</p> <p>La finestra Backwards Mode permette di realizzare la sequenza anche in ordine inverso: quando questa opzione è spuntata, a seguito del segnale di abilitazione Enable se il primo segnale di ingresso proviene dal Muting1 si segue l'ordine standard (da 1 a 12); se invece il primo segnale di ingresso arriva dal Muting4 si segue l'ordine inverso (da 12 a 1).</p> <p>La finestra OSSD Check Optional permette di escludere dal controllo i segnali provenienti dagli ingressi OSSDIn, andando quindi a controllare solo quelli provenienti dai segnali Muting.</p>
Ingressi	<p>Enable: porta di ingresso del segnale di abilitazione del blocco.</p> <p>Muting(x): porte di ingresso dei segnali.</p> <p>OSSDIn(x): ingressi dei segnali provenienti dal dispositivo di protezione (AOPD-Active Opto-electronic Protection Device), come per esempio una barriera fotoelettrica.</p> <p>EDM(x): tali ingressi sono destinati al circuito di retroazione del segnale di uscita.</p> <p>Tutti gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul> <p>Gli ingressi Enable e EDM(x) possono anche essere collegati a una variabile standard del PLC (l'output nel PLC %Q*).</p>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione sopra descritta. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate.</p> <p>MutingActive: uscita che rimane attiva finché viene rispettata la corretta sequenza e non viene superato il Max. Muting Time (minutes).</p> <p>MuteOut: riporta in uscita lo stato logico degli ingressi OSSDIn</p> <p>MuteDelOut: riporta in uscita lo stato logico degli ingressi OSSDIn con un tempo di ritardo specificato nella finestra Delay Time (ms)</p> <p>Tutte le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>

Note	<p>Le coppie di ingressi (Muting1 e Muting2, OSSDIn1 e OSSDIn2, Muting3 e Muting4) possono essere accoppiate con un tempo di discrepanza, impostabile direttamente all'interno della finestra del blocco funzionale. Qualora uno dei due segnali modifichi il proprio stato logico, anche il secondo ingresso deve riportare la stessa informazione, rispettando il tempo di discrepanza: se ciò non avviene, il blocco entra nello stato di errore, non prima di aver disattivato tutte le uscite. Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p>
------	---

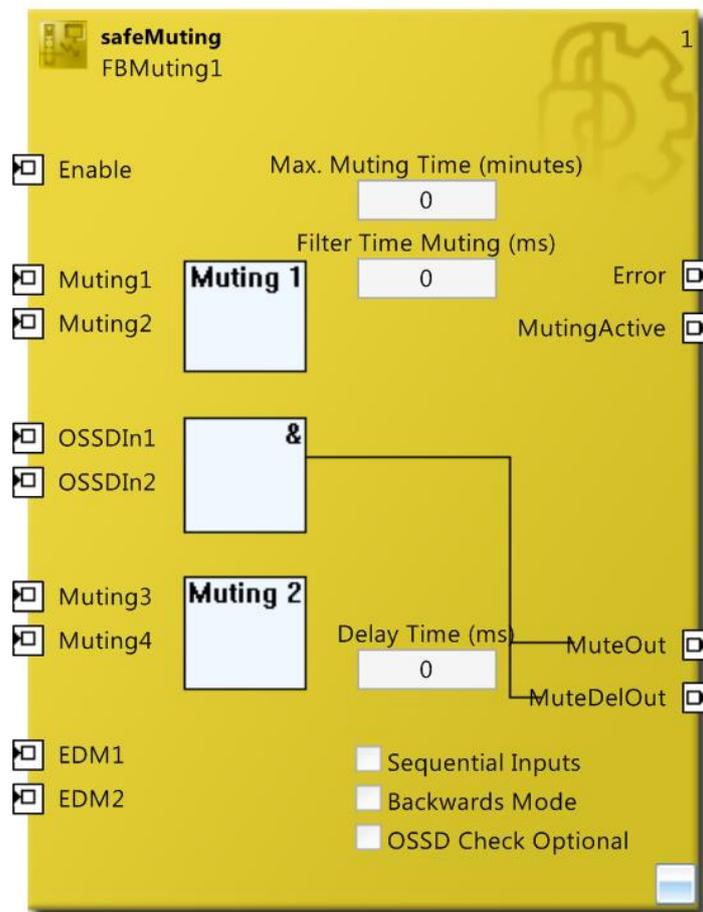


Figura 38: FB safeMuting

## 5.9. EDM

<b>FB EDM</b>	
Principio di base	Il blocco funzionale safeEdm (External Device Monitor) permette di monitorare i segnali Mon1 e Mon2. Accensione e spegnimento possono essere configurati.
Funzionamento	<p>Tale blocco serve per monitorare il cambio di stato logico dei segnali in ingresso. Quando il segnale Mon1 passa dallo stato logico FALSE a TRUE, il blocco verifica che il passaggio opposto si realizzi sull'altro ingresso entro un certo tempo, specificato nella finestra Switch On Monitoring (ms). Analogamente, quando il segnale Mon2 passa dallo stato logico FALSE a TRUE, il blocco verifica che il passaggio opposto si realizzi sull'altro ingresso entro un certo tempo, specificato nella finestra Switch Off Monitoring (ms).</p> <p>Se a fronte di una variazione di un ingresso anche l'altro non varia di conseguenza, il blocco va nello stato di errore, attivando la propria uscita.</p> <p>Per come funziona tale blocco, si capisce come questo vada nello stato di errore quando i due ingressi assumono lo stesso stato logico.</p>
Ingressi	<p>Mon1: porta di ingresso del primo segnale.            Mon2: porta di ingresso del secondo segnale.</p> <p>Entrambi gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> </ul>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione sopra descritta.</p> <p>L'uscita del blocco funzionale può essere collegata a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.

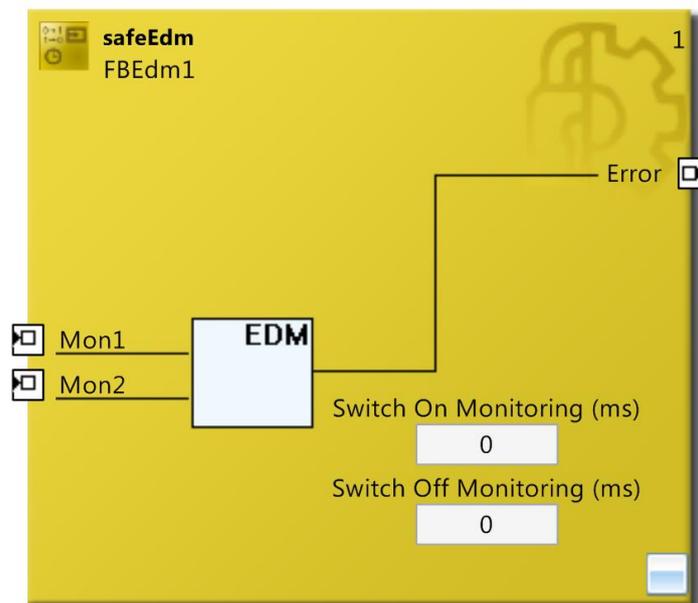


Figura 39: FB safeEdm

## 5.10. RS

<b>FB RS</b>	
Principio di base	Il blocco funzionale safeRs realizza la funzionalità reset/set.
Funzionamento	Tale blocco riceve in ingresso due segnali: Reset serve per impostare l'uscita nello stato logico FALSE; Set invece imposta l'uscita nello stato logico TRUE. Se i due segnali di ingresso sono attivi contemporaneamente, l'uscita RsOut viene impostata nello stato logico FALSE.
Ingressi	<p>Reset: porta di ingresso del segnale che imposta lo stato logico FALSE.</p> <p>Set: porta di ingresso del segnale che imposta lo stato logico TRUE.</p> <p>Entrambi gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul>
Uscite	<p>RsOut: porta di uscita del segnale.</p> <p>L'uscita del blocco funzionale può essere collegata a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>

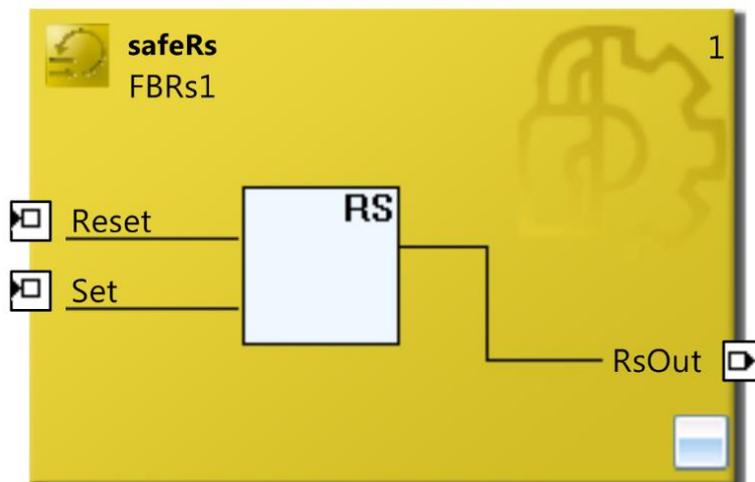


Figura 40: FB safeRs

## 5.11. SR

<b>FB SR</b>	
Principio di base	Il blocco funzionale safeSr realizza la funzionalità set/reset.
Funzionamento	Tale blocco riceve in ingresso due segnali: Set serve per impostare l'uscita nello stato logico TRUE; Reset invece imposta l'uscita nello stato logico FALSE. Se i due segnali di ingresso sono attivi contemporaneamente, l'uscita SrOut viene impostata nello stato logico TRUE.
Ingressi	<p>Set: porta di ingresso del segnale che imposta lo stato logico TRUE.</p> <p>Reset: porta di ingresso del segnale che imposta lo stato logico FALSE.</p> <p>Entrambi gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul>
Uscite	<p>SrOut: porta di uscita del segnale.</p> <p>L'uscita del blocco funzionale può essere collegata a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>

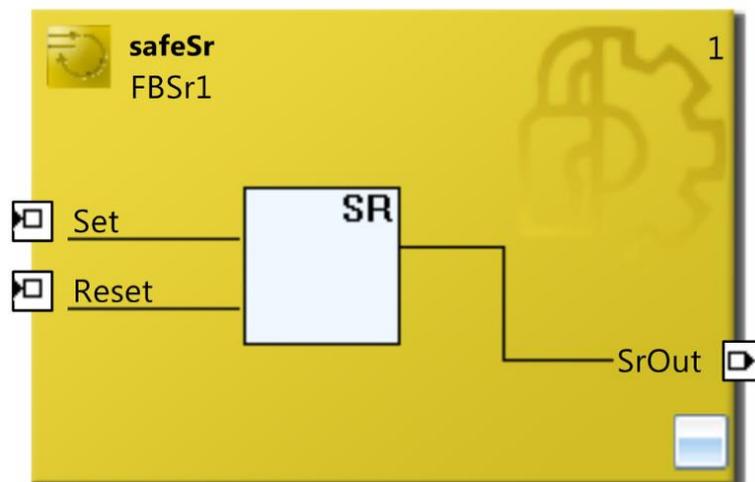


Figura 41: FB safeSr

## 5.12. TON

FB TON	
Principio di base	Il blocco funzionale safeTon permette di ritardare l'attivazione di un segnale.
Funzionamento	Tale blocco riceve in ingresso un segnale: quando questo passa allo stato logico TRUE, il blocco attende un tempo specificato nella finestra Delay Time (s) prima di attivare l'uscita.
Ingressi	TonIn: porta di ingresso del segnale che deve essere ritardato. L'ingresso del blocco funzionale può essere collegato a: <ul style="list-style-type: none"><li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li><li>• un'uscita di un altro blocco funzionale</li></ul>
Uscite	TonOut: porta di uscita del segnale. L'uscita del blocco funzionale può essere collegata a: <ul style="list-style-type: none"><li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li><li>• un ingresso di un altro blocco funzionale</li><li>• una variabile standard del PLC (l'input nel PLC %I*)</li></ul>

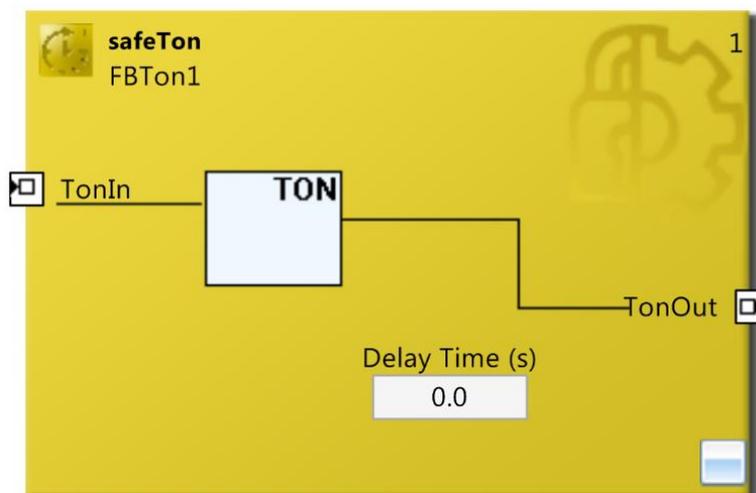


Figura 42: FB safeTon

## 5.13. TOF

<b>FB TOF</b>	
Principio di base	Il blocco funzionale safeTof permette di ritardare la disattivazione di un segnale.
Funzionamento	Tale blocco riceve in ingresso un segnale: quando questo passa allo stato logico FALSE, il blocco attende un tempo specificato nella finestra Delay Time (s) prima di disattivare l'uscita.
Ingressi	TofIn: porta di ingresso del segnale che deve essere ritardato. L'ingresso del blocco funzionale può essere collegato a: <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul>
Uscite	TofOut: porta di uscita del segnale. L'uscita del blocco funzionale può essere collegata a: <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>

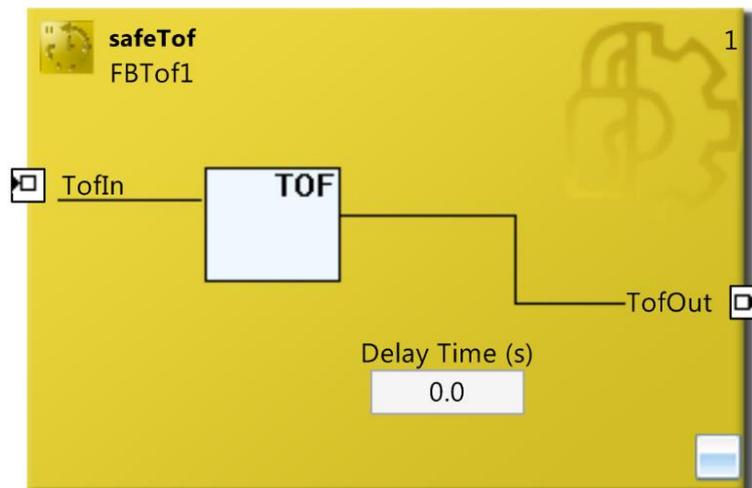


Figura 43: FB safeTof

## 5.14. CONNECTION SHUTDOWN

<b>FB CONNECTION SHUTDOWN</b>	
Principio di base	Il blocco funzionale safeConnShutdown permette di disattivare una connessione TwinSAFE.
Funzionamento	<p>Se l'ingresso del blocco si attiva, la connessione termina, un comando di shutdown viene inviato al partner FSoE e viene inviata una conferma all'uscita. Se il partner di comunicazione riceve il comando di shutdown, la comunicazione viene interrotta e l'output viene attivato.</p> <p>L'uscita torna allo stato logico FALSE quando la connessione con il partner FSoE viene ristabilita, ovvero quando quest'ultimo risponde ad una nuova connessione dopo che l'ingresso è stato disattivato.</p> <p>L'indirizzo identificativo del collegamento che deve essere disattivato va riportato nella finestra Connection ID.</p>
Ingressi	<p>Deactivate(x): porte di ingresso dei segnali.</p> <p>L'ingresso del blocco funzionale può essere collegato a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> </ul>
Uscite	<p>Error: uscita che si attiva se il blocco va nello stato di errore.</p> <p>Deactivated: porta di uscita del segnale.</p> <p>Entrambe le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>

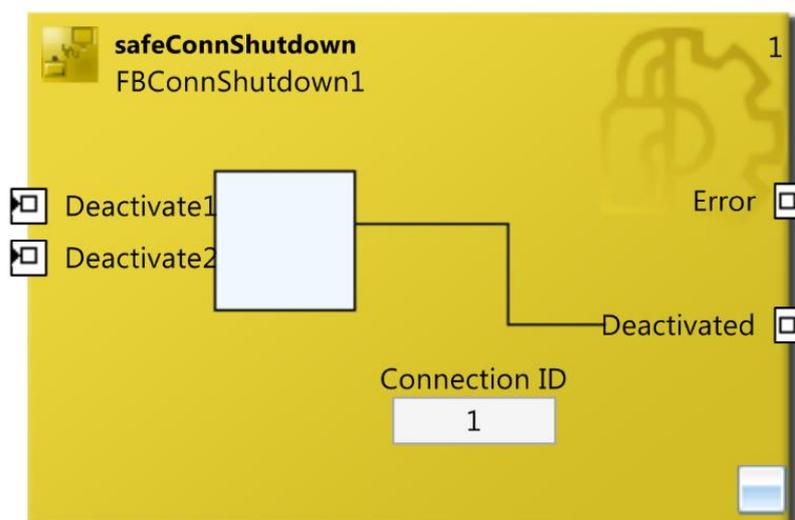


Figura 44: FB safeTof

## 5.15. ADD

<b>FB ADD</b>	
Principio di base	Il blocco funzionale safeAdd permette di sommare due segnali analogici in ingresso e trasferire il risultato dell'operazione all'uscita del blocco.
Funzionamento	Tale blocco riceve in ingresso due segnali analogici di cui ne somma il valore in ogni istante di tempo e ne riporta il risultato all'uscita del blocco funzionale stesso. Qualora si verificasse una situazione di overflow o di underflow il blocco si porta nello stato di errore.
Ingressi	AnalogIn1: porta di ingresso del primo segnale analogico. AnalogIn2: porta di ingresso del secondo segnale analogico. Entrambi gli ingressi del blocco funzionale possono essere collegati a: <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	Error: uscita che si attiva quando si verifica la condizione di overflow o di underflow. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate. AnalogOut: porta di uscita del segnale analogico elaborato. Entrambe le uscite del blocco funzionale possono essere collegate a: <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.

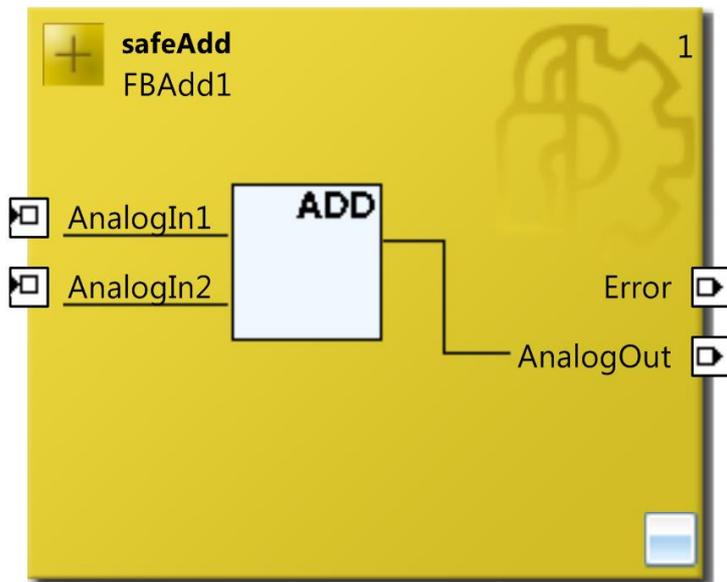


Figura 45: FB safeAdd

## 5.16. SUB

<b>FB SUB</b>	
Principio di base	Il blocco funzionale safeSub sottrae al primo segnale analogico in ingresso il secondo segnale analogico in ingresso e trasferisce il risultato dell'operazione all'uscita del blocco.
Funzionamento	Tale blocco riceve in ingresso due segnali analogici di cui sottrae al valore del primo il valore del secondo in ogni istante di tempo e ne riporta il risultato all'uscita del blocco funzionale stesso. Qualora si verificasse una situazione di overflow o di underflow il blocco si porta nello stato di errore.
Ingressi	AnalogIn1: porta di ingresso del primo segnale analogico. AnalogIn2: porta di ingresso del secondo segnale analogico. Entrambi gli ingressi del blocco funzionale possono essere collegati a: <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	Error: uscita che si attiva quando si verifica la condizione di overflow o di underflow. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate. AnalogOut: porta di uscita del segnale analogico elaborato. Entrambe le uscite del blocco funzionale possono essere collegate a: <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.

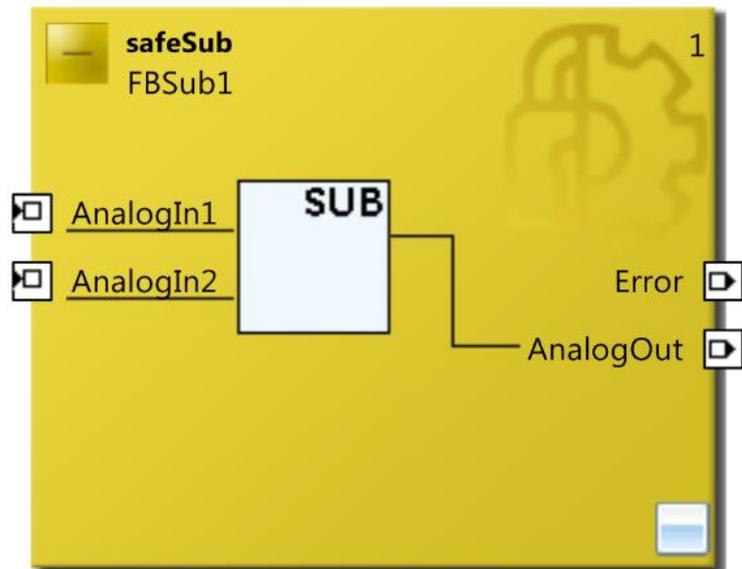


Figura 46: FB safeSub

## 5.17. MUL

<b>FB MUL</b>	
Principio di base	Il blocco funzionale safeMul permette di moltiplicare il valore di due segnali analogici in ingresso e trasferire il risultato dell'operazione all'uscita del blocco.
Funzionamento	Tale blocco riceve in ingresso due segnali analogici di cui ne realizza il prodotto in ogni istante di tempo e ne riporta il risultato all'uscita del blocco funzionale stesso. Qualora si verificasse una situazione di overflow o di underflow il blocco si porta nello stato di errore.
Ingressi	AnalogIn1: porta di ingresso del primo segnale analogico. AnalogIn2: porta di ingresso del secondo segnale analogico. Entrambi gli ingressi del blocco funzionale possono essere collegati a: <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	Error: uscita che si attiva quando si verifica la condizione di overflow o di underflow. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate. AnalogOut: porta di uscita del segnale analogico elaborato. Entrambe le uscite del blocco funzionale possono essere collegate a: <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.

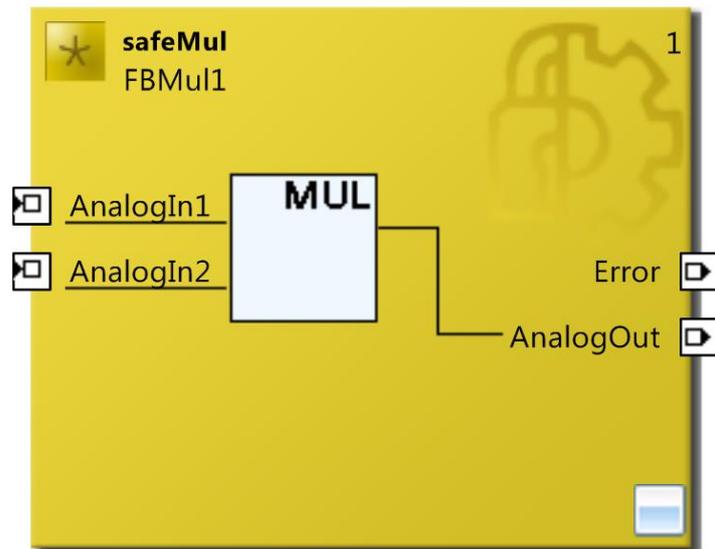


Figura 47: FB safeMul

## 5.18. DIV

<b>FB DIV</b>	
Principio di base	Il blocco funzionale safeDiv effettua una divisione tra i valori del primo segnale analogico in ingresso e quelli del secondo segnale analogico in ingresso e trasferisce il risultato dell'operazione all'uscita del blocco.
Funzionamento	<p>Tale blocco riceve in ingresso due segnali analogici di cui ne realizza la divisione in ogni istante di tempo e ne riporta il risultato all'uscita del blocco funzionale stesso.</p> <p>Il blocco permette di gestire il risultato dell'operazione, qualora questo non risultasse un numero intero. Per definire il tipo di approssimazione desiderata, si deve spuntare nella Division Rounding una tra le seguenti opzioni:</p> <ul style="list-style-type: none"> <li>• Floor tronca i valori decimali (approssimazione sempre per difetto)</li> <li>• Ceil riporta il valore all'intero successivo (approssimazione sempre per eccesso)</li> <li>• Round esegue un'approssimazione commerciale (per esempio il valore 2.5 è arrotondato a 3)</li> </ul> <p>Qualora si verificasse una situazione di overflow o di underflow il blocco si porta nello stato di errore.</p>
Ingressi	<p>AnalogIn1: porta di ingresso del primo segnale analogico.            AnalogIn2: porta di ingresso del secondo segnale analogico.            Entrambi gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione di overflow o di underflow. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate.</p> <p>AnalogOut: porta di uscita del segnale analogico elaborato.            Entrambe le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.

Se il valore del segnale di ingresso AnalogIn2 è 0, il valore del segnale di uscita AnalogOut viene impostato a 0 e non si hanno errori.

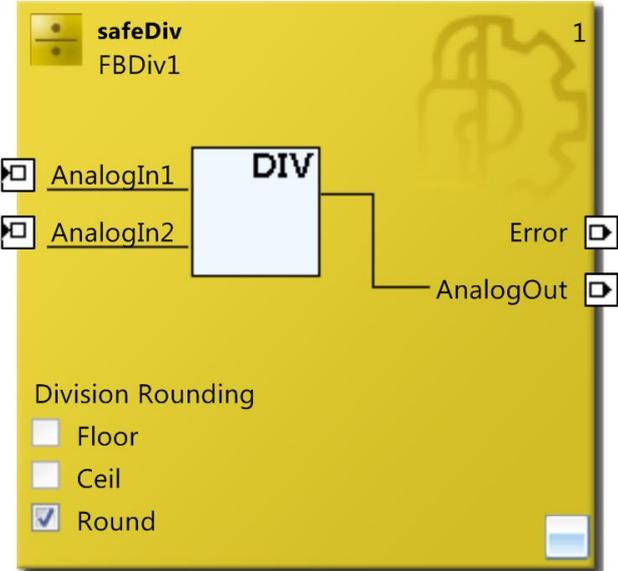


Figura 48: FB safeDiv

## 5.19. COMPARE

<b>FB COMPARE</b>	
Principio di base	Il blocco funzionale safeCompare verifica che un numero (variabile da 2 a 5) di segnali di ingresso analogici siano uguali negli stessi istanti di tempo. Sono permesse tolleranze sia sul valore del segnale che sugli istanti di tempo in cui questo viene confrontato.
Funzionamento	<p>Tale blocco riceve in ingresso da un minimo di due ad un massimo di cinque segnali analogici. Quelli che arrivano dal canale 2 al canale 4 vengono singolarmente confrontati con il segnale di riferimento del canale 1. Quando un numero minimo di canali coincide entro una certa tolleranza con il segnale di riferimento si attiva l'uscita IsValid del blocco. La sezione Architecture del blocco funzionale permette di selezionare quanti ingressi devono coincidere affinché l'uscita venga attivata:</p> <ul style="list-style-type: none"> <li>• 1oo2 (1 out of 2): almeno un ingresso deve coincidere con il riferimento</li> <li>• 2oo3 (2 out of 3): almeno due ingressi devono coincidere con il riferimento</li> <li>• 3oo5 (3 out of 5): almeno tre ingressi devono coincidere con il riferimento</li> </ul> <p>Nella sezione Allowed deviation si indica l'entità di massimo scostamento dal valore di riferimento.</p> <p>Nella sezione Tolerance time (ms) si indica il massimo intervallo di tempo entro il quale il segnale può essere shiftato. È presente un'ulteriore uscita analogica CompOut che riporta inalterato il segnale di riferimento.</p> <p>Qualora si verificasse una situazione di overflow o di underflow il blocco si porta nello stato di errore.</p>
Ingressi	<p>CompInx: <math>x = 1</math>, il segnale analogico è quello di riferimento;  <math>2 \leq x \leq 5</math>, il segnale deve essere confrontato con quello di riferimento.</p> <p>Tutti gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione di overflow o di underflow. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate.</p> <p>IsValid: indica lo stato logico della comparazione (esito positivo=1, esito negativo=0).</p>

	<p>CompOut: riporta in uscita il segnale di riferimento.</p> <p>Tutte le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	<p>Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p>

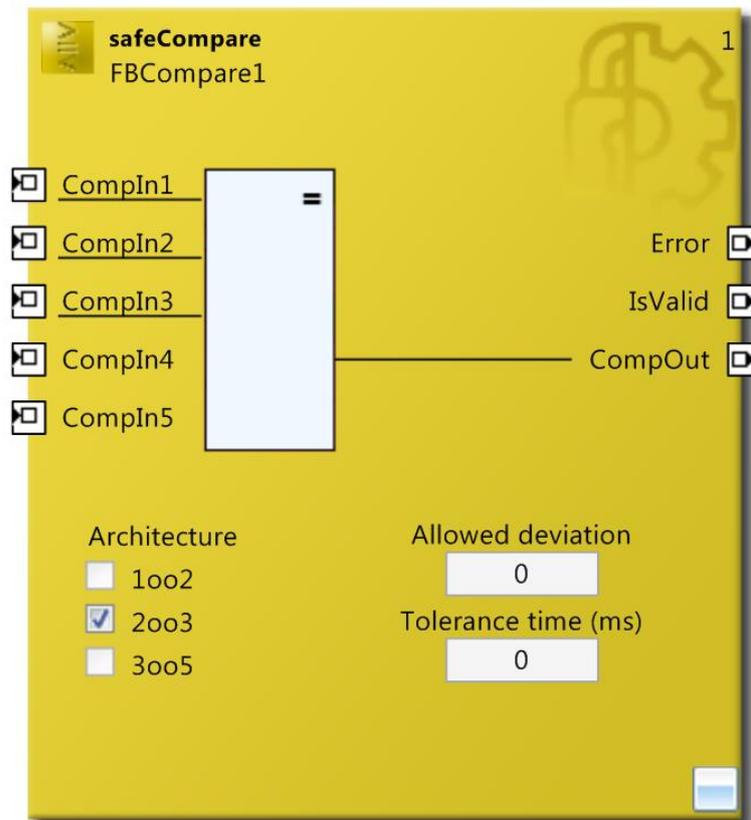


Figura 49: FB safeCompare

## 5.20. LIMIT

<b>FB LIMIT</b>	
Principio di base	Il blocco funzionale safeLimit permette di controllare il valore di un ingresso analogico e verificare che questo si trovi entro un definito range di valori.
Funzionamento	Tale blocco riceve in ingresso il segnale analogico che deve essere controllato: si verifica che questo non ecceda il valore massimo ne scenda sotto il valore minimo, questi ultimi parametri indicati dal programmatore rispettivamente nelle finestre Maximum Value e Minimum Value. Qualora questi limiti non fossero costanti nel tempo ma rappresentati ciascuno da un segnale elettrico, si possono riportare in ingresso al blocco nelle relative porte MaxValue e MinValue. Fintanto che il valore del segnale analogico di riferimento rimane entro i limiti specificati, il blocco attiva l'uscita InLimit; non appena questo supera il massimo si attiva l'uscita AboveMax mentre se scende sotto il valore minimo si attiva l'uscita BelowMin.
Ingressi	<p>AnalogIn: porta di ingresso del segnale analogico di riferimento che deve essere controllato.</p> <p>MinValue: porta di ingresso del segnale analogico che rappresenta il limite minimo.</p> <p>MaxValue: porta di ingresso del segnale analogico che rappresenta il limite massimo.</p> <p>Tutti gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione di errore nel gruppo a cui appartiene. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate.</p> <p>InLimit: uscita attiva quando il valore del segnale di ingresso è compreso entro i limiti.</p> <p>BelowMin: uscita attiva quando il valore del segnale di ingresso scende al di sotto del minimo.</p> <p>AboveMax: uscita attiva quando il valore del segnale di ingresso sale al di sopra del massimo.</p> <p>Tutte le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> </ul>

	<ul style="list-style-type: none"> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.

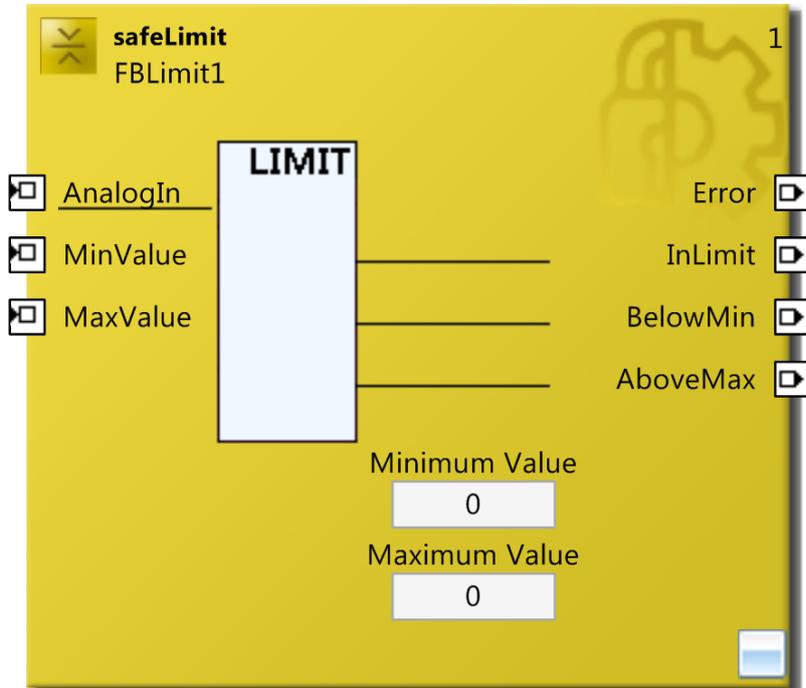


Figura 50: FB safeLimit

## 5.21. COUNTER

<b>FB COUNTER</b>	
Principio di base	Il blocco funzionale safeCounter permette di realizzare e gestire un contatore up/down.
Funzionamento	Tale blocco riceve in ingresso tre segnali: grazie ad essi si può incrementare o decrementare il valore del contatore oppure resettarlo. Inoltre, bisogna specificare nelle apposite finestre il valore di reset (ovvero il valore a cui si imposta il contatore quando arriva il segnale di reset) e il limite massimo che il contatore può raggiungere. Non appena lo si raggiunge, si attiva l'uscita CounterOut. L'uscita CounterZero si attiva, invece, ogni volta che il contatore assume valore 0. Infine, l'uscita ActValue riporta il valore attuale del contatore. Nel caso in cui questo si trovi a 0 e arriva il segnale di decremento, il blocco si porta nello stato di errore.
Ingressi	<p>Reset: imposta il contatore al valore specificato nel Preset Value</p> <p>CountUp: sul fronte di salita del segnale si incrementa il valore del contatore</p> <p>CountDown: sul fronte di salita del segnale si decrementa il valore del contatore</p> <p>Tutti gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione sopra descritta. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate.</p> <p>CounterOut: uscita che si attiva quando il valore del contatore supera quello specificato nel Counter Limit. Tale uscita è attiva anche quando i due valori coincidono.</p> <p>CountZero: uscita che si attiva quando il valore del contatore è zero.</p> <p>ActValue: uscita che riporta il valore attuale del contatore.</p> <p>Tutte le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>

Note	Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.
------	---

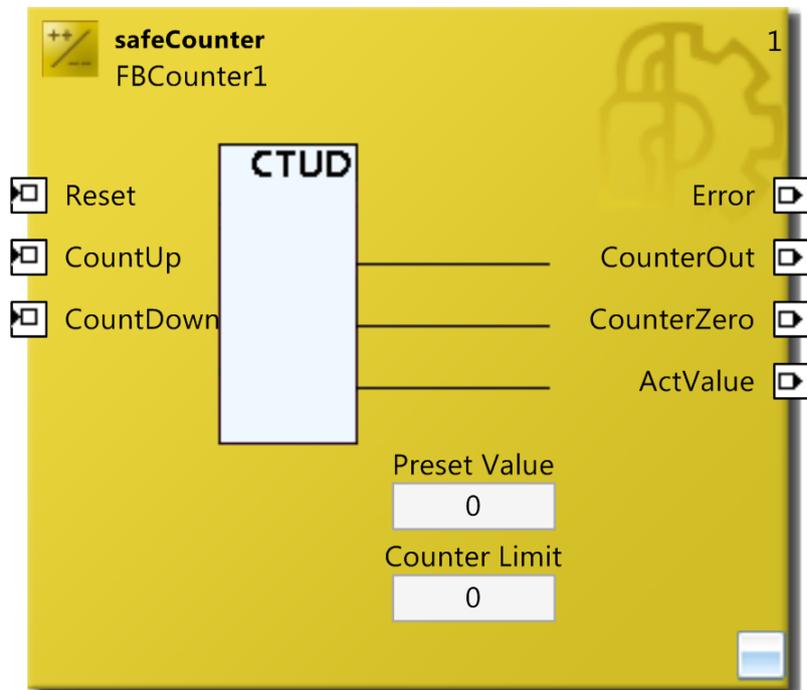


Figura 51: FB safeCounter

## 5.22. SCALE

<b>FB SCALE</b>	
Principio di base	Il blocco funzionale safeScaling permette di moltiplicare e dividere il valore di un segnale analogico in ingresso per poi aggiungere ad esso un valore di offset, prima di trasferirlo all'uscita del blocco stesso.
Funzionamento	<p>Tale blocco riceve in ingresso un segnale analogico che viene moltiplicato per lo Scaling Factor: questo parametro è rappresentato da una frazione nella quale si riporta a numeratore il valore per il quale si moltiplica l'ingresso e a denominatore il valore per il quale lo si divide. Nel caso in cui si voglia effettuare solo l'operazione di moltiplicazione o quella di divisione, è necessario imporre l'altro valore ad 1.</p> <p>Il blocco permette di gestire il risultato dell'operazione di divisione, qualora questo non risultasse un numero intero. Per definire il tipo di approssimazione desiderata, si deve spuntare nella Division Rounding una tra le seguenti opzioni:</p> <ul style="list-style-type: none"> <li>• Floor tronca i valori decimali (approssimazione sempre per difetto)</li> <li>• Ceil riporta il valore all'intero successivo (approssimazione sempre per eccesso)</li> <li>• Round esegue un'approssimazione commerciale (per esempio il valore 2.5 è arrotondato a 3)</li> </ul> <p>La finestra Scaling Offset somma algebricamente un numero (sia positivo che negativo) al risultato dell'operazione appena descritta.</p> <p>Il parametro Watchdog (ms) può essere utilizzato per specificare l'intervallo di tempo entro il quale il segnale analogico d'ingresso deve variare. Se il valore di input rimane invariato per un tempo superiore a quello specificato l'output StuckAtError viene impostato a TRUE. Se si vuole escludere questo controllo, si imposta il valore di tale parametro a 0.</p> <p>Qualora si verificasse una situazione di overflow o di underflow il blocco si porta nello stato di errore.</p>
Ingressi	<p>AnalogIn: porta di ingresso del segnale analogico. Questo può provenire da:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione di overflow o di underflow. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate.</p>

	<p>StuckAtError: uscita che si attiva quando viene violato il tempo di watchdog. Il blocco funzionale non entra nello stato di errore.</p> <p>AnalogOut: canele di uscita del segnale elaborato.</p> <p>Tutte le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	<p>Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p>

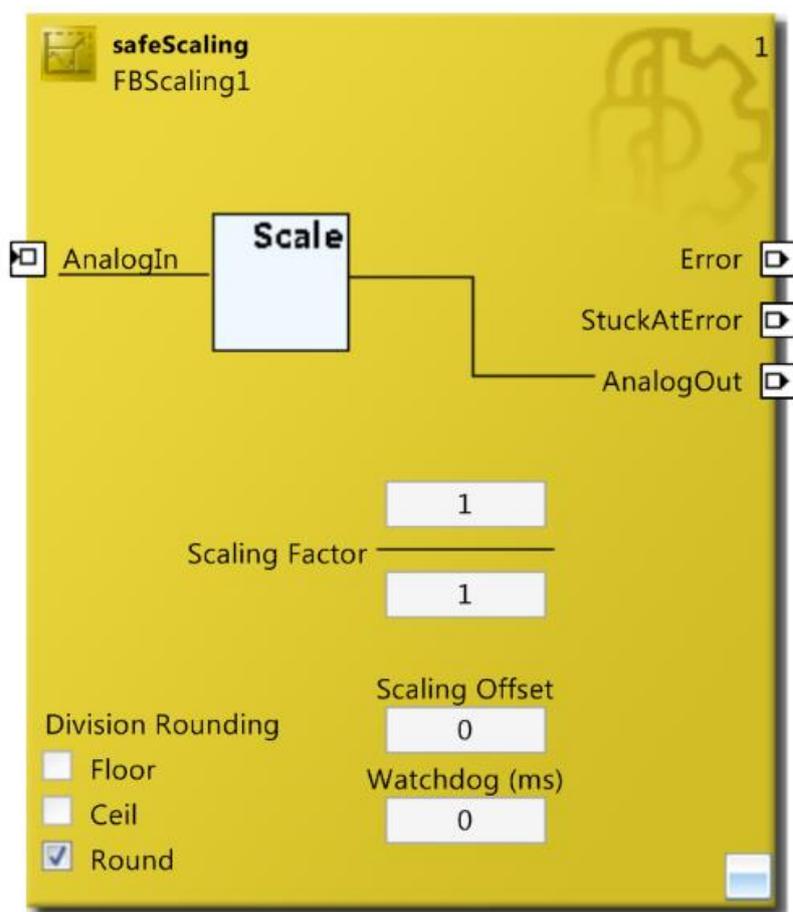


Figura 52: FB safeScaling

## 5.23. SPEED

<b>FB SPEED</b>	
Principio di base	Il blocco funzionale safeSpeed permette di calcolare la velocità di un elemento in movimento.
Funzionamento	Tale blocco riceve in ingresso un segnale analogico che corrisponde alla posizione reale dell'oggetto (per esempio il segnale di un encoder analogico). Nella finestra Time Interval (ms) si inserisce il tempo di campionamento del segnale. Il blocco funzionale genera in uscita il segnale di velocità istantanea dell'organo in movimento. Qualora si verificasse una situazione di overflow o di underflow il blocco si porta nello stato di errore.
Ingressi	Position: segnale analogico di posizione per il calcolo della velocità. L'ingresso del blocco funzionale può essere collegato a: <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	Error: uscita che si attiva quando si verifica la condizione di overflow o di underflow. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate. Speed: uscita che riporta la velocità istantanea calcolata. Entrambe le uscite del blocco funzionale possono essere collegate a: <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.

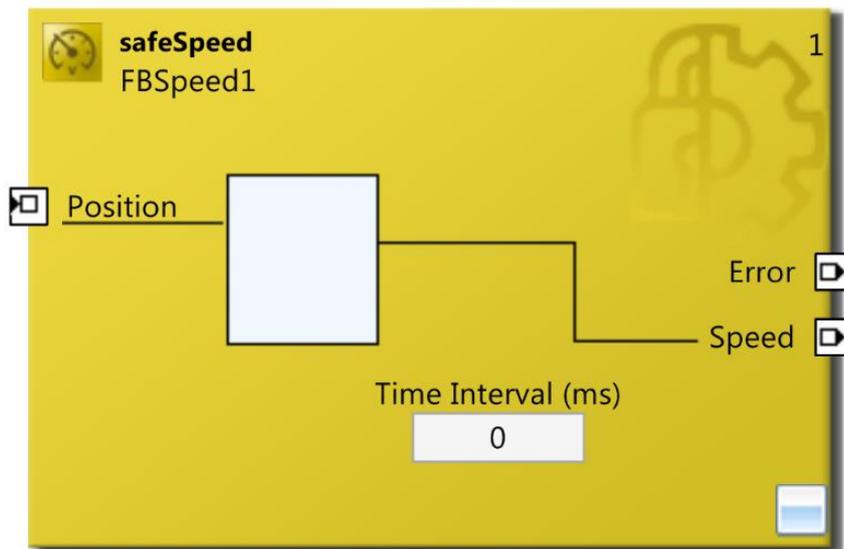


Figura 53: FB safeSpeed

## 5.24. LOADSENSING

<b>FB LOADSENSING</b>	
Principio di base	<p>Il blocco funzionale safeLoadSensing permette di verificare che l'andamento di una grandezza fisica rientri entro certi limiti. Ovvero, data una curva, permette di suddividere il suo dominio (campo di esistenza) in un massimo di 25 intervalli e per ognuno di questi di specificarne un range di accettabilità e uno di warning.</p>
Funzionamento	<p>Per configurare questo blocco è necessario anzitutto compilare una tabella nella quale si definisce il campo di variabilità della curva. Come si vede in Figura 55, bisogna specificare per ogni intervallo la coordinata X che definisce l'inizio dell'intervallo attuale ed il termine dell'intervallo precedente. Per ognuno di questi, si specificano poi altri quattro parametri:</p> <ul style="list-style-type: none"> <li>• WY1 e WY2 sono gli estremi del corretto campo di variabilità in cui il punto della curva può ricadere</li> <li>• Y1 e Y2 sono i valori estremi ammissibili, oltre i quali il punto non può più essere accettato</li> </ul> <p>In formule:</p> $Y1 \leq WY1 < WY2 \leq Y2$ <p>Dai canali di ingresso AnalogInX e AnalogInY arrivano i segnali contenenti rispettivamente le coordinate X e Y dei punti da verificare.</p> <p>Se il punto in esame si trova oltre i valori limite Y1 e Y2 dell'intervallo di appartenenza, entrambe le uscite del blocco funzionale rimangono nello stato logico FALSE. Se il punto invece assume un valore compreso tra Y1 e WY1 oppure tra WY2 e Y2 entrambe le uscite del blocco funzionale si portano nello stato logico TRUE. Infine, se il punto considerato rientra tra i valori WY1 e WY2, l'uscita Valid passa nello stato logico TRUE mentre l'uscita Warning rimane nello stato logico FALSE.</p> <p>Se si attiva l'ingresso Inactive il controllo effettuato dal blocco viene disattivato e le uscite vengono impostate allo stato logico FALSE.</p>
Ingressi	<p>Inactive: porta di ingresso del segnale per la disattivazione del blocco.</p> <p>AnalogInX: ingresso analogico delle coordinate x.</p> <p>AnalogInY: ingresso analogico delle coordinate y.</p> <p>Gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> </ul>

	<ul style="list-style-type: none"> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	<p>Valid: uscita che viene attivata se il punto è compreso tra i valori Y1 e Y2 del relativo intervallo.</p> <p>Warning: uscita che viene attivata se il punto è compreso tra i valori Y1 e WY1 oppure tra WY2 e Y2.</p> <p>Entrambe le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	<p>Se si spunta la finestra Outside [y1;y2] il controllo viene ribaltato, ovvero il blocco verifica che i punti in ingresso nei canali AnalogInX e AnalogInY cadano all'esterno dell'area indicata nella tabella, la quale deve essere ricompilata perché i valori di warning assumono significati differenti. In questo caso, la tabella si compila secondo la seguente logica:</p> $WY1 \leq Y1 < Y2 \leq WY2$



Figura 54: FB safeLoadSensing

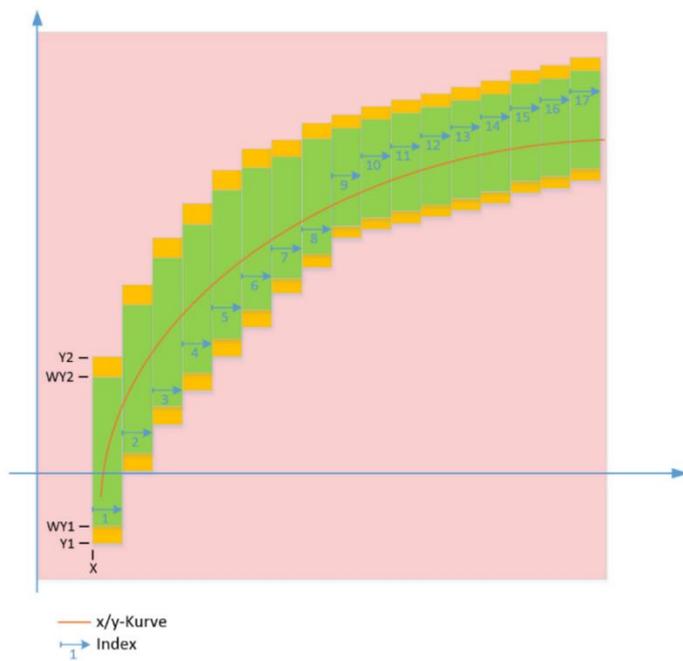


Figura 55: andamento di un generico campo di accettabilità di una curva

Index	X	Y1	Y2	WY1	WY2
1	-30	-200	-100	-180	-80
2	-25	-100	-50	-90	-60
3	-20	-50	0	-40	-10
4	-15	0	100	10	90
5	-10	100	200	110	190
6	-5	200	300	210	290
7	0	300	400	310	390
8	5	400	500	410	490
9	10	500	600	510	590
10	15	600	700	610	690
11	20	700	800	710	790
12	25	800	900	810	890
13	30	900	1000	910	990
14	35	1000	1100	1010	1090
15	40	1100	1200	1110	1190
16	45	1200	1300	1210	1290
17	50	1300	1400	1310	1390
18	60	1400	1500	1410	1490
19	80	1500	1600	1510	1590
20	100	1600	1700	1610	1690
21	140	1700	1800	1710	1790
22	180	1800	2000	1810	1990
23	250	2000	3000	2010	2990
24	500	3000	4000	3010	3990
25	1000	4000	5000	4010	4990

Tabella 4: esempio di tabella del saveLoadSensing

## 5.25. CAMMONITOR

Il blocco funzionale safeCamMonitor viene utilizzato per realizzare un controllo elettronico delle camme. In particolare, questo blocco funzionale offre la possibilità di gestire il movimento secondo il moto oscillatorio di un pendolo (Pendulum Mode) oppure secondo un moto rotativo più classico (Excentric Mode), realizzato in senso orario oppure in senso antiorario. La scelta della configurazione da attuare si realizza spuntando la relativa casella predisposta all'interno del blocco stesso. Non è possibile realizzare il passaggio da una configurazione ad all'altra in maniera dinamica in quanto i segnali e i dati d'ingresso assumono significati differenti nelle due configurazioni; per questo motivo, la programmazione di questo blocco viene presentata come se si stessero studiando blocchi differenti, andando ad approfondire il significato di ogni ingresso e di ogni uscita per ciascuna configurazione.

### 5.25.1. CAMMONITOR (Pendulum Mode)

<b>FB CAMMONITOR</b>	
Principio di base	Il blocco funzionale safeCamMonitor viene utilizzato per realizzare un controllo elettronico delle camme. Sono supportate due modalità di funzionamento: pendolo (Pendulum Mode) e eccentrica (Excentric Mode).
Funzionamento	<p>Nella modalità Pendulum Mode sono permesse entrambe le direzioni di rotazione. In questa configurazione sono parametrizzati due punti superiori di inversione. Poiché la curva possa essere regolata per ciascun prodotto, si imposta come limite dei punti di inversione il massimo range della corsa oscillante. Il limite di inversione inferiore BDC viene impostato invece con un limite superiore e uno inferiore.</p> <p>Nella modalità Pendulum Mode il sistema verifica che i limiti superiori TDC1_UpperLimit e TDC2_UpperLimit non vengano mai superati. Se ciò accade, l'uscita CamMonOK passa allo stato logico FALSE.</p> <p>Il ciclo si avvia quando l'ingresso Reset registra un fronte di discesa: quando ciò avviene, si può iniziare qualsiasi tipo di movimento (pulsante, inverso, ...) fino al raggiungimento del punto morto inferiore BDC. Oltre questo punto il solo movimento consentito è verso l'alto e quando questo avviene l'uscita UpwardsMove passa nello stato logico TRUE. Il successivo avvio del ciclo viene abilitato quando l'ingresso Reset registra un nuovo fronte di discesa: se il movimento sta procedendo verso il basso e non viene rilevato nessun fronte di discesa sull'ingresso Reset, il sistema si arresta e l'uscita CamMonOK passa nello stato logico FALSE.</p> <p>Per maggior chiarezza, fare riferimento alla Figura 57.</p>

Ingressi	<p>Automatic: se impostato allo stato logico FALSE si ha la normale modalità di funzionamento; se impostato a TRUE la modalità automatica è attiva.</p> <p>SettingMode: quando questo ingresso si trova allo stato logico TRUE, si possono modificare i parametri interni del blocco. In particolare, in questa configurazione TDC1_LowerLimit, TDC1_UpperLimit, TDC2_LowerLimit, TDC2_UpperLimit, BDC_LowerLimit e BDC_UpperLimit vengono modificati retroattivamente e salvati in modo non volatile.</p> <p>Position: segnale analogico di ingresso che riporta la posizione attuale dell'elemento in rotazione.</p> <p>Reset: prima dell'inizio di ogni ciclo deve essere rilevato un fronte di discesa su questo ingresso.</p> <p>Max Position: casella utilizzata per specificare il massimo valore permesso durante la rotazione di 360°.</p> <p>PressStarted: se questa porta di ingresso è abilitata, è previsto un movimento o un cambio di posizione qualora lo stato logico dell'ingresso fosse TRUE. Inoltre, quando questo ingresso è attivo bisogna che siano configurate anche le caselle PressStartDelayTime, MoveDetectionTime e MinPositionChange che si attivano nella finestra Properties.</p> <p>TDC1_LowerLimit: casella in cui bisogna inserire l'estremo inferiore del TDC del "lato sinistro". Deve assumere un valore superiore a quello associato al BDC e inferiore a quello associato al TDC1_UpperLimit.</p> <p>TDC1_UpperLimit: casella in cui bisogna inserire l'estremo superiore del TDC del "lato sinistro". Deve assumere un valore superiore a quello associato al TDC1_LowerLimit e inferiore a quello associato al Max Position.</p> <p>TDC2_LowerLimit: casella in cui bisogna inserire l'estremo inferiore del TDC del "lato destro". Deve assumere un valore superiore a quello associato al BDC e inferiore a quello associato al TDC2_UpperLimit.</p> <p>TDC2_UpperLimit: casella in cui bisogna inserire l'estremo superiore del TDC del "lato destro". Deve assumere un valore superiore a quello associato al TDC2_LowerLimit e inferiore a quello associato al Max Position.</p> <p>BDC_LowerLimit: casella in cui bisogna inserire l'estremo inferiore del punto morto inferiore. Il suo valore deve essere inferiore a Max Position/2 e superiore a OverrunMax.</p> <p>BDC_UpperLimit: casella in cui bisogna inserire l'estremo superiore del punto morto inferiore. Il suo valore deve essere superiore a Max Position/2 e inferiore a TDC1_LowerLimit.</p> <p>OverrunMax: in questa configurazione deve essere disabilitato o configurato nello stato logico FALSE.</p> <p>BackwardsMove: in questa configurazione deve essere disabilitato.</p>
----------	---

	<p>OverrunCam: in questa configurazione deve essere disabilitato.</p> <p>UpwardCam: in questa configurazione deve essere disabilitato.</p>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione di sopra descritta. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate.</p> <p>TDC: uscita che passa allo stato logico TRUE quando la posizione attuale è compresa tra gli estremi TDC1_LowerLimit e TDC1_UpperLimit oppure tra gli estremi TDC2_LowerLimit e TDC2_UpperLimit</p> <p>BDC: uscita che passa allo stato logico TRUE quando la posizione attuale è compresa tra gli estremi BDC_LowerLimit e BDC_UpperLimit.</p> <p>CamMonOK: uscita che passa allo stato logico TRUE quando tutte le verifiche interne sopra descritte non riscontrano errori. Quando il gruppo in cui è stato allocato il blocco in questione entra nello stato di RUN, questa uscita si attiva solamente in corrispondenza del primo fronte di discesa registrato nell'ingresso Reset.</p> <p>OverrunTDC: non utilizzato in questa configurazione.</p> <p>OverrunStop: uscita analogica che rappresenta la differenza tra il valore assunto da Position quando è stato rilevato il fronte di discesa sull'ingresso PressStarted e il valore della posizione attuale.</p>
Note	<p>Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p> <p>Per configurare correttamente questo blocco funzionale, è necessario configurare anche i parametri che sono presenti nella finestra Properties.</p> <p>Allowed Position Jitter: oscillazioni del valore di posizione ammissibile anche in situazione di arresto.</p> <p>Stop Detection Time: poiché la posizione solitamente viene ricevuta tramite una connessione TwinSAFE, il suo valore non cambierà ad ogni ciclo. Si deve specificare durante la fase di arresto il tempo entro il quale la posizione può oscillare in prossimità di Allowed Position Jitter.</p> <p>Move Detection Time: se è attivo l'ingresso PressStarted, è necessario specificare il tempo entro il quale la posizione deve cambiare dopo che viene rilevato un movimento per la prima volta.</p> <p>Min Position Change: se è attivo l'ingresso PressStarted, è necessario specificare il valore per indicare la minima variazione della posizione all'interno di Move Detection Time.</p> <p>Press Start Delay Time: se è attivo l'ingresso PressStarted, è necessario specificare il tempo entro il quale deve essere rilevato un movimento a seguito di un fronte di salita sull'ingresso PressStarted.</p>

Operation Mode: parametro che permette la scelta tra Pendulum Mode e Excentric Mode.

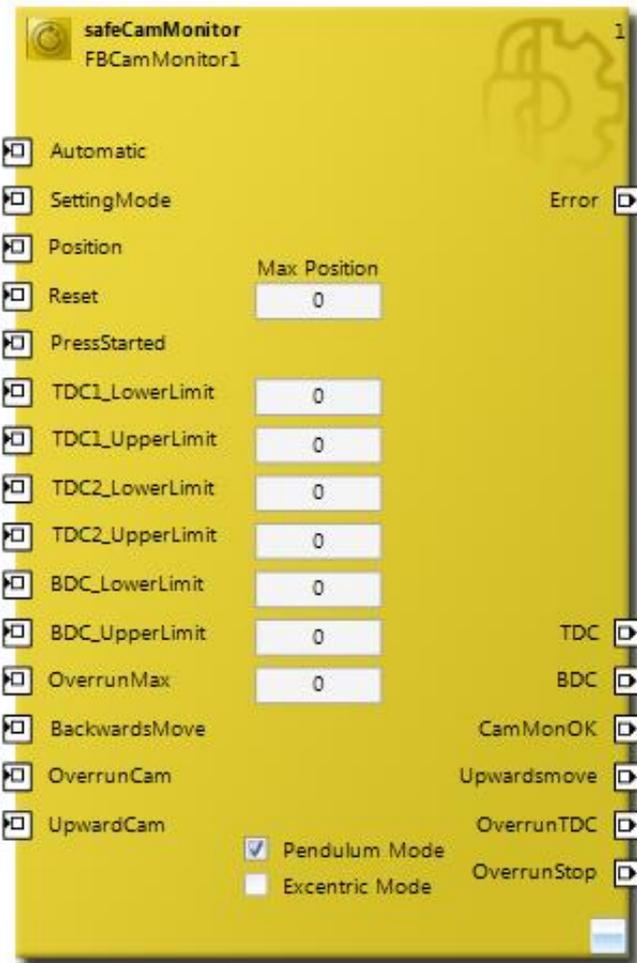


Figura 56: FB safeCamMonitor in Pendulum Mode

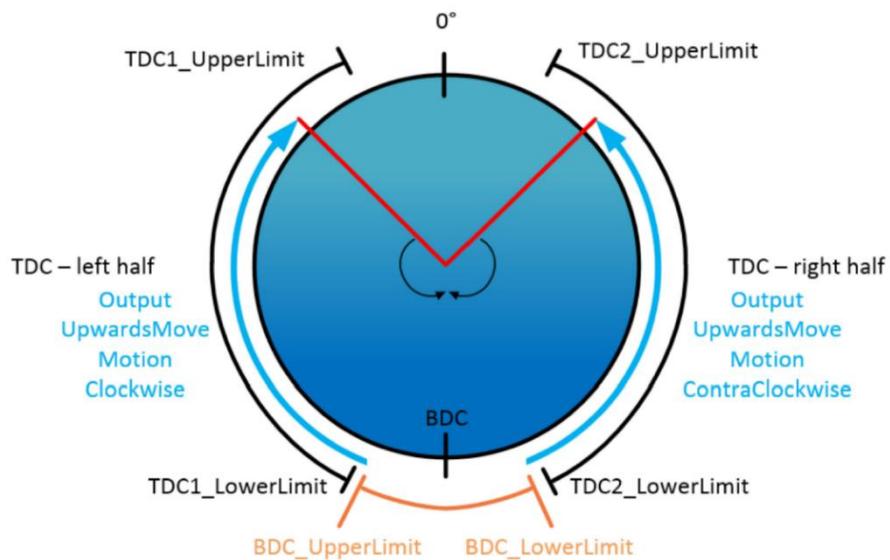


Figura 57: Pendulum Mode – diagramma schematico dei valori da configurare

## 5.25.2. CAMMONITOR (Excentric Mode)

<b>FB CAMMONITOR</b>	
Principio di base	Il blocco funzionale safeCamMonitor viene utilizzato per realizzare un controllo elettronico delle camme. Sono supportate due modalità di funzionamento: pendolo (Pendulum Mode) e eccentrica (Excentric Mode).
Funzionamento	<p>Nella modalità Excentric Mode è permessa una sola direzione di rotazione. Il blocco funzionale verifica che dopo un ciclo la posizione di arresto sia compresa tra il punto morto superiore TDC (Top Dead Center) e un overrun massimo, indicato nella casella OverrunMax. Il TDC è definito come un intervallo di valori, permettendo così di gestire una certa tolleranza: per questo motivo vengono definiti i punti TDC1_LowerLimit e TDC1_UpperLimit. Analogamente per il punto morto inferiore BDC (Bottom Dead Center) si definiscono gli estremi dell'intervallo che lo identificano: tali valori vengono riportati nelle caselle BDC_UpperLimit e BDC_LowerLimit.</p> <p>Raggiunto lo stato di arresto, il nuovo ciclo è consentito solo in seguito ad un fronte negativo sull'ingresso Reset.</p> <p>L'overrun corrente o la posizione angolare dell'organo in rotazione possono essere acquisiti dall'uscita OverrunTDC.</p> <p>L'uscita CamMonOK viene impostata allo stato logico FALSE se l'organo in rotazione si arresta senza aver raggiunto e superato il TDC oppure se il senso di rotazione è invertito.</p> <p>Superato il BDC si risale verso il TDC: in questa fase l'uscita Upwardsmove passa nello stato logico TRUE.</p> <p>L'ingresso BackwardsMove viene utilizzato quando si vuole invertire il senso della rotazione: questa operazione è consentita solo se ci si trova tra TDC1_UpperLimit e BCD_LowerLimit e termina quando si raggiunge il punto TDC1_UpperLimit.</p> <p>Per maggior chiarezza, fare riferimento alla Figura 59.</p>
Ingressi	<p>Automatic: se impostato allo stato logico FALSE si ha la normale modalità di funzionamento; se impostato a TRUE la modalità automatica è attiva.</p> <p>SettingMode: quando questo ingresso si trova allo stato logico TRUE, si possono modificare i parametri interni del blocco. In particolare, in questa configurazione TDC1_LowerLimit, TDC1_UpperLimit, BDC_LowerLimit, BDC_UpperLimit e OverrunMax vengono modificati retroattivamente e salvati in modo non volatile.</p> <p>Position: segnale analogico di ingresso che riporta la posizione attuale dell'elemento in rotazione.</p> <p>Reset: prima dell'inizio di ogni ciclo deve essere rilevato un fronte di discesa su questo ingresso.</p>

	<p>Max Position: casella utilizzata per specificare il massimo valore permesso durante la rotazione di 360°.</p> <p>PressStarted: se questa porta di ingresso è abilitata, è previsto un movimento o un cambio di posizione qualora lo stato logico dell'ingresso fosse TRUE. Inoltre, quando questo ingresso è attivo bisogna che siano configurate anche le caselle PressStartDelayTime, MoveDetectionTime e MinPositionChange che si attivano nella finestra Properties.</p> <p>TDC1_LowerLimit: casella in cui bisogna inserire l'estremo inferiore del punto morto superiore. Si deve trovare alla sinistra del TDC.</p> <p>TDC1_UpperLimit: casella in cui bisogna inserire l'estremo superiore del punto morto superiore. Si deve trovare alla destra del TDC.</p> <p>TDC2_LowerLimit: non utilizzato in questa configurazione. Valore impostato a 0.</p> <p>TDC2_UpperLimit: non utilizzato in questa configurazione. Valore impostato a 0.</p> <p>BDC_LowerLimit: casella in cui bisogna inserire l'estremo inferiore del punto morto inferiore. Il suo valore deve essere inferiore a Max Position/2 e superiore a OverrunMax.</p> <p>BDC_UpperLimit: casella in cui bisogna inserire l'estremo superiore del punto morto inferiore. Il suo valore deve essere superiore a Max Position/2 e inferiore a TDC1_LowerLimit.</p> <p>OverrunMax: posizione limite d'arresto; se l'arresto avviene oltre questo valore l'uscita CamMonOK passa nello stato logico FALSE. Questo valore deve essere compreso tra TDC1_UpperLimit e BDC_LowerLimit.</p> <p>BackwardsMove: ingresso che, se attivo, permette la rotazione in direzione inversa; tale rotazione è consentita solo entro gli estremi BDC_LowerLimit e TDC1_UpperLimit e termina quando quest'ultimo viene raggiunto.</p> <p>OverrunCam: non utilizzato in questa configurazione.</p> <p>UpwardCam: non utilizzato in questa configurazione.</p>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione di sopra descritta. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate.</p> <p>TDC: uscita che passa allo stato logico TRUE quando la posizione attuale è compresa tra gli estremi TDC1_LowerLimit e TDC1_UpperLimit.</p> <p>BDC: uscita che passa allo stato logico TRUE quando la posizione attuale è compresa tra gli estremi BDC_LowerLimit e BDC_UpperLimit.</p> <p>CamMonOK: uscita che passa allo stato logico TRUE quando tutte le verifiche interne sopra descritte non riscontrano errori. Quando il gruppo in cui è stato allocato il blocco in questione entra nello stato di RUN, questa uscita si attiva solamente in</p>

	<p>corrispondenza del primo fronte di discesa registrato nell'ingresso Reset.</p> <p>UpwardsMove: uscita che passa allo stato logico TRUE quando la posizione attuale è compresa tra gli estremi BDC_UpperLimit e la posizione iniziale a 0°.</p> <p>OverrunTDC: uscita analogica che rappresenta la differenza tra TDC1_LowerLimit e il valore della posizione attuale.</p> <p>OverrunStop: uscita analogica che rappresenta la differenza tra il valore assunto da Position quando è stato rilevato il fronte di discesa sull'ingresso PressStarted e il valore della posizione attuale.</p>
Note	<p>Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p> <p>Per configurare correttamente questo blocco funzionale, è necessario configurare anche i parametri che sono presenti nella finestra Properties.</p> <p>Allowed Position Jitter: massima oscillazione ammissibile del valore di posizione in situazione di arresto.</p> <p>Stop Detection Time: poiché la posizione solitamente viene ricevuta tramite una connessione TwinSAFE, il suo valore non cambierà ad ogni ciclo. Si specifica il tempo entro il quale il sistema si deve arrestare, ovvero può oscillare in prossimità del punto morto superiore di una quantità Allowed Position Jitter.</p> <p>Move Detection Time: se è attivo l'ingresso PressStarted, è necessario specificare il tempo entro il quale la posizione deve cambiare dopo che viene rilevato un movimento per la prima volta.</p> <p>Min Position Change: se è attivo l'ingresso PressStarted, è necessario specificare il valore per indicare la minima variazione della posizione all'interno di Move Detection Time.</p> <p>Press Start Delay Time: se è attivo l'ingresso PressStarted, è necessario specificare il tempo entro il quale deve essere rilevato un movimento a seguito di un fronte di salita sull'ingresso PressStarted.</p> <p>Operation Mode: parametro che permette la scelta tra Pendulum Mode e Excentric Mode.</p>

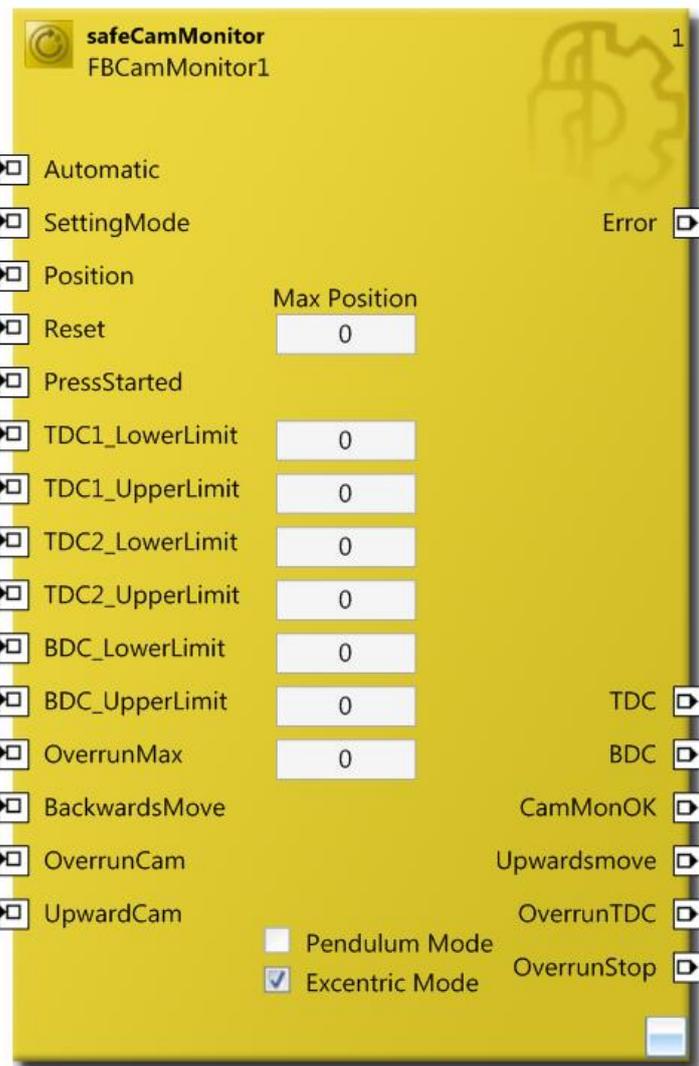


Figura 58: FB safeCamMonitor in Excentric Mode

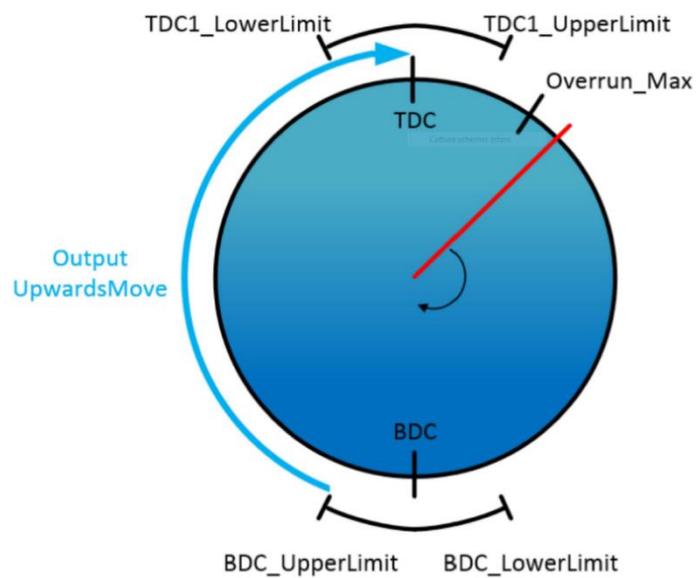


Figura 59: Excentric Mode – diagramma schematico dei valori da configurare

### 5.25.3. Descrizione del processo

Di seguito viene descritto nel particolare il funzionamento del blocco quando si abilita l'ingresso PressStarted, ponendo particolare attenzione a tutte quelle voci che bisogna compilare nella finestra Properties.

Il blocco funzionale safeCamMonitor rileva lo stato di fermo (Stopped = TRUE) se la variazione della posizione all'interno di StopDetectionTime è minore o uguale al valore riportato in AllowedPositionJitter. Ciò significa che il blocco è in grado di gestire delle piccole oscillazioni ovvero dei segnali di rumore quando l'elemento in rotazioni risulta fermo

Se l'ingresso PressStarted si trova nello stato logico TRUE, il blocco monitora il movimento. Il blocco funzionale rileva un movimento solo se la variazione di posizione (MoveDetectionTime) assume un valore superiore a quello specificato in MinPositionChange.

Quando l'ingresso PressStarted passa dallo stato logico FALSE a TRUE, si attiva il timer PressStartDelayTimer; se scaduto questo tempo non è stato rilevato alcun movimento, il blocco funzionale passa nello stato di errore, attivando l'uscita Error e disattivando tutte le altre.

Per poter rilevare un movimento, la posizione deve cambiare di un valore superiore a quello specificato nel MaxPositionJitter prima che il timer StopDetectionTime termini il conteggio.

Un movimento all'indietro o in senso antiorario (MoveContraClockwise = TRUE) viene rilevato se la posizione è cambiata in senso antiorario.

Un movimento in avanti o in senso orario (MoveClockwise = TRUE) viene rilevato se la posizione è cambiata in senso orario.

Se l'ingresso SettingMode è impostato, in modalità excentric i valori fissi TDC1\_LowerLimit, TDC1\_Upperlimit, BDC\_Lowerlimit, BDC\_Upperlimit e OverrunMax vengono modificati retroattivamente e salvati in modo non volatile. In modalità pendolo questo vale per i valori TDC1\_Lowerlimit, TDC1\_Upperlimit, TDC2\_Lowerlimit, TDC2\_Upperlimit, BDC\_Lowerlimit e BDC\_Upperlimit.

## 5.26. SLI

<b>FB SLI</b>	
Principio di base	Il blocco funzionale safeSLI viene utilizzato per effettuare un controllo di posizione.
Funzionamento	<p>Tale blocco riceve in ingresso due segnali: quando si rileva un fronte di salita sull'ingresso SLI, viene salvato il valore letto in quell'istante dall'input Position. Fintanto che lo stato logico di SLI rimane TRUE, si controlla che il segnale Position non si distacchi oltre un offset superiore (specificato nella finestra Limit pos) né al di sotto di un offset inferiore (specificato nella finestra Limit neg) rispetto al valore salvato all'attivazione del blocco. Se questa condizione è verificata, si attiva l'uscita SLIActive; qualora venisse violato uno dei due limiti, si disattiva l'uscita SLIActive e il blocco va in errore, impostando lo stato logico TRUE sull'uscita Error.</p> <p>L'uscita PositionDiff rimane sempre attiva. Dall'istante in cui si attiva l'ingresso SLI, questa uscita riporta lo scostamento (sia positivo che negativo) tra il valore attuale dell'ingresso Position e il valore salvato all'istante del fronte positivo. Tale uscita assume il valore 0 fintanto che non è presente l'ingresso SLI oppure quando il blocco si trova nello stato di errore.</p> <p>Il controllo termina quando viene rilevato un fronte di discesa del segnale SLI.</p>
Ingressi	<p>SLI: porta di ingresso del segnale di abilitazione del blocco che salva il valore assunto dall'ingresso Position quando viene attivata.</p> <p>Position: porta di ingresso del segnale analogico.</p> <p>Entrambi gli ingressi del blocco funzionale possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul>
Uscite	<p>Error: uscita che si attiva quando si verifica la condizione sopra descritta. Quando questa uscita è attiva, l'uscita SLIActive viene disattivata.</p> <p>SLIActive: uscita che si attiva quando il valore del segnale di ingresso rientra nei limiti previsti e in caso contrario si disattiva.</p> <p>PositionDiff: uscita analogica che rappresenta lo scostamento dall'ingresso Position dal valore salvato sul fronte di salita dell'ingresso SLI.</p>

	<p>Tutte le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	<p>Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p>

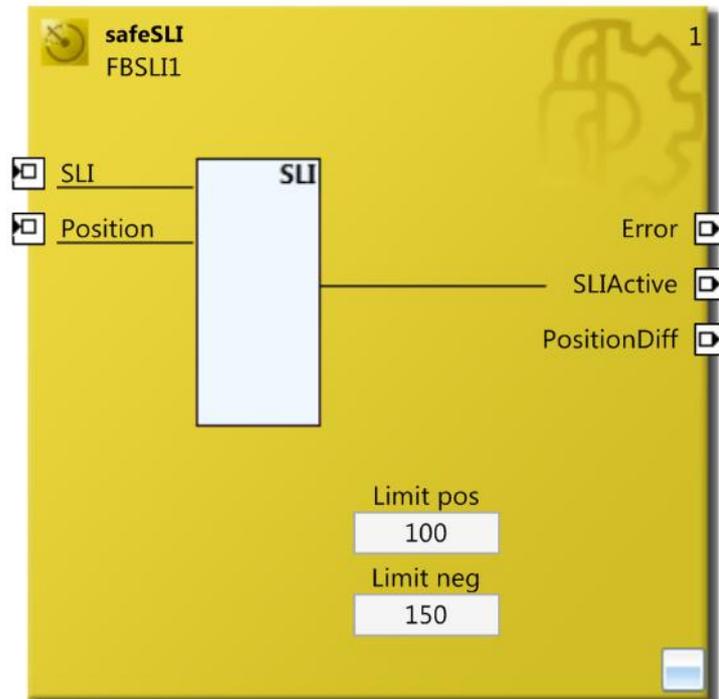


Figura 60: FB safeSLI

## 5.27. ENVELOPE

<b>FB ENVELOPE</b>	
Principio di base	Il blocco funzionale safeEnvelope permette di creare una curva di inviluppo a partire dal valore dell'ingresso InValue. Ogni volta che il blocco viene richiamato verifica che l'ingresso non violi la curva di inviluppo.
Funzionamento	<p>In condizione di regime, l'ingresso SafeFunction si trova nello stato logico TRUE, così come l'uscita SafeFunctionOut. Il blocco realizza il controllo quando si rileva un fronte di discesa sul segnale d'ingresso SafeFunction. Preso come riferimento iniziale l'istante di tempo in cui questo si verifica, si genera una curva di inviluppo definita dai parametri Offset e Max Time (ms): come si vede dalla Figura 62, al crescere del valore di Offset cresce anche la pendenza della curva di inviluppo (rossa a tratto continuo) mentre all'aumentare del valore Max Time (ms) la sua pendenza diminuisce. Per definire completamente la curva di inviluppo è necessario specificare anche il valore della finestra Target Value: questo rappresenta l'intervallo attorno al valore 0 entro il quale si deve assestare il valore del segnale InValue (curva rossa tratteggiata). Il tempo specificato nella finestra Time after in target (ms) inizia a scorrere non appena il segnale InValue raggiunge un valore compreso tra -Target Value e +Target Value. Se il segnale InValue rimane nel range specificato, non appena è trascorso il tempo Time after in target, l'uscita SafeFunctionOut viene impostata allo stato logico FALSE.</p> <p>Mentre scorre il timer se il segnale d'ingresso oltrepassa il limite specificato senza però oltrepassare il valore della curva di inviluppo, lo stesso timer fa ripartire il conteggio del tempo. In un qualsiasi istante se il valore del segnale analogico InValue oltrepassa la curva di inviluppo, il blocco entra nello stato di errore mettendo in sicurezza le uscite.</p>
Ingressi	<p>SafeFunction: ingresso per l'attivazione della funzione.            InValue: porta di ingresso del segnale analogico.            Quest'ultimo può essere collegato a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> <li>• un ingresso di una connessione TwinSAFE (tecnologia TwinSAFE SC)</li> </ul> <p>La stessa cosa non si può dire dell'ingresso SafeFunction, il quale si può collegare solamente a una porta input di un modulo TwinSAFE (per esempio EL1904) oppure all'uscita di un altro blocco funzionale.</p>

Uscite	<p>Error: uscita che si attiva quando si verifica la situazione sopra descritta. Quando questa uscita è attiva, tutte le altre uscite vengono disattivate.</p> <p>SafeFunctionOut: porta di uscita che passa nello stato logico FALSE quando vengono rispettate le condizioni impostate.</p> <p>Entrambe le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	<p>Per resettare la condizione di errore del blocco funzionale è necessario che l'ErrAck del gruppo corrispondente passi allo stato logico TRUE.</p> <p>Se per l'intera durata dell'intervallo Max Time (ms) non viene oltrepassata la curva di involuppo, al termine di tale intervallo l'uscita SafeFunctionOut viene impostata allo stato logico FALSE.</p>

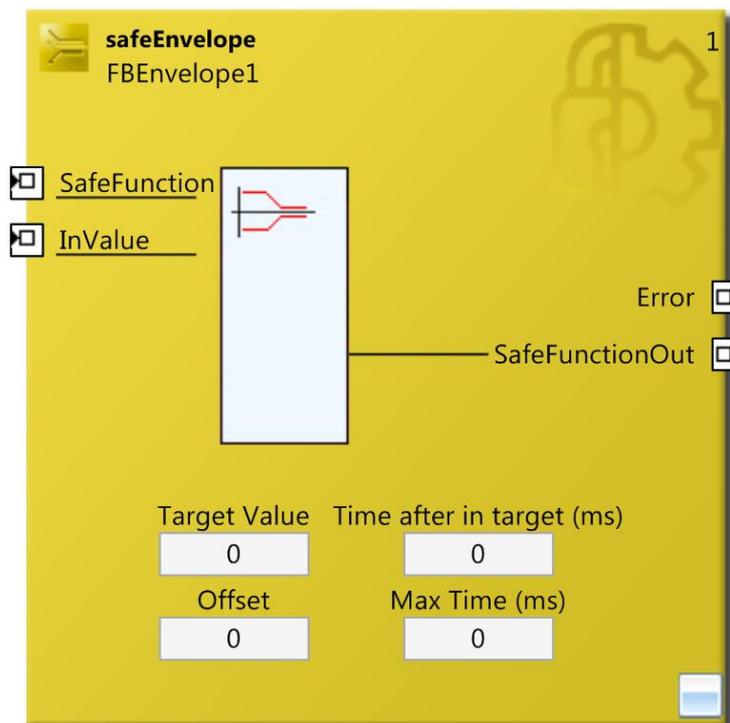


Figura 61: FB safeEnvelope

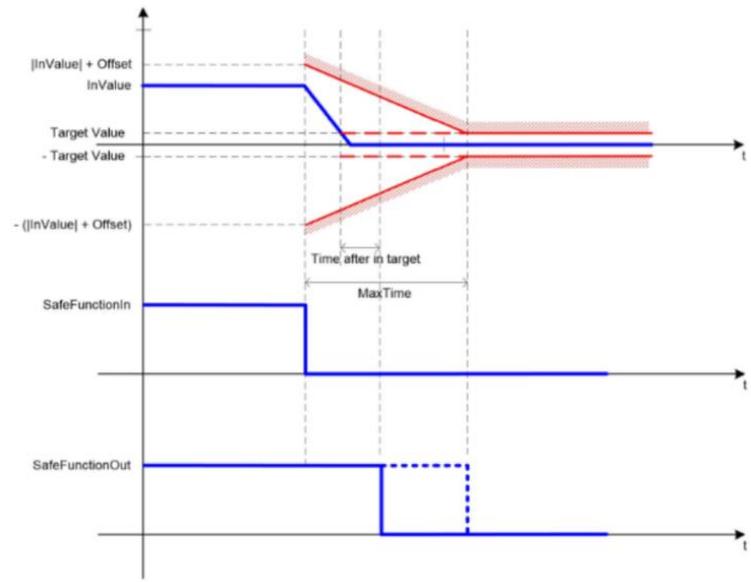


Figura 62: FB logica di funzionamento del blocco safeEnvelope

## 5.28. VIOLATIONCNT

<b>FB VIOLATIONCNT</b>	
Principio di base	Il blocco funzionale safeViolationCNT permette di realizzare un contatore di errore ponderato.
Funzionamento	Tale blocco viene attivato dall'ingresso Enable. Se il contatore è disattivato, l'uscita InputOK si trova nello stato logico di FALSE e l'uscita ActViolationCnt assume valore 0. Se il segnale d'ingresso Enable si trova allo stato logico TRUE, l'ingresso Input viene interrogato. Se questo si trova allo stato logico FALSE, il contatore viene decrementato del valore specificato nella finestra Decrement No Error; se l'ingresso invece si trova allo stato logico TRUE, il contatore viene incrementato del valore specificato nella finestra Increment Error. L'uscita InputOK passa allo stato logico TRUE solamente quando il blocco è attivato e il valore del contatore è inferiore al valore specificato nella finestra Counter Limit.
Ingressi	<p>Enable: porta di ingresso del segnale di abilitazione del blocco.            Input: ingresso che determina se il contatore vada incrementato o decrementato.</p> <p>Entrambi gli ingressi possono essere collegati a:</p> <ul style="list-style-type: none"> <li>• una porta input di un modulo TwinSAFE, per esempio EL1904</li> <li>• un'uscita di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'output nel PLC %Q*)</li> </ul>
Uscite	<p>InputOK: questa uscita rimane nello stato logico TRUE fintanto che il contatore assume un valore inferiore a quello specificato nella finestra Counter Limit.</p> <p>ActViolationCnt: uscita che riporta il valore attuale del contatore.</p> <p>Entrambe le uscite del blocco funzionale possono essere collegate a:</p> <ul style="list-style-type: none"> <li>• una porta output di un modulo TwinSAFE, per esempio EL2904</li> <li>• un ingresso di un altro blocco funzionale</li> <li>• una variabile standard del PLC (l'input nel PLC %I*)</li> </ul>
Note	Se il segnale di ingresso Input viene negato, al suo stato logico FALSE corrisponde un incremento del contatore mentre al suo stato logico TRUE corrisponde un decremento dello stesso.

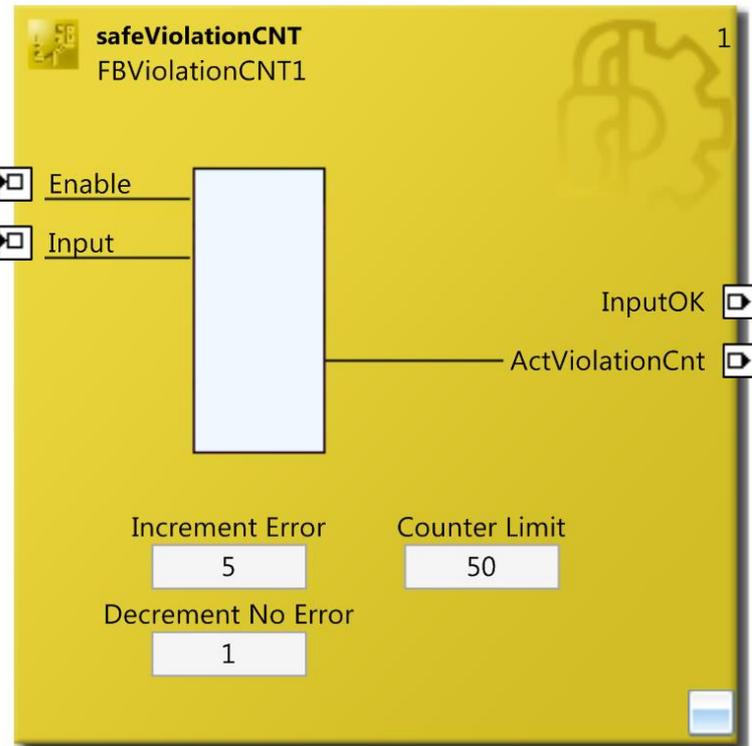


Figura 63: FB safeViolationCNT

## 6. ESEMPI APPLICATIVI

Di seguito vengo riportati alcuni esempi nei quali si propone l'utilizzo di alcuni dei blocchi funzionali descritti, ambientati in contesti lavorativi.

### 6.1. MUTING

Il blocco *safeMuting* è un blocco che consente di realizzare una successione di stati ben definita a priori dal programmatore. Per comprendere a fondo come funziona e come poterlo utilizzare si presenta un esempio pratico.

Si immagini di aver un robot pallettizzatore che provvede a creare il pallet partendo dalle singole unità. Questo si trova all'interno di un'area di lavoro chiusa e recintata, inaccessibile dall'esterno se non che dal personale autorizzato e munito di apposito badge, come rappresentato in Figura 64. Una volta che il pallet è pronto per essere trasferito, l'operatore deve accedere all'area di lavoro del robot e prelevarlo servendosi di un carrello elevatore. Per compiere questa operazione in sicurezza, tutti gli azionamenti del robot devono rimanere disarmati fintanto che l'operatore e il mezzo di movimentazione sono presenti all'interno del volume di lavoro del robot stesso. Quest'ultimo può riprendere a funzionare solamente quando il mezzo di movimentazione con l'operatore a bordo è uscito dal suo volume di lavoro e l'accesso libero al robot è stato bloccato.

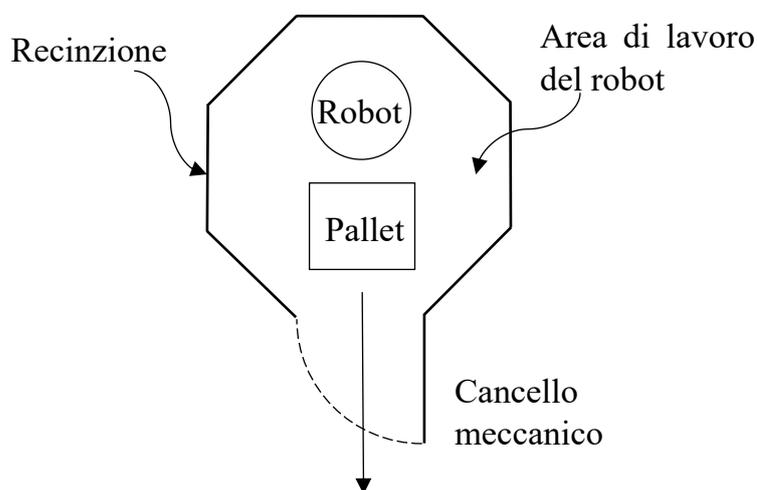


Figura 64: robot pallettizzatore

Operando in queste condizioni si hanno significativi tempi morti del robot che potrebbero essere ridotti con un piccolo accorgimento: si ipotizza di sostituire il cancello meccanico con una barriera elettronica formata dalla successione di tre fotocellule in serie, come rappresentato in Figura 65 e Figura 66.



Figura 65: serie di fotocellule che sostituiscono il cancello meccanico

Se durante il funzionamento del robot il visore di una singola fotocellula venisse oscurato, tutti gli azionamenti del pallettizzatore devono essere disarmati in condizioni di sicurezza, di modo che questo non possa provocare danni all'operatore che si è avvicinato al robot.

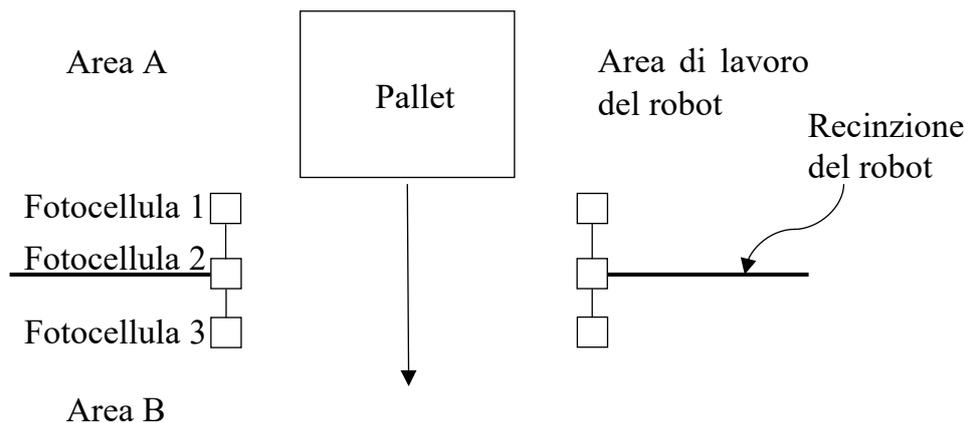


Figura 66: schema della serie di fotocellule che sostituiscono il cancello meccanico

Questa modifica consente di realizzare un piccolo sistema automatico che porta i pallet terminati dalla Area A all'Area B, attraversando le tre fotocellule di sicurezza senza che ciò comporti un arresto del robot, che può continuare a lavorare al pallet successivo. Perché questo sia possibile, si devono realizzare una determinata

successione di stati per il quale, nonostante si oscurino le fotocellule, il sistema non disarma gli attuatori:

1. il pallet è terminato e arriva un segnale di consenso mentre tutte le fotocellule sono attive: tale segnale di consenso deve rimanere attivo per l'intera durata del processo
2. il pallet inizia la fase di uscita, oscurando la fotocellula 1, mentre la fotocellula 2 e la fotocellula 3 rimangono attive
3. il pallet avanza oscurando anche la fotocellula 2, mentre la fotocellula 3 rimane attiva
4. anche la fotocellula 3 viene oscurata
5. si riattiva la fotocellula 1, mentre la fotocellula 2 e la fotocellula 3 rimangono oscurate
6. si riattiva anche la fotocellula 2, mentre la fotocellula 3 rimane oscurata
7. infine si riattiva anche la fotocellula 3 e si disattiva il consenso, tornando nella situazione iniziale e pronti a ricevere un nuovo consenso per far ripartire lo stesso ciclo

Per poter realizzare quanto appena descritto, è necessario utilizzare il blocco *safeMuting* con i collegamenti realizzati come in Figura 67.

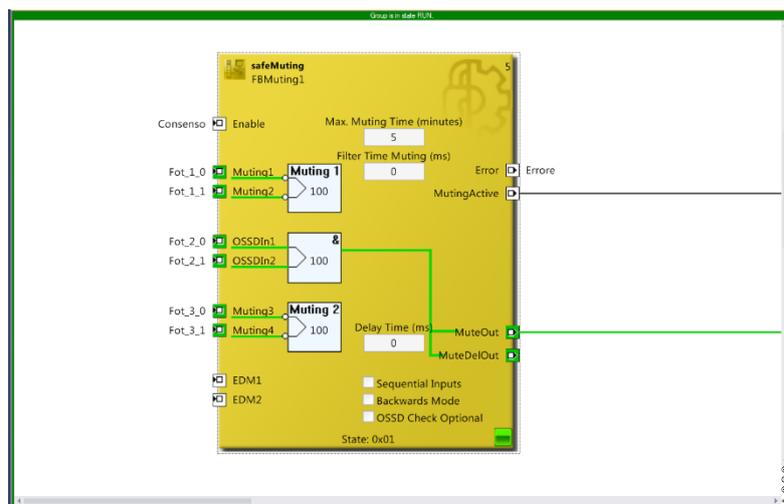


Figura 67: *safeMuting* durante lo stato di esecuzione

Nell'immagine si vede come il canale "Enable" si collega al segnale di consenso proveniente dal robot. Di seguito i 6 canali d'ingresso (Muting1, Muting2, OSSDIn1, OSSDIn2, Muting3, Muting4) rappresentano i segnali delle tre fotocellule: per ognuna di queste si è pensato di utilizzare un doppio segnale d'ingresso, impostando un tempo di discrepanza di 100 ms. Gli ingressi OSSD si distinguono dagli ingressi Muting in quanto nei primi si collega il dispositivo di sicurezza mentre nei secondi dei normali sensori. L'uscita "Error" è stata collegata con un segnale di errore che disarma il robot quando la sequenza di oscuramento delle fotocellule sopracitata non viene rispettata. L'uscita "MutingActive" è invece un segnale che si porta nello stato logico 1 solo quando l'ingresso "Enable" rimane anch'esso nello stato logico 1 e viene rispettata la sequenza di oscuramento delle telecamere: in particolare si attiva

non appena viene oscurata la prima fotocellula e si disattiva solamente quando viene scoperta l'ultima fotocellula. Tale uscita viene utilizzata per inibire il segnale di disarmo del robot quando si oscura una fotocellula (ovvero permette al robot di continuare a lavorare durante l'uscita del pallet). Infine vi sono le uscite "MuteOut" e "MuteDelOut" che riportano lo stato logico del dispositivo di sicurezza; la differenza tra le due uscite è che mentre la prima commuta non appena commuta il segnale d'ingresso della fotocellula 2, il secondo può essere commutato con un tempo di ritardo espresso in millisecondi nella casella "Delay Time (ms)".

Sono inoltre presenti altre due caselle che il programmatore deve riempire per completare questo blocco.

La cella "Max. Muting Time (minutes)" rappresenta l'intervallo di tempo massimo per il quale il blocco può rimanere attivo, ovvero il tempo massimo che può impiegare il pallet per uscire dal volume di lavoro del robot. Se viene superato il tempo consentito dal "Max. Muting Time (minutes)", il blocco funzionale passa nello stato logico di ERROR. Se tale parametro viene impostato a 0, il controllo viene disattivato.

Il "Filter Time Muting (ms)" rappresenta invece il tempo di discrepanza che intercorre tra l'arrivo dei segnali di Muting: per poter impostare il tempo di discrepanza bisogna scrivere il valore 0 in questa casella.

Nel lato riservato agli input sono presenti altri due ingressi che per lo svolgimento di questo esercizio non sono stati utilizzati, in quanto gestiscono i segnali di retroazione.

Nel blocco funzionale sono inoltre presenti tre voci, anch'esse non utilizzate nell'esercizio, che servono per gestire situazioni più complesse di quella presentata.

Se si spunta la casella "Sequential Inputs", il blocco funziona con una logica differente: i segnali di Muting1 e Muting2 non rappresentano più l'ingresso di uno stesso segnale, bensì due segnali differenti provenienti da sensori diversi, che quindi devono arrivare in tempi differenti, ma sempre rispettando l'ordine logico (prima Muting1 e successivamente Muting2); così anche per gli ingressi OSSDIn1, OSSDIn2, che rappresentano gli ingressi di differenti dispositivi di sicurezza. In questa situazione gli ingressi Muting3 e Muting4 possono anche rimanere disabilitati.

Invece, se si spunta la casella "Backwards Mode" la sequenza di attivazione degli ingressi può essere realizzata al contrario, ovvero in seguito al segnale di "Enable" si può attivare per primo il Muting4 e poi di seguito tutti gli altri fino ad arrivare al Muting1.

Infine, se si spunta la casella "OSSO Check Optional" si escludono dalla sequenza gli ingressi dei dispositivi di sicurezza: la sequenza dei segnali contempla quindi solo gli ingressi dei sensori contraddistinti dalla voce "Muting".

Per comprendere a fondo la differenza tra sensori e dispositivi di sicurezza si può fare la seguente considerazione: nell'esempio del pallettizzatore, ogni fotocellula

potrebbe essere sia sensore che dispositivo di sicurezza. Si definisce sensore la fotocellula che come uscita produce lo stato logico 0 e commuta nello stato logico 1 quando il pallet attraversa il suo campo visivo. Si definisce invece dispositivo di sicurezza la fotocellula che come uscita produce lo stato logico 1 e commuta nello stato logico 0 quando il pallet attraversa il suo campo visivo. Di fronte ad una stessa situazione di emergenza però i due elementi si comportano in maniera differente. Si immagina che per un qualche motivo vengano recisi i cavi di alimentazione del sensore: questo in condizioni nominali fornisce un segnale logico 0, ma non essendo alimentato, se un operatore attraversa il campo di visione del sensore, non invia il segnale per disarmare il robot, mettendo a repentaglio la salute del lavoratore e il funzionamento corretto della macchina. Viceversa se vengono recisi i cavi del dispositivo di sicurezza, che normalmente fornisce come segnale logico 1, il PLC che riceve come informazione in ingresso lo stato logico 0 interrompe l'alimentazione del robot, mettendo tutto il sistema in condizioni di sicurezza.

Osservando la Figura 67 si nota che nell'esempio in questione le interfacce utilizzate sono stati tutti dispositivi di sicurezza. Dato che vengono considerati per default collegamenti riservati ai dispositivi di sicurezza, affinché il blocco funzionale *saveMuting* funzioni correttamente è necessario che entrambi gli ingressi OSSDIn siano nello stato logico 1 prima che arrivi il consenso "Enable"; se così non fosse, non appena una fotocellula viene oscurata il PLC disabilita l'alimentazione degli azionamenti, nonostante in precedenza abbia ricevuto il consenso.

Di seguito è riportato in Figura 68 l'intero progetto che governa il disarmo del pallettizzatore presentato in precedenza.

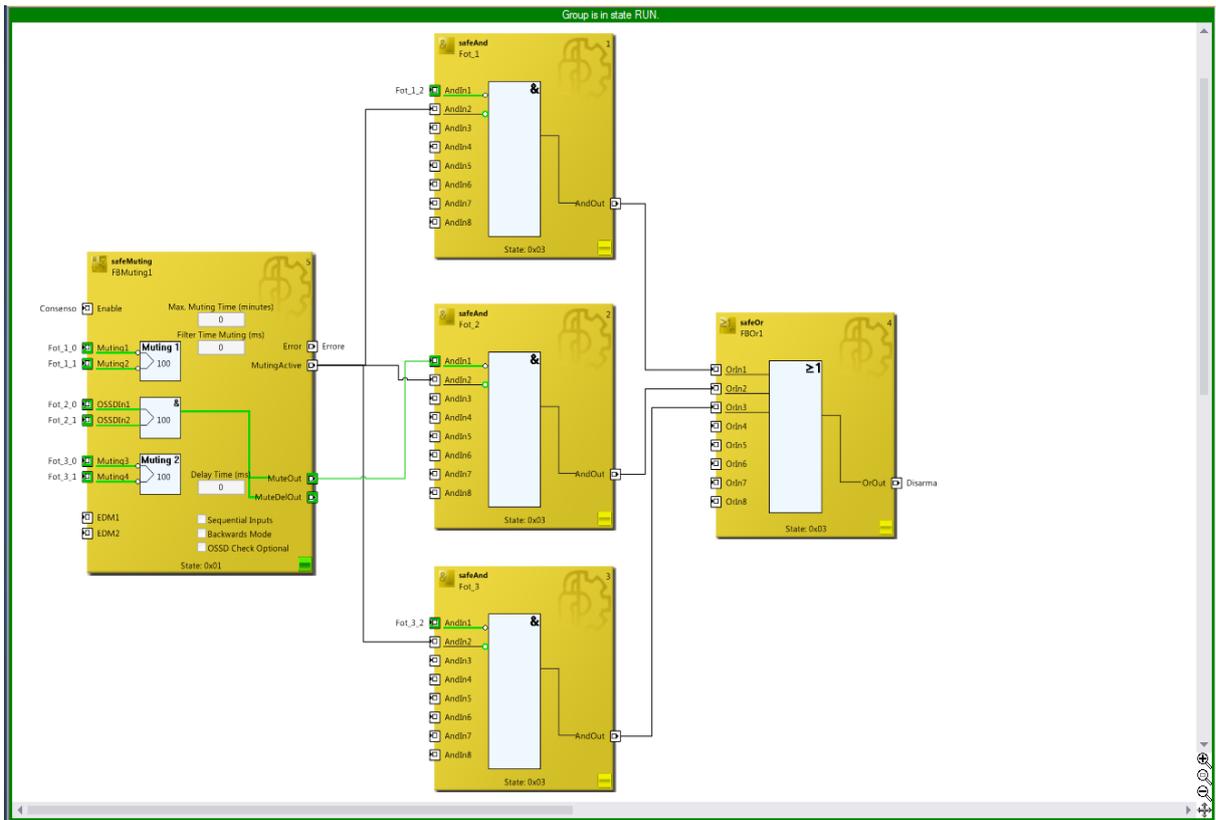


Figura 68: progetto per il disarmo del pallettizzatore

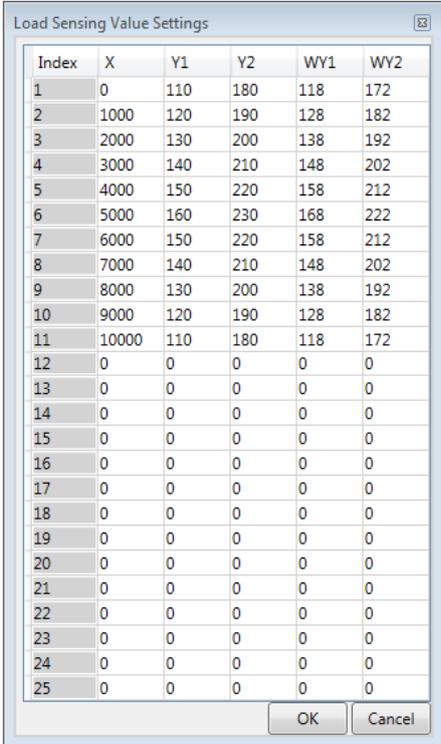
In seguito al blocco di safeMuting sono presenti tre blocchi safeAnd identici, uno per ogni fotocellula: questi verificano istante per istante che ogni singola fotocellula non venga mai oscurata se non che quando sia presente il segnale “MutingActive”. Se dovesse verificarsi il caso per cui il “MutingActive” non sia attivo e una qualsiasi fotocellula venga oscurata, il blocco safeAnd attiva la propria uscita. Le tre uscite dei blocchi safeAnd sono convogliate tutte nel blocco di safeOr, il quale non appena un suo ingresso commuta in uno stato logico TRUE, attiva la propria uscita che disarma il robot.

## 6.2. LOADSENSING

Questo blocco funzionale è stato testato servendosi dell'encoder rotativo e di una base tempi, appositamente creata lato MPS.

È stata introdotta una variabile chiamata Tempo che viene incrementata di un valore fissato (1 nel caso dell'esempio) ad ogni tempo di scansione. Quando questo raggiunge un valore massimo fissato a 10000 il tempo azzerata automaticamente. Il valore di questa variabile è stato quindi passato come ingresso al FB in studio nel canale di ingresso AnalogInX, creando in questo modo un asse tempi. L'altro ingresso analogico, AnalogInY, è stato collegato alla variabile che riporta la posizione angolare dell'encoder. Questo valore non viene utilizzato così come viene letto dal sensore sul campo ma viene scalato con un blocco safeScaling per motivi di comodità pratica. L'ingresso per disattivare il controllo, Inactive, è stato collegato tramite etichetta ad un pulsante bistabile. Le due uscite Valid e Warning sono state collegate tramite etichette a due led che si accendono non appena si attiva la relativa uscita.

In precedenza è stata compilata la tabella per stabilire in ogni intervallo di tempo i valori limite e quelli di warning, che viene di seguito riportata; sempre per motivi di praticità, non è stato utilizzato l'intero campo di variabilità del sensore:



Index	X	Y1	Y2	WY1	WY2
1	0	110	180	118	172
2	1000	120	190	128	182
3	2000	130	200	138	192
4	3000	140	210	148	202
5	4000	150	220	158	212
6	5000	160	230	168	222
7	6000	150	220	158	212
8	7000	140	210	148	202
9	8000	130	200	138	192
10	9000	120	190	128	182
11	10000	110	180	118	172
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0

Tabella 5: tabella del safeLoadSensing utilizzata nell'esempio

Di seguito si riportano delle immagini che mostrano come al variare del tempo e della posizione angolare assunta dall'encoder le uscite si attivano o si disattivano.

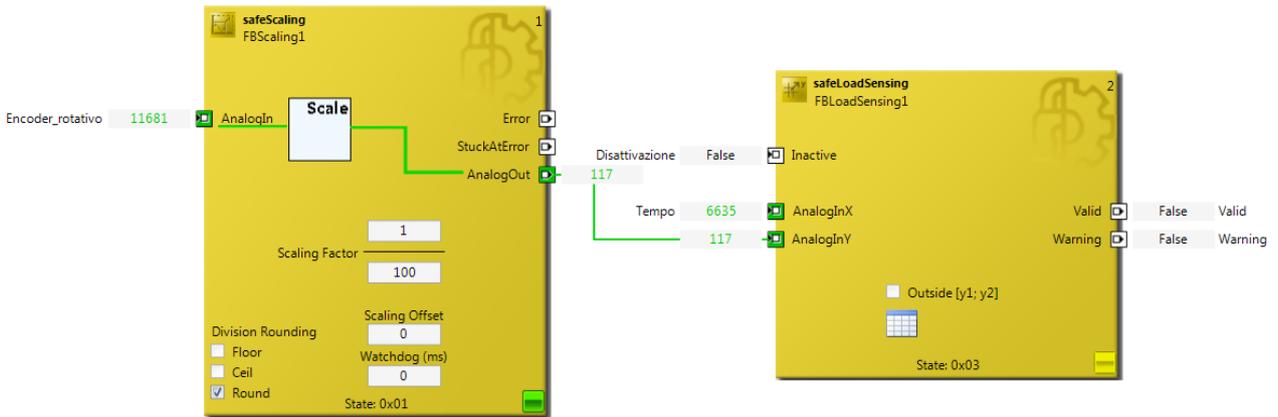


Figura 69: se nell'intervallo di tempo considerato la posizione assunta dall'encoder rotativo non rientra nei limiti definiti, entrambe le uscite vengono disattivate

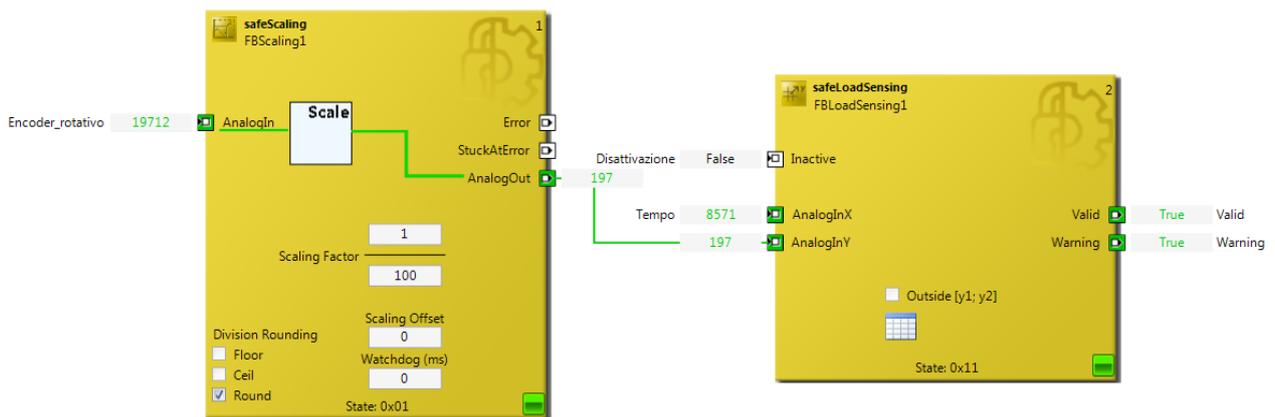


Figura 70: se nell'intervallo di tempo considerato la posizione assunta dall'encoder rotativo rientra nei limiti definiti dai valori di warning dell'intervallo, entrambe le uscite vengono attivate

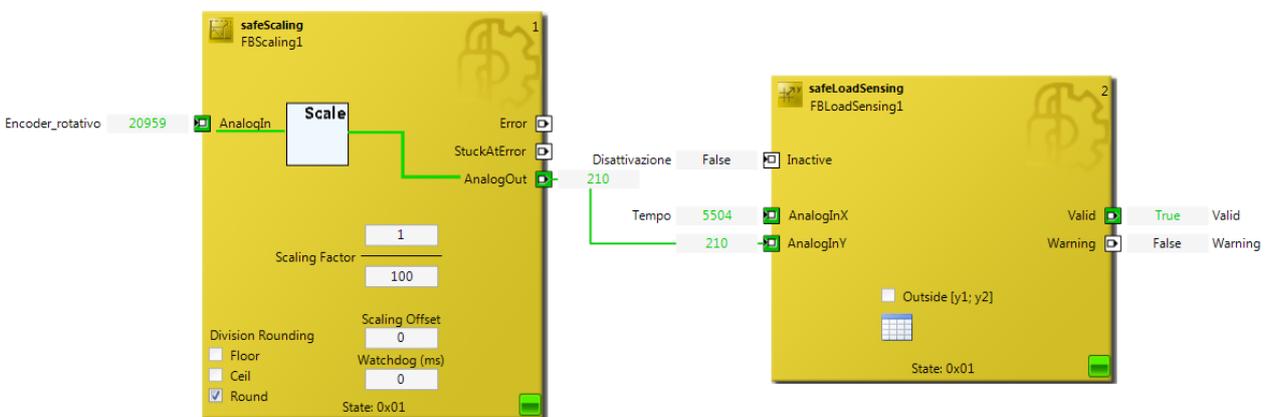


Figura 71: se nell'intervallo di tempo considerato la posizione assunta dall'encoder rotativo rientra nei limiti definiti, solamente l'uscita Valid viene attivata

### 6.3. CAMMONITOR (Excentric Mode)

Il blocco funzionale safeCamMonitor in modalità excentric mode serve per controllare la posizione di arresto di un albero in seguito ad una singola rotazione. Il suo utilizzo può essere inteso come la verifica software della posizione di un asse durante la fase di arresto: si verifica che questo si assesti entro limiti considerati accettabili per l'applicazione e non slitti. Questo inoltre consente di ridurre gli ingombri, in quanto non sono più necessari sensori di posizione come i finecorsa.

Lo studio di questo blocco è stato effettuato montando un encoder sull'albero del motore rotativo a disposizione; per ragioni di praticità, il segnale dall'encoder è stato scalato dal FB safeScaling prima di essere elaborato. Nell'esempio riportato sono stati impostati i parametri rappresentati in Figura 72:

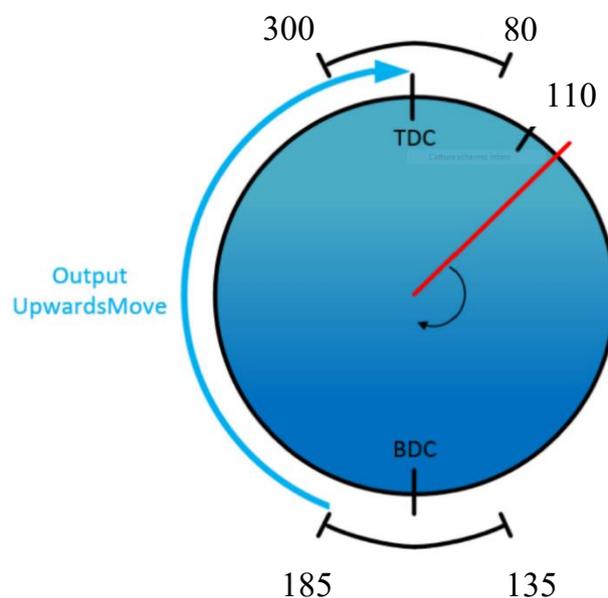


Figura 72: Excentric Mode

Non appena si avvia l'esecuzione del progetto, il FB in studio risulta in uno stato d'errore in quanto non ancora attivato dal primo fronte negativo sull'ingresso Reset, come mostrato in Figura 73.

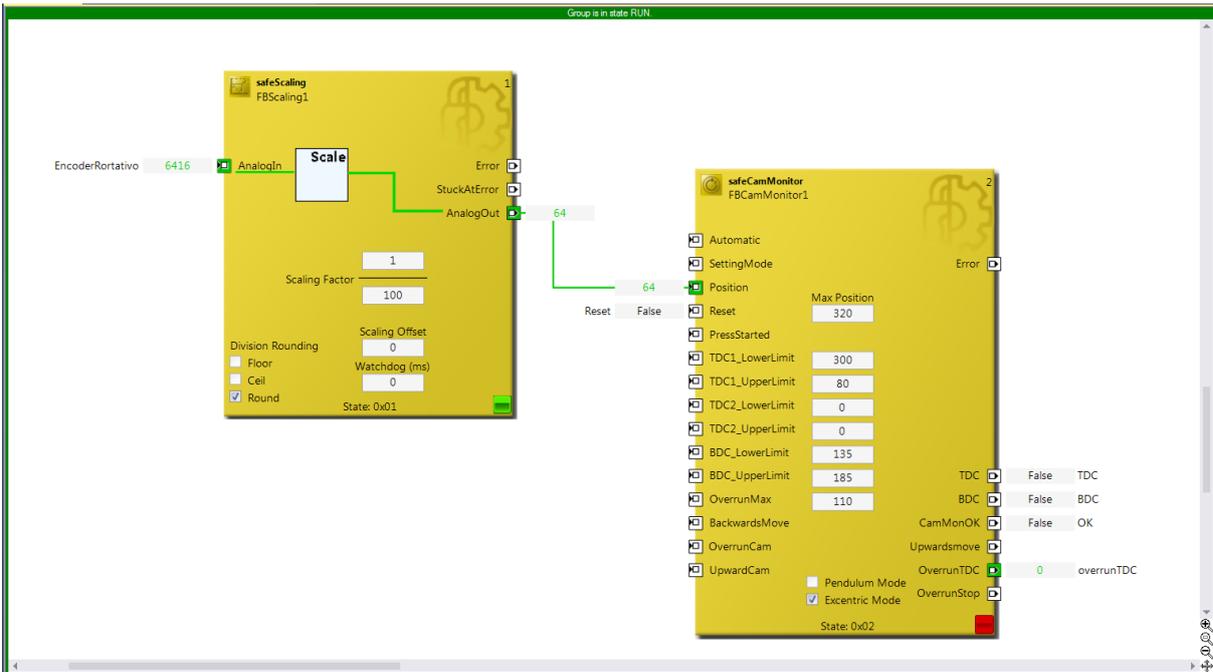


Figura 73: safeCamMonitor in fase di avvio del gruppo

Quando si presenta il fronte negativo sul citato ingresso, il blocco funzionale si porta nello stato di RUN ed è pronto per verificare il posizionamento dell'asse ad esso associato. Se questo termina la propria rotazione all'interno della posizione definita dagli estremi che definiscono il TDC, il blocco funzionale attiva le uscite TDC, CamMonOK e OverrunTDC, come si vede in Figura 74: in questo modo si riporta nella condizione di inizio ciclo in attesa di un nuovo fronte negativo del segnale di Reset.

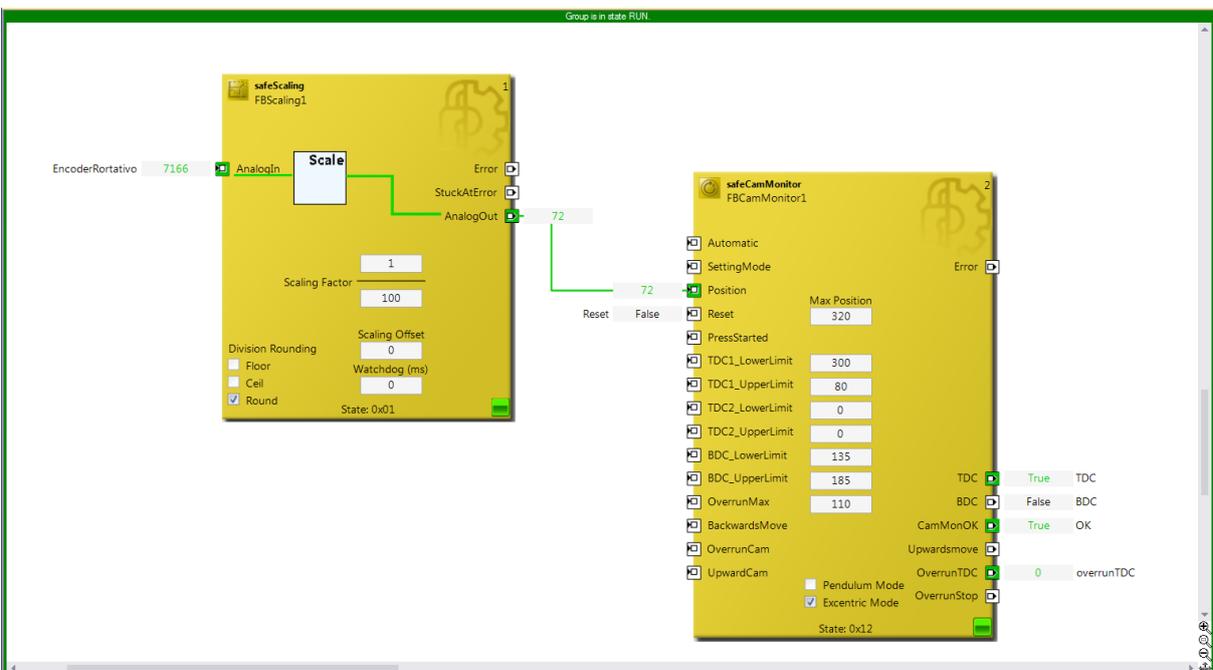


Figura 74: safeCamMonitor in RUN

Qualora il motore rotativo si fermasse oltre il TDC, ma comunque entro l'OverrunMax, le uscite che si attivano sono CamMonOK e OverrunTDC, come riportato in Figura 75.

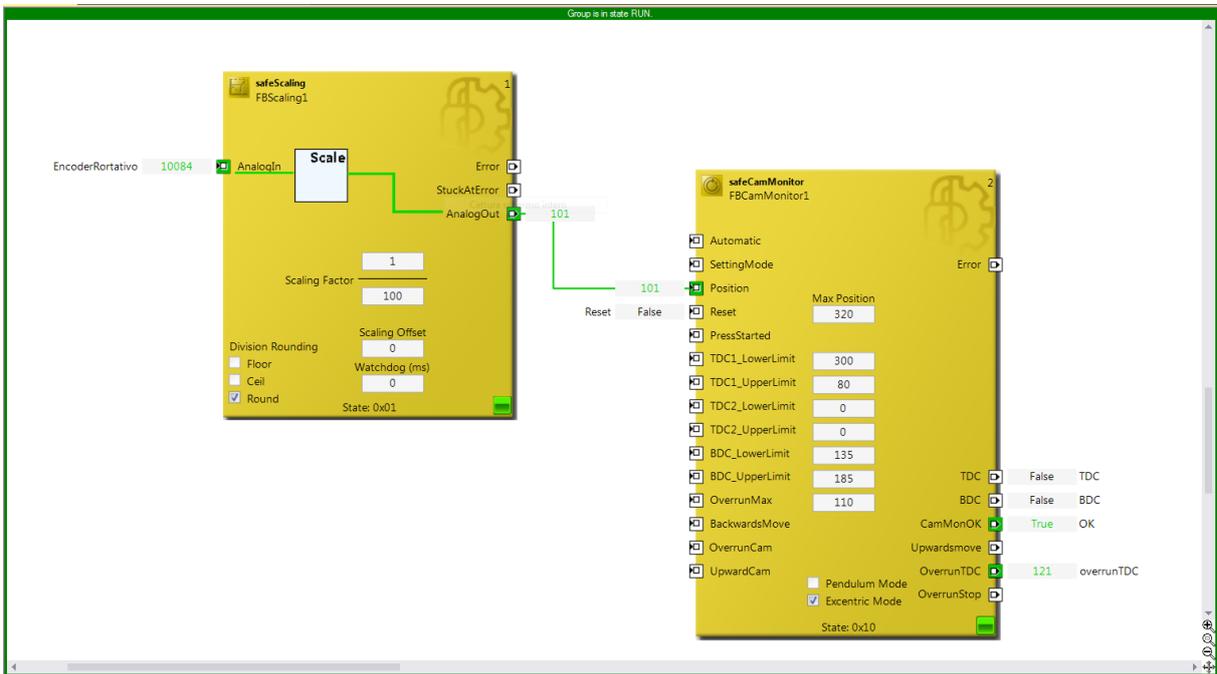


Figura 75: motore rotativo oltre TDC ma entro l'OverrunMax

Durante la fase di rotazione, in corrispondenza degli intervalli definiti, si attivano in successione, e mai contemporaneamente, le uscite TDC, BDC e UpwardsMove; in Figura 76 è riportata un'istantanea durante la fase di risalita, in cui la sola uscita UpwardsMove tra quelle appena citate è attiva.

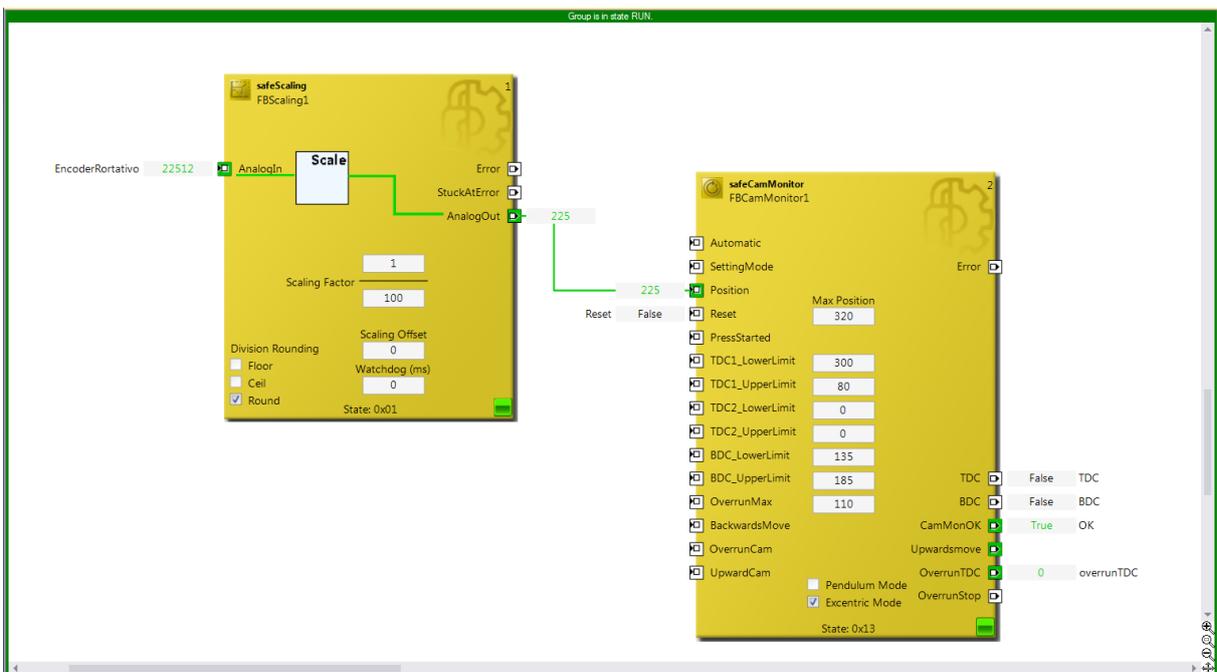


Figura 76: motore rotativo in risalita, oltre BDC

Se per qualche motivo l'asse invece di fermarsi entro la posizione definita dall'OverrunMax prosegue la propria rotazione, il blocco funzionale si porta nello stato di errore, attivando l'uscita Error e disattivando tutte le altre uscite. La stessa configurazione di errore può esser raggiunta qualora durante la fase di rotazione il motore si arresta per un tempo superiore a Stop Detection Time in una posizione qualsiasi.

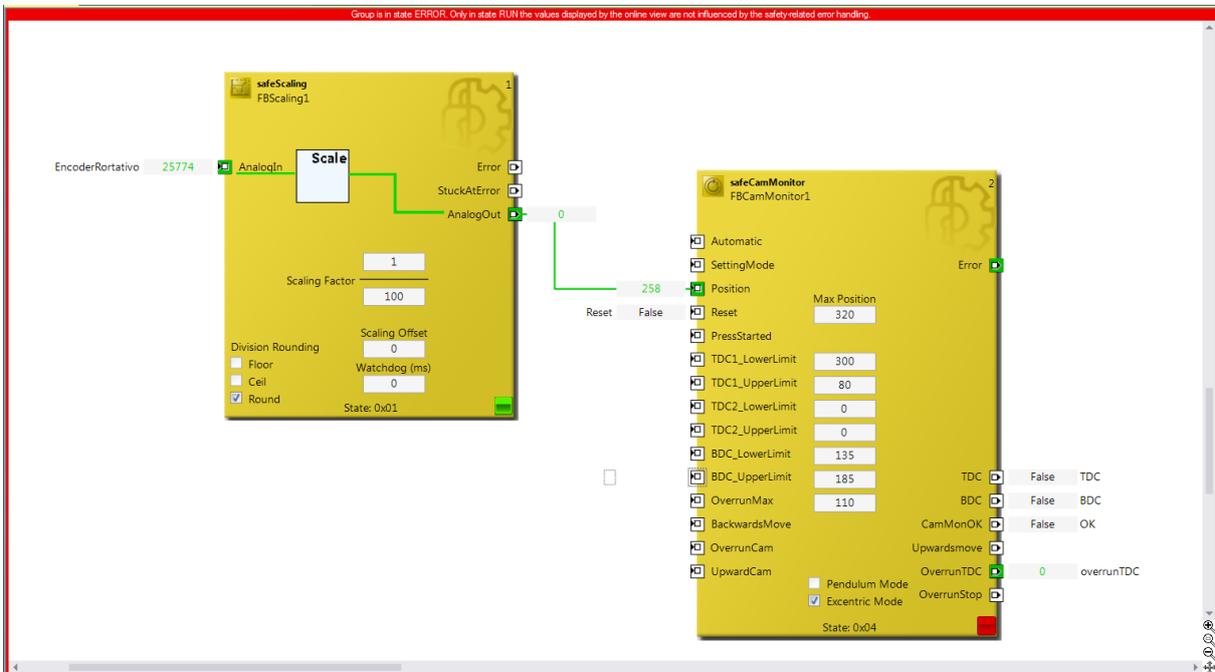


Figura 77: motore rotativo arrestato in posizione non corretta

## 6.4. COMPARE, LIMIT, MON

Di seguito si riporta un esempio di come si può utilizzare la tecnologia TwinSAFE SC per gestire la misura del livello di un liquido in un serbatoio. In questo caso, la misura viene realizzata con due sensori differenti. Il primo è un sensore ultrasonico il cui segnale è collegato ad un ingresso del modulo EL3174-0090, il quale riceve in ingresso un segnale analogico I/O – Link e lo trasmette al modulo safety EL6910 che lo elabora. Questo segnale viene considerato quindi sicuro. Il secondo è una sonda di livello, disposta sul fondo del serbatoio: il segnale proveniente da questo sensore viene collegato ad un modulo standard EtherCAT, quindi considerato non sicuro.

La logica del modulo safety 6910 permette di confrontare i due segnali provenienti dai differenti sensori grazie al FB safeCompare, dopo che il primo è stato scalato con un safeScaling per eguagliare i range di variabilità dei due segnali. Inoltre, si verifica con un FB safeLimit che il valore rilevato sul canale sicuro rientri entro certi limiti, per evitare che il serbatoio venga riempito più della propria capacità massima. I segnali di uscita StuckAtError, IsValid e InLimit dei blocchi funzionali citati vengono poi portati agli ingressi del FB safeMon, la cui uscita è collegata con il modulo d'uscita safety EL2904. Come si vede dalla Figura 78, fintanto che è assente il segnale StuckAtError (ingresso negato nel FB safeMon) e sono attivi i segnali IsValid e InLimit, non appena rileva la corretta sequenza di stati logici sull'ingresso Restart, il blocco funzionale in esame attiva le proprie uscite, chiudendo i contatti K1 e K2. Nell'immagine sono stati rappresentati con linea continua nera i collegamenti fisici, mentre con linea tratteggiata azzurra i collegamenti software.

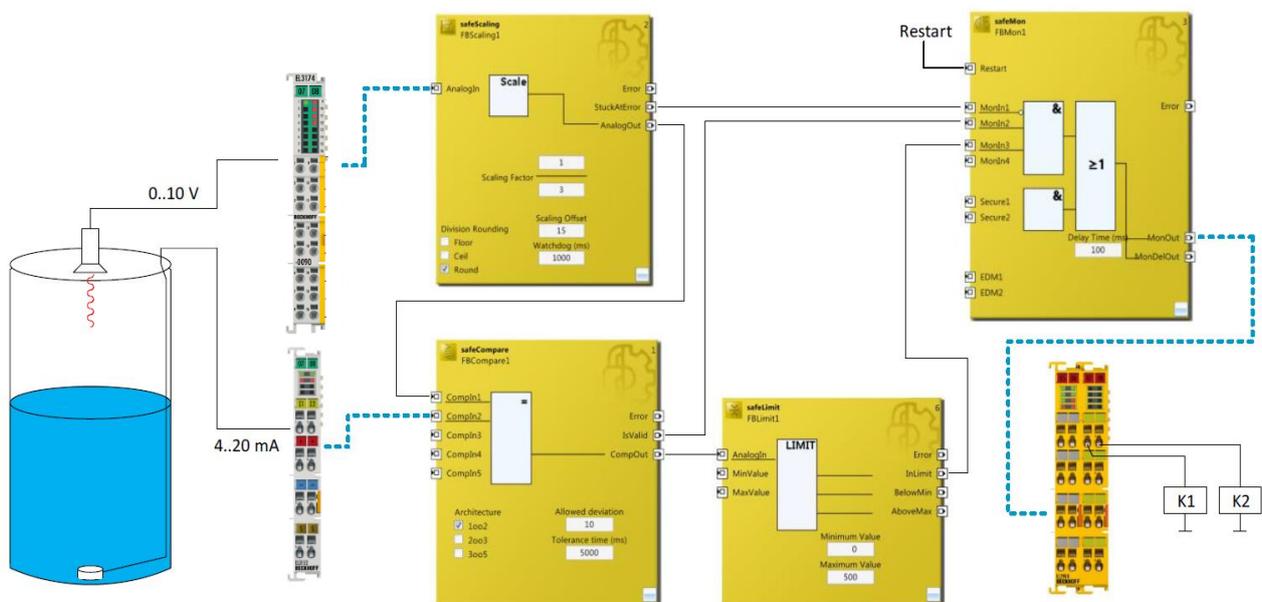


Figura 78: Schema per il controllo del livello del liquido in un serbatoio

## 7. CONCLUSIONI

In primo luogo durante l'attività di tesi è stato presentato il gruppo aziendale, i prodotti che esso propone sul mercato, la storia dalla quale è nato e si è sviluppato, la filosofia che sta alla base delle scelte aziendali orientate ad un futuro in cui l'obiettivo è quello della crescita sia dell'intero gruppo che delle persone al suo interno.

In seguito sono stati affrontati argomenti più tecnici, legati al mondo della programmazione delle macchine automatiche: è stata presentata la loro architettura hardware e software, partendo da un punto di vista più generale per poi approfondire il discorso sulla sicurezza, in particolare sul mondo Safety.

L'attività di tesi è servita per approfondire la conoscenza del software TwinCAT, utilizzato in azienda per la programmazione dei PLC. Dato che tale software viene utilizzato da poco tempo, non se ne conoscono appieno le potenzialità: al fine di ampliare il knowhow aziendale, è stato studiato il funzionamento dei blocchi funzionali presenti all'interno delle sue librerie, con particolare attenzione a quei FB che gestiscono segnali analogici, non ancora utilizzati in ambito produttivo. In futuro sarà interessante comprendere come questi possano essere utilizzati a bordo macchina per controllare tutte le operazioni tipiche della realtà produttiva. In quest'ottica, durante l'attività di tesi si è affrontato il tema anche da un punto di vista pratico: nell'area ricerca e sviluppo è stato mostrato un piccolo robot che l'azienda sta studiando sul quale è stato possibile applicare alcune delle nozioni apprese.

Inoltre, grazie all'attività di tesi ho avuto l'opportunità di entrare in contatto con il mondo lavorativo; questo mi ha permesso di vedere, capire e apprezzare tutto il lavoro, la progettazione e la programmazione che si nascondono dietro le macchine automatiche.

In conclusione, il lavoro di ricerca svolto vuole esse un contributo per l'innovazione tecnologica per un'automazione industriale sempre più precisa, flessibile e competitiva.

## 8. BIBLIOGRAFIA

- ❖ Alberto Martini. Integrazione del concetto di Clock distribuito in un Master EtherCAT per la valutazione delle performance di moduli eXtreme Fast Control. Tesi di Laurea AA 2010/2011
  
- ❖ Documentazione Marchesini Group
  - Manuale MPS2000
  - Manuale MAC2000
  - Manuale moduli Safety
  - Guida ai moduli Safety (FSOE) Beckhoff
  
- ❖ Documentazione Beckhoff
  - Application Guide TwinSAFE
  - EL6224-00x0 IO-Link Terminal
  - TwinCAT Safety PLC
  - TwinSAFE Logic FB
  - Operation Manual for EL6910 TwinSAFE Logic Terminal