



Istruzioni generali per i controllori programmabili Micro800

Numeri di catalogo 2080-LC10, 2080-LC20, 2080-LC30,
2080-LC50, 2080-LC70

Pubblcazione Rockwell Automation 2080-RM001J-IT-E - Marzo 2021
Sostituisce la pubblicazione -2080-RM001I-IT-E - Gennaio 2019



Importanti informazioni utente

Leggere questa documentazione e i documenti elencati nella sezione delle risorse aggiuntive riguardo l'installazione, la configurazione e l'operatività di questa attrezzatura prima di installare, configurare, attivare o eseguire la manutenzione di questo prodotto. Agli utenti viene richiesto di acquisire familiarità con le istruzioni per l'installazione e il cablaggio, oltre al soddisfacimento dei requisiti di tutti i codici, leggi e normative in vigore.

È necessario che attività come installazione, regolazione, messa in servizio, utilizzo, assemblaggio, disassemblaggio e manutenzione siano eseguiti da personale opportunamente formato in ottemperanza al codice di pratica.

Se questa attrezzatura viene utilizzata in una modalità non specificata dal produttore, la protezione fornita da questa attrezzatura potrebbe risultarne compromessa.

In nessun caso Rockwell Automation, Inc. potrà essere ritenuto responsabile dei danni indiretti o consequenziali risultanti dall'applicazione di queste apparecchiature.

Gli esempi e i diagrammi in questo manuale sono inclusi a solo scopo informativo. A causa delle numerose variabili e requisiti associati a una particolare installazione, Rockwell Automation, Inc. non può essere ritenuto responsabile dell'uso effettivo basato sugli esempi e i diagrammi.

Nessuna responsabilità riguardo brevetti viene assunta da Rockwell Automation, Inc. in relazione all'utilizzo di informazioni, circuiti, apparecchiature o software descritti in questo manuale.

È proibita la riproduzione del contenuto di questo manuale, in parte o in toto, senza il consenso scritto di Rockwell Automation, Inc.

All'interno del manuale, quando necessario, vengono utilizzate note per evidenziare considerazioni sulla sicurezza.



AVVERTENZA: Identifica informazioni relative a procedure o circostanze che possono provocare un'esplosione in un ambiente pericoloso, che può portare a lesioni personali o morte, danni materiali o perdite economiche.



ATTENZIONE: Identifica informazioni relative a procedure o circostanze che possono portare a lesioni personali o morte, danni materiali o perdite economiche. I segnali di attenzione consentono di identificare un pericolo, evitare un pericolo e riconoscere le conseguenze.

IMPORTANTE Identifica informazioni che sono critiche per la corretta applicazione e comprensione del prodotto.

Le etichette che forniscono precauzioni specifiche possono trovarsi all'esterno o all'interno dell'attrezzatura.



RISCHIO DI FOLGORAZIONE: Etichette possono essere applicate sull'apparecchiatura o al suo interno, ad esempio una trasmissione o un motore, per avvisare le persone della possibile presenza di tensione elettrica pericolosa.



RISCHIO DI USTIONI: Etichette possono essere applicate sull'apparecchiatura o al suo interno, ad esempio una trasmissione o un motore, per avvisare le persone che le superfici possono raggiungere temperature pericolose.



RISCHIO DI ARCO ELETTRICO: Etichette possono essere applicate sull'apparecchiatura o al suo interno, per esempio il centro di controllo motore, per avvisare le persone del possibile rischio di arco elettrico. L'arco elettrico può causare lesioni gravi e in casi estremi morte. Indossare dispositivi di protezione individuale (DPI) adeguati. Seguire TUTTE le disposizioni in materia di sicurezza sul lavoro e di dispositivi di protezione individuale (DPI).

Introduzione	In questo manuale	13
	Controllori supportati	13
	Risorse aggiuntive	13
	Avvisi legali	14
	Capitolo 1	
Ricerca di informazioni sulle istruzioni e sugli elementi ladder	Blocchi di istruzione	15
	Operatori	15
	Funzioni	16
	Blocchi funzione	17
	Istruzione impostate in ordine alfabetico	18
	Capitolo 2	
Linguaggio LD (Ladder Diagram)	Programmazione in Ladder Diagram	25
	Ambiente di sviluppo del programma LD	26
	Barra degli strumenti Istruzione (LD)	26
	Aggiungere elementi istruzione dalla barra degli strumenti	
	Istruzione	27
	Capitolo 3	
Elementi del ladder diagram (LD)	Piolo	29
	Aggiunta di un ramo a un programma LD	30
	Etichette ramo	31
	Commenti piolo	32
	Diramazione	33
	Aggiunta di un ramo a un programma LD	34
	Blocco istruzione (LD)	37
	Bobina	37
	Aggiunta di elementi a bobina	38
	Bobina diretta	41
	Bobina invertita	41
	Bobina fronte di discesa impulso	42
	Bobina fronte di salita impulso	43
	Bobina impostata	44
	Bobina ripristinata	45
	Contatto	47
	Aggiunta di un contatto a un programma LD	47
	Contatto diretto	50
	Contatto invertito	51
	Contatto fronte di salita impulso	51
	Contatto fronte di discesa impulso	52
	Ritorno	53
	Salto	54
	Blocchi di istruzione nei programmi LD	55
	Utilizzo nell'editor di lingue LD	56
	Esempi di programma ladder diagram (LD)	57
	Tasti di scelta rapida LD	59

	Capitolo 4	
Istruzione allarme	LIM_ALRM (limite allarme).....	63
	Capitolo 5	
Istruzioni aritmetiche	ABS (valore assoluto)	67
	ACOS (arcocoseno di origine)	69
	ACOS_LREAL (arcocoseno Long Real).....	71
	Addition	72
	ASIN (arcoseno).....	74
	ASIN_LREAL (arcoseno Long Real)	76
	ATAN (arcotangente)	77
	ATAN_LREAL (arcotangente Long Real).....	79
	COS (coseno).....	81
	COS_LREAL (coseno Long Real).....	82
	Division.....	84
	EXPT (esponente).....	86
	LOG (log base 10).....	87
	MOD (modulo).....	89
	MOV (sposta)	91
	Multiplication.....	92
	Neg (negazione).....	94
	POW (incremento alimentazione).....	95
	RAND (valore casuale).....	97
	SIN (seno)	99
	SIN_LREAL (seno Long Real).....	100
	SQRT (radice quadrata)	102
	Subtraction	103
	TAN (tangente).....	105
	TAN_LREAL (tangente Long Real)	107
	TRUNC (troncamento).....	108
	Capitolo 6	
Istruzioni porta seriale ASCII	ABL (prova ASCII per riga del buffer)	111
	ACL (buffer cancellazione ASCII)	113
	AHL (righe di arbitraggio ASCII)	115
	ACB (caratteri ASCII nel buffer)	118
	ARD (lettura ASCII)	120
	ARL (riga di lettura ASCII)	123
	AWA (scrittura ASCII aggiunta)	125
	AWT (scrittura ASCII)	128
	Dettagli dei parametri ASCII	130
	Tipo di dati ABLACB	131
	Tipo di dati ACL	131
	Tipo di dati AHL ChannelSts	131
	Tipo di dati AHLI	132
	Tipo di dati ARDARL.....	132
	Tipo di dati AWAAWT	133
	Capitolo 7	

Istruzioni binarie	AND_MASK (maschera AND).....	135
	BSL (spostamento del bit a sinistra)	137
	BSR (spostamento del bit a destra).....	141
	NOT_MASK (maschera NOT bit-to-bit)	144
	OR_MASK (maschera OR bit-to-bit)	146
	ROL (rotazione a sinistra).....	147
	ROR (rotazione a destra)	149
	SHL (spostamento a sinistra)	151
	SHR (spostamento a destra)	153
	XOR_MASK (maschera OR esclusivo)	155
	Capitolo 8	
Istruzioni booleane	F_TRIG (rilevamento fronte di discesa)	159
	R_TRIG (rilevatore fronte di salita)	161
	RS (reset/impostazione).....	163
	OR.....	164
	AND.....	165
	XOR (OR esclusivo)	166
	NOT.....	166
	SR (impostazione/reset).....	167
	TTABLE (tavola della verità).....	169
	MUX8B (multiplexer di ingressi BOOL 8)	172
	MUX4B (multiplexer di ingressi BOOL 4)	175
	Capitolo 9	
Istruzioni di comunicazione	COM_IO_WDOG	179
	MSG_CIPGENERIC (messaggio generico per Common Industrial Protocol).....	182
	Tipo di dati CIPAPPCFG.....	184
	Tipo di dati CIPCONTROLCFG	185
	Tipo di dati CIPSTATUS	186
	Bit di stato CIPSTATUS.....	187
	Codici di errore CIPSTATUS	187
	Tipo di dati CIPTARGETCFG	188
	Percorso di destinazione per la messaggistica CIP	189
	Connessioni per messaggio CIP/EIP	190
	Timer per timeout messaggio CIP	191
	MSG_CIPSYMBOLIC (messaggio simbolico per Common Industrial Protocol)	191
	Tipo di dati CIPSYMBOLICCFG	193
	Sintassi scrittura/lettura dei messaggi simbolici.....	194
	Dimensione pacchetto dati supportata per la funzione CIP seriale	195
	MSG_MODBUS (messaggio modbus)	196
	Tipo di dati MODBUSLOCPARA.....	198
	Processo di esecuzione del messaggio (Rung = TRUE)	200
	Tipo di dati MODBUSTARPARA.....	202
	MSG_MODBUS2 (messaggio MODBUS/TCP)	202
	Tipo di dati MODBUS2LOCPARA.....	205

Tipo di dati MODBUS2TARPARA	206
Processi di esecuzione di messaggi e diagrammi di temporizzazione	208
Processo di esecuzione del messaggio (generale)	209
Processo di esecuzione del messaggio (Rung = TRUE)	210
Diagramma di temporizzazione dell'esecuzione del messaggio (Rung = TRUE)	211
Processo di esecuzione del messaggio (Rung = FALSE)...	211
Diagramma di temporizzazione dell'esecuzione del messaggio (Rung = FALSE).....	212
Processo di esecuzione di messaggi (errore)	213
Diagramma di temporizzazione dell'esecuzione del messaggio (errore)	214
Utilizzo dei blocchi funzione di comunicazione (messaggio)	215
Configurazione dei valori dei dati oggetto per i messaggi espliciti (MSG_CIPGENERIC)	215
Esempio: Come creare un programma di generazione messaggi MSG_CIPGENERIC per leggere i dati da un controllore	217
Identificazione dei valori iniziali per le variabili di input, MSG_CIPGENERIC	218
Aggiunta di variabili e di un blocco funzione MSG_CIPGENERIC	219
Configurazione dei valori iniziali per le variabili	220
Aggiunta di un contatto e di una bobina.....	222
Aggiunta di un blocco funzione COP, di variabili e di un contatto (MSG_CIPGENERIC)	223
Verifica della corretta configurazione IP sul controllore B	225
Esempio: Come creare un programma di generazione messaggi MSG_CIPSYMBOLIC per scrivere i dati in un controllore	226
Aggiunta di un blocco funzione COP, di variabili e di un contatto (MSG_CIPSYMBOLIC).....	227
Aggiunta di un operatore Equal e di una bobina.....	228
Aggiunta di un blocco funzione MSG_CIPSYMBOLIC, di variabili e di un contatto	230
Verifica della corretta configurazione IP sul controllore B	232
Creazione di una variabile globale per il controllore B... ..	233
Revisione dei risultati del programma completo	234
Esempio: Come configurare una comunicazione Modbus per la lettura da e la scrittura in una unità	236
Supporto per protocolli di comunicazione.....	240
Canali di comunicazione integrati	242

Capitolo 10

Istruzioni per il confronto

Uguale	243
Greater than.....	246
Greater than or equal	247

	Less than.....	249
	Less than or equal.....	250
	Not equal.....	251
	 Capitolo 11	
Istruzioni contatore	CTD (conteggio decrescente)	255
	CTU (conteggio crescente)	257
	CTUD (conteggio crescente decrescente).....	259
	 Capitolo 12	
Istruzioni per la conversione dati	ANY_TO_BOOL.....	263
	ANY_TO_BYTE	264
	ANY_TO_DATE	265
	ANY_TO_DINT.....	267
	ANY_TO_DWORD	268
	ANY_TO_INT	269
	ANY_TO_LINT	270
	ANY_TO_LREAL.....	272
	ANY_TO_LWORD.....	273
	ANY_TO_REAL.....	274
	ANY_TO_SINT	275
	ANY_TO_STRING	276
	ANY_TO_TIME	277
	ANY_TO_UDINT	278
	ANY_TO_UINT.....	279
	ANY_TO_ULINT.....	280
	ANY_TO_USINT	281
	ANY_TO_WORD.....	282
	 Capitolo 13	
Istruzioni per l'elaborazione dei dati	MEDIA	285
	COP	287
	Copia in un tipo di dati differente.....	290
	Esempio di array della stringa COP	290
	MIN (minimo).....	292
	MAX (massimo).....	294
	 Capitolo 14	
Istruzioni per l'High Speed Counter (HSC)	Che cos'è l'High-Speed Counter?	297
	HSC (High Speed Counter).....	298
	Valori HSCCmd.....	300
	Tipo di dati HSCAPP.....	301
	Tipo di dati HSCSTS.....	304
	Tipo di dati PLS.....	311
	Tipo di dati HSCE_CHANNEL	312
	Tipo di dati HSCE_STS.....	313
	Tipo di dati PLS_HSCE.....	314

HSC_SET_STS (stato di impostazione dell'High-Speed Counter)	314
Utilizzo delle istruzioni dell'High-Speed Counter	317
Aggiornamento dei dati dell'applicazione HSC	317
Finestra di dialogo Interruzione da parte dell'utente dell'High-Speed Counter (HSC)	318
Configurazione di un Interrupt utente dell'High Speed Counter (HSC)	318
Aggiunta e configurazione di un'interruzione da parte dell'utente dell'High-Speed Counter (HSC)	319
Proprietà interruzione HSC	320
Parametri interruzione HSC	321
Configurazione di un interruttore di fine corsa programmabile (PLS)	323
Esempio: come creare un programma High-Speed Counter (HSC)	324
Creazione di un ladder diagram e aggiunta di variabili	325
Assegnazione di valori alle variabili HSC	328
Assegnazione di variabili e compilazione del programma	329
Test del programma ed esecuzione dell'High-Speed Counter	331
Add a Programmable Limit Switch (PLS) function	334
Esempio: interruttore fine corsa programmabile (PLS) abilitato	335

Capitolo 15

Istruzioni HSCE

HSCE	337
HSCE_CFG	341
HSCE_CFG_PLS	344
HSCE_READ_STS	346
HSCE_SET_STS	348
Codici di errore HSCE	351

Capitolo 16

Istruzioni ingresso/uscita

LCD	356
LCD_BKLT_REM (retroilluminazione LCD remoto)	358
LCD_REM (LCD remoto)	360
RHC (lettura high speed clock)	364
RPC (lettura checksum del programma)	365
DLG (registro dati)	366
IIM (ingresso immediato)	368
IOM (uscita istantanea)	370
KEY_READ (lettura tasti su LCD)	372
KEY_READ_REM (lettura chiavi per LCD remoto)	375
MM_INFO (informazioni sul modulo di memoria)	377
Tipo di dati MMINFO	380
MODULE_INFO	380

MODULE_INFO - Informazioni sui moduli plug-in e di espansione.....	384
Diagrammi di temporizzazione dell'istruzione MODULE_INFO	386
PLUGIN_INFO (informazioni sui plug-in).....	392
PLUGIN_READ (lettura plug-in)	394
PLUGIN_RESET (reset plug-in)	397
PLUGIN_WRITE (plug-in di scrittura)	398
RCP (ricetta).....	400
RTC_READ (lettura Real-Time Clock)	403
RTC_SET (impostare Real-Time Clock).....	405
SYS_INFO.....	407
Tipo di dati SYS_INFO.....	409
TRIMPOT_READ (lettura trimpot)	410
Definizione ID Trimpot.....	411

Capitolo 17

Istruzioni di interruzione

STIS (selezione avvio temporizzato).....	413
UIC (bit clear interrupt lost)	415
UID (disattivazione interrupt)	416
UIE (abilitazione interrupt).....	418
UIF (interrupt flush in attesa).....	420

Capitolo 18

Istruzioni di controllo del movimento

Regole generali per i blocchi funzione di controllo movimento.	424
Dettagli relativi ai parametri del blocco funzione per il controllo del movimento	426
Stati asse di controllo movimento	426
Numeri dei parametri del blocco funzione per il controllo del movimento	429
ID errori del blocco funzione di controllo movimento.....	430
Scenari di errore asse	431
Tipo di dati AXIS_REF	432
Tipo di dati FB_AXIS_REF.....	433
Variabili dell'asse	434
MC_AbortTrigger (trigger di interruzione del controllo movimento)	434
MC_Halt (halt di controllo movimento)	437
MC_Home (home del controllo movimento)	440
MC_MoveAbsolute (movimento assoluto del controllo movimento)	445
MC_MoveRelative (movimento relativo del controllo movimento)	448
MC_MoveVelocity (velocità di movimento del controllo movimento).....	453
MC_Power (potenza del controllo movimento).....	457
MC_ReadActualPosition (posizione effettiva di lettura del controllo movimento).....	460

MC_ReadActualVelocity (velocità effettiva di lettura del controllo movimento).....	464
MC_ReadAxisError (errore dell'asse di lettura del controllo movimento).....	468
Codici di errore AxisErrorID	471
MC_ReadBoolParameter (parametro BOOL di lettura del controllo movimento).....	473
MC_ReadParameter (parametro di lettura del controllo movimento).....	475
MC_ReadStatus (stato di lettura del controllo movimento).....	478
MC_Reset (reset del controllo movimento)	484
MC_SetPosition (posizione impostazione del controllo movimento).....	486
MC_Stop (arresto del controllo movimento)	489
MC_TouchProbe (sonda a contatto del controllo movimento)..	492
MC_WriteBoolParameter (parametro BOOL di scrittura del controllo movimento)	496
MC_WriteParameter (parametro di scrittura del controllo movimento).....	499

Capitolo 19

Istruzioni di controllo dei processi

DERIVATE	503
FFL (caricamento FIFO).....	505
Diagrammi di temporizzazione istruzione FFL e FFU	510
FFU (Scarica FIFO)	515
HYSTER (isteresi)	520
INTEGRAL	522
AND.....	527
LFL (carico LIFO)	528
LFU (scarica LIFO).....	530
PWM (Pulse Width Modulation).....	533
SCALER (scala)	535
STACKINT (valori interi dello stack)	537
SCL	540
Esempi di diagrammi di temporizzazione dell'istruzione SCL	544
TND (interruzione programma corrente).....	547
LIMIT (prova limite)	549

Capitolo 20

Istruzione per il controllo del programma

AFI (sempre false)	553
NOP (nessuna operazione).....	553
SUS (sospensione)	554

Capitolo 21

Istruzione proporzionale integrale derivativa (PID)

Che cos'è il controllo proporzionale integrale derivativo (PID)?	557
IPIDCONTROLLER (controllore proporzionale–integrale–derivativo)	557
Tipo di dati GAIN_PID.....	561

Tipo di dati AT_Param	562
Modalità di implementazione del controllo PID da parte del blocco funzione IPIDController.....	563
Operazione del blocco funzione IPIDController	564
Utilizzo dell'istruzione proporzionale integrale derivativa	565
Utilizzo dell'auto tuning con il blocco funzione IPIDController	568
Auto tuning in sistemi di primo e secondo ordine.....	569
Configurazione dell'auto tuning.....	570
Utilizzo di un STI (Selectable Timed Interrupt) con auto tuning.....	577
Esempio: IPIDController con auto tuning.....	577
Esempio: come creare un ciclo di feedback per il valore manipolato	579
Esempio: come aggiungere un UDFB a un programma PID	580
Esempio: come creare un programma IPIDController per controllare la temperatura.....	581
Esempio: come creare un programma IPIDController per controllare il livello di erogazione dell'acqua	583
PID (proporzionale integrale derivativa)	586
Stato della macchina di istruzione PID.....	591
Diagrammi di temporizzazione dell'istruzione PID	592

Capitolo 22

Istruzioni Real Time Clock (RTC)	RTC_READ (lettura Real-Time Clock)	597
	RTC_SET (impostare Real-Time Clock)	599

Capitolo 23

Istruzioni socket	SOCKET_ACCEPT.....	604
	SOCKET_CREATE.....	607
	SOCKET_DELETE.....	612
	SOCKET_DELETEALL	615
	SOCKET_INFO	618
	SOCKET_OPEN	623
	SOCKET_READ.....	628
	SOCKET_WRITE	632
	Tipo di dati SOCKADDR_CFG	636
	Tipo di dati SOCK_STATUS.....	637
	Codici di errore del socket.....	637
	Bit di stato per l'istruzione del socket.....	641
	Diagrammi di temporizzazione dell'istruzione Socket.....	641
	Diagrammi di transazione dell'istruzione Socket	649
	Transazioni per client TCP	649
	Transazioni per server TCP.....	649
	Transazioni per UDP con connessione aperta	650
	Transazioni per UDP senza connessioni aperte.....	650
	Diagrammi di stato della macchina per TCP	651
	Diagrammi dello stato della macchina per UDP	654

	Capitolo 24	
Istruzioni di elaborazione della stringa	ASCII	657
	CHAR (codice ASCII a carattere della stringa)	659
	DELETE (eliminare una sottostringa).....	661
	FIND (trova sottostringa)	663
	INSERT (inserimento stringa).....	664
	LEFT (estrae a sinistra di una stringa)	667
	MID (estrae dal centro di una stringa)	668
	MLEN (lunghezza della stringa)	671
	RIGHT (estrazione a destra di una stringa)	672
	REPLACE (sostituisce la sottostringa).....	674
	Capitolo 25	
Istruzioni del timer	Configurazione delle istruzioni del timer	677
	TOF (timer, ritardo alla diseccitazione)	678
	TON (timer, ritardo all'eccitazione)	680
	TONOFF (ritardo, on, off)	682
	TP (tempo impulsi)	685
	RTO (timer di ritrasmissione, Ritardo all'eccitazione)	687
	DOY (anno di controllo per Real-Time clock)	689
	TDF (differenza di tempo).....	691
TOW (settimana di controllo per Real Time Clock)	693	

In questo manuale

In questa guida vengono fornite informazioni di riferimento relative all'insieme di istruzioni disponibili per lo sviluppo di programmi da utilizzare nei sistemi di controllo Micro800. L'insieme di istruzioni include il supporto dei linguaggi di programmazione ST (Structured Text), LD (Ladder Diagram) e FBD (Function Block Diagram). Inoltre, vengono definiti elementi ladder supportati nell'ambiente di sviluppo Connected Components Workbench.

Controllori supportati

Connected Components Workbench include il supporto della configurazione per i seguenti controllori Micro800.

- 2080-LC10-12AWA
- 2080-LC10-12DWD
- 2080-LC10-12QBB
- 2080-LC10-12QWB
- 2080-LC20-20AWB
- 2080-LC20-20QBB
- 2080-LC20-20QWB
- 2080-LC30-10QVB
- 2080-LC30-10QWB
- 2080-LC30-16AWB
- 2080-LC30-16QVB
- 2080-LC30-16QWB
- 2080-LC30-24QBB
- 2080-LC30-24QVB
- 2080-LC30-24QWB
- 2080-LC30-48AWB
- 2080-LC30-48QBB
- 2080-LC30-48QVB
- 2080-LC30-48QWB
- 2080-LC50-24AWB
- 2080-LC50-24QBB
- 2080-LC50-24QVB
- 2080-LC50-24QWB
- 2080-LC50-48AWB
- 2080-LC50-48QBB
- 2080-LC50-48QVB
- 2080-LC50-48QWB
- 2080-LC50-48QWB - SIM
- 2080-LC50-24AWB
- 2080-LC70-24QBB
- 2080-LC70-24QWB

Risorse aggiuntive

Questi documenti contengono informazioni aggiuntive relative a prodotti Rockwell Automation correlati.

Risorsa	Descrizione
<p>Criteria per il cablaggio e la messa a terra in automazione industriale, pubblicazione 1770-4.1 disponibile all'indirizzo http://literature.rockwellautomation.com/idc/groups/literature/documents/in/1770-in041_-en-p.pdf.</p>	<p>Fornisce criteri generali per l'installazione di un sistema industriale Rockwell Automation.</p>
<p>Sito Web delle certificazioni prodotto, http://www.ab.com</p>	<p>Fornisce dichiarazioni di conformità, certificati e altri dettagli di certificazione.</p>
<p>Guida alla migrazione dei controller MicroLogix a Micro800, disponibile all'https://literature.rockwellautomation.com/idc/groups/literature/documents/rm/2080-rm002_-en-e.pdf</p>	<p>Viene illustrato come utilizzare gli strumenti software per selezionare un controller Micro800 adatto e come convertire i programmi MicroLogix in modo che funzionino con il controller Micro800.</p>
<p>Micro800 Controllers Starter Pack Quick Start, disponibile all'https://literature.rockwellautomation.com/idc/groups/literature/documents/qs/2080-qs004_-en-e.pdf</p>	<p>Viene illustrato come utilizzare un controller Micro800 con un terminale PanelView 800.</p>
<p>Configurazione dei controller Micro800 su FactoryTalk Gateway, disponibile all'indirizzo https://literature.rockwellautomation.com/idc/groups/literature/documents/qs/2080-qs005_-en-e.pdf</p>	<p>Vengono fornite istruzioni di avvio rapido per la configurazione di un controller Micro800 in FactoryTalk Linx Gateway.</p>

Avvisi legali

È possibile visualizzare o scaricare pubblicazioni all'indirizzo <http://www.rockwellautomation.com/literature>. Per ordinare copie cartacee di documentazione tecnica, contattare il distributore Rockwell Automation locale o un rappresentante commerciale.

Rockwell Automation pubblica le note legali, quali informative sulla privacy, accordi di licenza, divulgazioni del marchio e altri termini e condizioni, nella pagina [Note legali](#) del sito Web di Rockwell Automation.

Contratto di licenza con l'utente finale (EULA)

Il contratto di licenza con l'utente finale di Rockwell Automation ("EULA") può essere consultato aprendo il file License.rtf archiviato nella cartella di installazione del prodotto sul disco rigido del PC.

La posizione predefinita del file è:

C:\Program Files (x86)\Common Files\Rockwell\license.rtf.

Licenze software open source

Il software incluso in questo prodotto contiene software con copyright concesso in licenza con una o più licenze open source.

È possibile visualizzare l'elenco completo di tutti i software open source utilizzati nel prodotto e le relative licenze, aprendo il file Indice.html archiviato nella cartella OPENSOURCE del prodotto sul disco rigido del PC.

La posizione predefinita del file è:

C:\Program Files (x86)\Common Files\Rockwell\Help\Connected Components Workbench\Release Notes\OPENSOURCE\Indice.htm

Il codice sorgente dei pacchetti open source inclusi nel prodotto è disponibile nei rispettivi siti Web del progetto. In alternativa, è possibile ottenere il codice sorgente completo contattando Rockwell Automation tramite il modulo di **Contact** disponibile sul sito Web di Rockwell Automation: <http://www.rockwellautomation.com/global/about-us/contact/contact.page>. Inserire "Open source" nel testo della richiesta.

Ricerca di informazioni sulle istruzioni e sugli elementi ladder

Connected Components Workbench include un set completo di istruzioni con strutture e vettori, ambienti di sviluppo per la logica ladder, testo strutturato, diagramma a blocchi funzionali e programmi di blocchi funzione definiti dall'utente.

Inoltre, Connected Components Workbench include strumenti di configurazione dell'interfaccia utente per i controllori Micro800, unità PowerFlex®, un dispositivo relè di sicurezza, terminali grafici del componente PanelView™ e opzioni di connettività di rete e seriali.

Per informazioni su una specifica istruzione, inclusi descrizione, dettagli dei parametri ed esempi di linguaggio, individuare l'istruzione nel sommario o in uno dei seguenti argomenti di riferimento.

- [Riferimento sulle istruzioni in ordine alfabetico](#) a pagina 18

Per una descrizione degli elementi ladder utilizzati in Connected Components Workbench, vedere la seguente sezione:

- [Riferimento linguaggio LD \(Ladder Diagram\)](#) a pagina 25

Blocchi di istruzione

Il set di istruzioni Connected Components Workbench comprende i blocchi di istruzioni conformi allo standard IEC 61131-3 per i controllori Micro800. I blocchi di istruzioni comprendono operatori, funzioni e blocchi funzione.

Vedere anche

[Operatori](#) a pagina 15

[Funzioni](#) a pagina 16

[Blocchi funzione](#) a pagina 17

Operatori

Il set di istruzioni Connected Components Workbench include i blocchi istruzione conformi a IEC 61131-3 per i controllori Micro800; gli operatori sono inclusi nel set di istruzioni Connected Components Workbench.

Un operatore è un'operazione logica di base, ad esempio aritmetica, booleana, di comparatore o conversione dati.

Vedere anche

[Blocchi funzione](#) a pagina 17

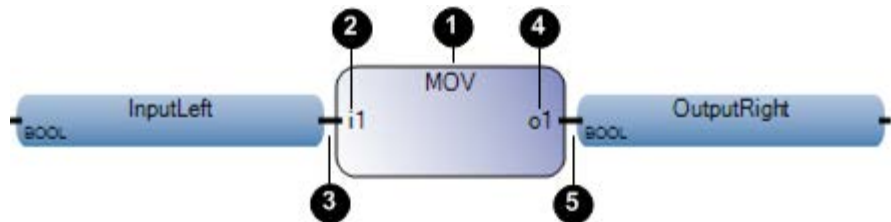
[Funzioni](#) a pagina 16

Funzioni

Le funzioni hanno uno o più parametri di ingresso e uno o più parametri di uscita.

Formato del blocco di istruzioni

Un blocco di istruzioni è rappresentato da un singolo rettangolo e ha un numero fisso di punti di connessione di ingresso e di uscita. Un blocco di istruzioni elementare esegue una sola funzione.

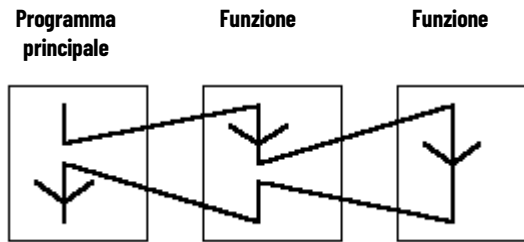


Elemento num.	Elemento	Descrizione
1	Nome blocco	Il nome della funzione da eseguire dal blocco di istruzione è scritto all'interno del rettangolo (in alto).
2	Ingresso	Ogni ingresso di un blocco di istruzioni è etichettato e possiede un tipo predefinito.
3	Connessione di ingresso	Gli ingressi sono connessi sul bordo sinistro.
4	Uscita	Ogni uscita di un blocco di istruzioni è etichettata e possiede un tipo predefinito.
5	Connessione di uscita	Le uscite sono connesse sul bordo destro.

Chiamata di una funzione

Connected Components Workbench™ non supporta chiamate a funzioni ricorsive. Quando una funzione della sezione Funzioni viene chiamata da se stessa o da una delle sue funzioni chiamate, si verifica un errore di runtime. Inoltre, le funzioni non archiviano i valori locali delle proprie variabili locali. Dato che le funzioni non sono oggettivate, non possono richiamare blocchi funzione.

- Una funzione può essere richiamata da un programma, una funzione o un blocco funzione.
- Qualsiasi programma di qualsiasi sezione può chiamare una o più funzioni. Una funzione può avere variabili locali.
- Una funzione non ha nessun significato di istanza, ovvero i dati locali non sono archiviati e di norma andranno persi da una chiamata all'altra.
- Poiché l'esecuzione di una funzione è controllata dal relativo programma principale, l'esecuzione di quest'ultimo è sospesa fino al termine della funzione.



Definizione dei nomi di parametri e funzioni

L'interfaccia di una funzione deve essere definita esplicitamente con tipo e nome univoco per ciascuno dei parametri chiamanti (ingresso) o dei parametri di ritorno (uscita). Una funzione ha un solo parametri di ritorno. Il valore del parametro di ritorno di un blocco funzione varia a seconda del linguaggio di programmazione (FBD, LD, ST).

I nomi delle funzioni e dei relativi parametri possono contenere un massimo di 128 caratteri. I nomi del parametro della funzione può iniziare con una lettera o un carattere di sottolineatura, seguiti da lettere, cifre e singoli caratteri di sottolineatura.

Vedere anche

[Blocchi funzione](#) a pagina 17

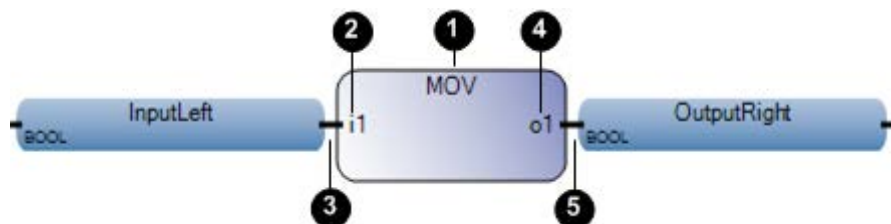
[Operatori](#) a pagina 15

Blocchi funzione

Un blocco funzione è un blocco di istruzioni con parametri di ingresso e uscita e che funziona con dati interni (parametri). Può essere scritto nei linguaggi testo strutturato, ladder diagram e diagramma a blocchi funzionali.

Formato del blocco di istruzioni

Un blocco di istruzioni è rappresentato da un singolo rettangolo e ha un numero fisso di punti di connessione di ingresso e di uscita. Un blocco di istruzioni elementare esegue una sola funzione.



Elemento num.	Elemento	Descrizione
---------------	----------	-------------

1	Nome blocco	Il nome della funzione da eseguire dal blocco di istruzione è scritto all'interno del rettangolo (in alto).
2	Ingresso	Ogni ingresso di un blocco di istruzioni è etichettato e possiede un tipo predefinito.
3	Connessione di ingresso	Gli ingressi sono connessi sul bordo sinistro.
4	Uscita	Ogni uscita di un blocco di istruzioni è etichettata e possiede un tipo predefinito.
5	Connessione di uscita	Le uscite sono connesse sul bordo destro.

Chiamata di un blocco funzione

Quando in un programma viene chiamato un blocco funzione, in realtà viene chiamata un'istanza di esso. L'istanza utilizza lo stesso codice, ma viene creata un'istanza dei parametri di ingresso e uscita, ovvero per ogni istanza del blocco funzione vengono copiate le variabili locali. I valori delle variabili dell'istanza di un blocco funzione sono archiviati da un ciclo all'altro.

Un blocco funzione può essere chiamato da un programma o da un altro blocco funzione, mentre non può essere chiamato dalle funzioni, in quanto di queste non viene creata un'istanza.

Definizione dei nomi di parametri e blocchi funzione

L'interfaccia di un blocco funzione deve essere definita esplicitamente con tipo e un nome unico per ciascuno dei parametri chiamanti (ingresso) o dei parametri di ritorno (uscita). I blocchi funzione possono avere più di un parametro di uscita. Il valore del parametro di ritorno di un blocco funzione varia a seconda del linguaggio di programmazione (FBD, LD, ST).

I nomi dei blocchi funzione e dei relativi parametri possono contenere un massimo di 128 caratteri. I nomi del parametro del blocco funzione può iniziare con una lettera o un carattere di sottolineatura, seguiti da lettere, numeri e singoli caratteri di sottolineatura.

Vedere anche

[Funzioni](#) a pagina 16

[Operatori](#) a pagina 15

Istruzione impostate in ordine alfabetico

La tabella seguente elenca le istruzioni per il controllore Micro800 disponibili in Connected Components Workbench e le relative istruzioni di mappatura nel tema Logix, in ordine alfabetico.

Istruzione	Istruzione mappata (tema Logix)	Categoria	Tipo	Descrizione
- a pagina 103	SUB	Aritmetico	Operatore	Sottrarre un valore Integer, Real o Time da un altro valore Integer, Real o Time.

Istruzione	Istruzione mappata (tema Logix)	Categoria	Tipo	Descrizione
* a pagina 92	MUL	Aritmetico	Operatore	Moltiplica due o più valori Integer o Real.
/ a pagina 84	DIV	Aritmetico	Operatore	Divisione di due valori Integer o Real.
+ a pagina 72	ADD	Aritmetico	Operatore	Somma due o più valori Integer, Real, Time o String.
≤ a pagina 249	LES	Confronta	Operatore	Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo è inferiore al secondo.
≤= a pagina 250	LEQ	Confronta	Operatore	Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo è inferiore o uguale al secondo.
≥ a pagina 251	NEQ	Confronta	Operatore	Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo non è uguale al secondo.
≡ a pagina 243	EQU	Confronta	Operatore	Testa se un valore è uguale a un altro.
≥ a pagina 246	GRT	Confronta	Operatore	Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo è maggiore del secondo.
≥= a pagina 247	GEQ	Confronta	Operatore	Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo è maggiore o uguale al secondo.
ABL a pagina 111	ABL	Comunicazioni	Blocco funzione	Conta il numero di caratteri nel buffer fino al carattere di fine riga incluso.
ABS a pagina 67	ABS	Aritmetico	Funzione	Restituisce il valore assoluto di un valore Real.
ACB a pagina 118	ACB	Comunicazioni	Blocco funzione	Conta il numero totale dei caratteri nel buffer.
ACL a pagina 113	ACL	Comunicazioni	Blocco funzione	Elimina i buffer di ricezione e trasmissione.
ACOS a pagina 69	ACS	Aritmetico	Funzione	Calcola l'arcocoseno di un valore Real.
ACOS_LREAL a pagina 71	ACOS_LREAL	Aritmetico	Funzione	Calcola l'arcocoseno di un valore Long Real.
AFI a pagina 553	AFI	Controllo programma	Funzione	Disabilita un ramo temporaneamente quando si esegue il debug.
AHL a pagina 115	AHL	Comunicazioni	Blocco funzione	Impostare o reimpostare linee di arbitraggio modem.
AND a pagina 165	AND	Operazioni booleane	Operatore	Esegue un'operazione booleana AND tra due o più valori.
AND_MASK a pagina 135	AND_MASK	Operazioni binarie	Funzione	Esegue AND bit-to-bit tra due valori Integer.
ANY_TO_BOOL a pagina 263	ANY_TO_BOOL	Conversione dati	Funzione	Converte un valore non booleano in uno booleano.
ANY_TO_BYTE a pagina 264	ANY_TO_BYTE	Conversione dati	Funzione	Converte un valore in un byte.
ANY_TO_DATE a pagina 265	ANY_TO_DATE	Conversione dati	Funzione	Converte un tipo di dati String, Integer, Real o Time in un tipo di dati Date.
ANY_TO_DINT a pagina 267	STOD	Conversione dati	Funzione	Converte un valore in Double Integer.
ANY_TO_DWORD a pagina 268	ANY_TO_DWORD	Conversione dati	Funzione	Converte un valore in un valore Double Word.
ANY_TO_INT a pagina 269	ACI	Conversione dati	Funzione	Converte un valore in Integer.
ANY_TO_LINT a pagina 270	ANY_TO_LINT	Conversione dati	Funzione	Converte un valore in Long Integer.
ANY_TO_LREAL a pagina 272	ANY_TO_LREAL	Conversione dati	Funzione	Converte un valore in Long Real.
ANY_TO_LWORD a pagina 273	ANY_TO_LWORD	Conversione dati	Funzione	Converte un valore in Long Word.
ANY_TO_REAL a pagina 274	STOR	Conversione dati	Funzione	Converte un valore in Real.
ANY_TO_SINT a pagina 275	ANY_TO_SINT	Conversione dati	Funzione	Converte un valore in Short Integer.
ANY_TO_STRING a pagina 276	DTOS	Conversione dati	Funzione	Converte un valore in String.
ANY_TO_TIME a pagina 277	ANY_TO_TIME	Conversione dati	Funzione	Converte un valore in un tipo di dati Time.
ANY_TO_UDINT a pagina 278	ANY_TO_UDINT	Conversione dati	Funzione	Converte un valore in Unsigned Double Integer.
ANY_TO_UINT a pagina 279	ANY_TO_UINT	Conversione dati	Funzione	Converte un valore in Unsigned Integer.
ANY_TO_ULINT a pagina 280	ANY_TO_ULINT	Conversione dati	Funzione	Converte un valore in Unsigned Long Integer.
ANY_TO_USINT a pagina 281	ANY_TO_USINT	Conversione dati	Funzione	Converte un valore in Unsigned Short Integer.
ANY_TO_WORD a pagina 282	ANY_TO_WORD	Conversione dati	Funzione	Converte un valore in Word.
ARD a pagina 120	ARD	Comunicazioni	Blocco funzione	Legge i caratteri dal buffer di ingresso e posiziona gli stessi in una stringa.

Capitolo 1 Ricerca di informazioni sulle istruzioni e sugli elementi ladder

Istruzione	Istruzione mappata (tema Logix)	Categoria	Tipo	Descrizione
ARL a pagina 123	ARL	Comunicazioni	Blocco funzione	Legge una riga di caratteri dal buffer di ingresso e posiziona gli stessi in una stringa.
ASCII a pagina 657	ASCII	Manipolazione stringa	Funzione	Restituire il codice ASCII per i caratteri in una stringa. Carattere -> codice ASCII.
ASIN a pagina 74	ASN	Aritmetico	Funzione	Calcola l'arcoseno di un valore Real.
ASIN_LREAL a pagina 76	ASN_LREAL	Aritmetico	Funzione	Calcola l'arcoseno di un valore Long Real.
ATAN a pagina 77	ATN	Aritmetico	Funzione	Calcola l'arcotangente di un valore Real.
ATAN_LREAL a pagina 79	ATAN_LREAL	Aritmetico	Funzione	Calcola l'arcotangente di un valore Long Real.
MEDIA a pagina 285	AVE	Manipolazione dati	Blocco funzione	Calcola una media in esecuzione su un numero di campioni definito.
AWA a pagina 125	AWA	Comunicazioni	Funzione	Scrivere una stringa con due caratteri aggiunti (configurati dall'utente) su un dispositivo esterno.
AWT a pagina 128	AWT	Comunicazioni	Funzione	Scriva i caratteri da una stringa sorgente a un dispositivo esterno.
BSL a pagina 137	BSL	Operazioni binarie	Blocco funzione	Sposta un bit in un elemento vettore verso sinistra.
BSR a pagina 141	BSR	Operazioni binarie	Blocco funzione	Sposta un bit in un elemento vettore verso destra.
CHAR a pagina 659	CHAR	Manipolazione stringa	Funzione	Restituisce una stringa di un carattere per un codice ASCII. Codice ASCII -> Carattere.
COM_IO_WDOG a pagina 179	COM_IO_WDOG	Comunicazioni	Blocco funzione	Monitora la comunicazione con il controllore.
COP a pagina 287	COP	Conversione dati	Blocco funzione	Copia i dati binari nell'elemento di origine per l'elemento di destinazione.
COS a pagina 81	COS	Aritmetico	Funzione	Calcola il coseno di un valore Real.
COS_LREAL a pagina 82	COS_LREAL		Funzione	Calcola il coseno di un valore Long Real.
CTD a pagina 255	CTD	Contatore	Funzione	Conta gli Integer da un valore dato fino a 0, 1 a 1.
CTU a pagina 257	CTU	Contatore	Funzione	Conta gli Integer da 0 fino a un valore dato, 1 a 1.
CTUD a pagina 259	CTUD	Contatore	Funzione	Conta gli Integer da 0 fino a un valore dato, 1 a 1, o da un valore dato a 0, 1 a 1.
DELETE a pagina 661	DELETE	Manipolazione stringa	Funzione	Elimina i caratteri da una stringa.
DERIVATE a pagina 503	DERIVATE	Controllo del processo	Blocco funzione	Differenziazione di un valore reale su una durata del ciclo definito.
DLG a pagina 366	DLG	Ingresso/uscita	Blocco funzione	Scriva i valori di variabile dal motore di runtime in un file di registrazione dati su una scheda SD.
DOY a pagina 689	DOY	Durata	Funzione	Attiva un output se il valore del real-time clock è compreso nell'intervallo dell'impostazione Ora dell'anno.
EXPT a pagina 86	EXPT	Aritmetico	Funzione	Calcola il valore Real di un numero di base elevato alla potenza dell'esponente intero.
F_TRIG a pagina 159	OSF	Operazioni booleane	Blocco funzione	Rileva un fronte di discesa di una variabile booleana.
FFL a pagina 515	FFL	Controllo del processo	Blocco funzione	Carica i dati a 8 bit, 16 bit, 32 bit o 64 bit in un vettore creato dall'utente denominato stack FIFO.
FFU a pagina 515	FFU	Controllo del processo	Blocco funzione	Scarica i dati a 8, 16, 32 e 64 bit da un vettore creato dall'utente denominato stack FIFO (First In First Out, primo a entrare primo a uscire) nello stesso ordine in cui i dati sono stati caricati usando l'istruzione FFL.
FIND a pagina 663	FIND	Manipolazione stringa	Funzione	individua e fornisce la posizione delle sottostringhe nelle stringhe.
HSC a pagina 298	HSC	Ingresso/uscita	Blocco funzione	HSC applica all'high-speed counter valori alti e bassi preimpostati e valori di origine per l'uscita.
HSC_SET_STS a pagina 314	HSC_SET_STS	Ingresso/uscita	Blocco funzione	HSC_SET_STS permette l'impostazione o il ripristino manuale dello stato del conteggio dell'HSC.
HSCE a pagina 337	HSCE	Ingresso/uscita	Blocco funzione	HSCE: avvio, arresto e lettura del valore dell'accumulatore.
HSCE_CFG a pagina 341	HSCE_CFG	Ingresso/uscita	Blocco funzione	HSCE_CFG è la configurazione dell'High Speed Counter.
HSCE_CFG_PLS a pagina 344	HSCE_CFG_PLS	Ingresso/uscita	Blocco funzione	HSCE_CFG_PLS è la configurazione dell'interruttore di fine corsa programmabile (PLS) dell'High Speed Counter.
HSCE_READ_STS a pagina 346	HSCE_READ_STS	Ingresso/uscita	Blocco funzione	HSCE_READ_STS legge lo stato dell'High Speed Counter.

Istruzione	Istruzione mappata (tema Logix)	Categoria	Tipo	Descrizione
HSCE_SET_STS a pagina 348	HSCE_SET_STS	Ingresso/uscita	Blocco funzione	HSCE_SET_STS imposta/esegue il reset manuale dello stato dell'High Speed Counter.
HYSTER a pagina 520	HYSTER	Controllo del processo	Blocco funzione	Isteresi booleana sulla differenza tra numeri reali.
IIM a pagina 368	IIM	Ingresso/uscita	Blocco funzione	Aggiorna gli ingressi prima della normale scansione in uscita.
INSERT a pagina 664	INSERT	Manipolazione stringa	Funzione	inserisce sottostringhe in posizioni definite dall'utente all'interno delle stringhe.
INTEGRAL a pagina 522	INTEGRAL	Controllo del processo	Blocco funzione	Integra un valore real durante la durata del ciclo definito.
IOM a pagina 370	IOM	Ingresso/uscita	Blocco funzione	Aggiorna le uscite prima della normale in uscita.
IPIDCONTROLLER a pagina 557	IPIDCONTROLLER	Controllo del processo	Blocco funzione	Configurare e controllare gli ingressi e le uscite utilizzate per la logica proporzionale-integrale-derivativa (PID).
KEY_READ a pagina 372		Ingresso/uscita	Blocco funzione	Solo Micro810. Legge lo stato della chiave su un modulo LCD opzionale quando il display dell'utente è attivo.
KEY_READ_REM a pagina 375	KEY_READ_REM	Ingresso/uscita	Blocco funzione	Solo Micro820. Legge lo stato della chiave su un modulo LCD remoto opzionale quando il display dell'utente è attivo.
LCD a pagina 356		Ingresso/uscita	Funzione	Solo Micro810. Visualizza una stringa o un numero su una schermata LCD.
LCD_BKLT_REM a pagina 358	LCD_BKLT_REM	Ingresso/uscita	Funzione	Imposta i parametri di retroilluminazione dell'LCD remoto all'interno del programma utente.
LCD_REM a pagina 360	LCD_REM	Ingresso/uscita	Funzione	Visualizza i messaggi definiti dall'utente per l'LCD remoto.
LEFT a pagina 667	LEFT	Manipolazione stringa	Funzione	Estrae i caratteri dal lato sinistro di una stringa.
LFL (carico LIFO) a pagina 528	LFL	Controllo del processo	Blocco funzione	Carica i dati a 8 bit, 16 bit, 32 bit o 64 bit in un vettore creato dall'utente denominato stack LIFO.
LFU (scarica LIFO) a pagina 530	LFU	Controllo del processo	Blocco funzione	Scarica i dati a 8 bit, 16 bit, 32 bit e 64 bit da un vettore creato dall'utente denominato stack LIFO (Last In First Out, ultimo a entrare primo a uscire) nello stesso ordine in cui i dati sono stati caricati usando l'istruzione LFL.
LIM_ALARM a pagina 63	LIM	Allarme	Blocco funzione	Un allarme con isteresi di un valore Real per i limiti superiore e inferiore.
LIMIT a pagina 549	LIMIT	Controllo del processo	Funzione	riduce i valori interi a un intervallo definito.
LOG a pagina 87	LOG	Aritmetico	Funzione	Calcola il logaritmo (base 10) di un valore Real.
MAX a pagina 294	MAX	Manipolazione dati	Funzione	Calcola il valore massimo tra due valori interi.
MC_AbortTrigger a pagina 434	MC_AbortTrigger	Controllo assi	Blocco funzione	Interrompe i blocchi funzione Controllo assi collegati agli eventi di attivazione.
MC_Halt a pagina 437	MC_Halt	Controllo assi	Blocco funzione	Comanda un arresto controllato del movimento in condizioni operative normali.
MC_Home a pagina 440	MAH	Controllo assi	Blocco funzione	Comanda l'asse per l'esecuzione della sequenza <search home>.
MC_MoveAbsolute a pagina 445	MAM	Controllo assi	Blocco funzione	Determina un movimento controllato verso una posizione assoluta specificata.
MC_MoveRelative a pagina 448	MC_MoveRelative	Controllo assi	Blocco funzione	Determina un movimento controllato di una distanza specificata rispetto alla posizione attuale nel momento dell'esecuzione.
MC_MoveVelocity a pagina 453	MCD	Controllo assi	Blocco funzione	Determina un movimento controllato infinito a una velocità specificata.
MC_Power a pagina 457	MSO	Controllo assi	Blocco funzione	Controllo dello stato di alimentazione (ON o OFF).
MC_ReadActualPosition a pagina 464	MC_ReadActualPosition	Controllo assi	Blocco funzione	Restituisce la posizione effettiva dell'asse di feedback.
MC_ReadActualVelocity a pagina 464	MC_ReadActualVelocity	Controllo assi	Blocco funzione	Ritorna alla velocità corrente dell'asse di feedback.
MC_ReadAxisError a pagina 468	MC_ReadAxisError	Controllo assi	Blocco funzione	Legge gli errori dell'asse non in relazione ai blocchi funzione Controllo movimento.
MC_ReadBoolParameter a pagina 473	MC_ReadBoolParameter	Controllo assi	Blocco funzione	Restituisce il valore di un parametro specifico del produttore di tipo BOOL.
MC_ReadParameter a pagina 475	MC_ReadParameter	Controllo assi	Blocco funzione	Restituisce il valore di un parametro specifico del fornitore di tipo Real.

Capitolo 1 Ricerca di informazioni sulle istruzioni e sugli elementi ladder

Istruzione	Istruzione mappata (tema Logix)	Categoria	Tipo	Descrizione
MC_ReadStatus a pagina 478	MC_ReadStatus	Controllo assi	Blocco funzione	Restituisce lo stato dell'asse rispetto al movimento attualmente in corso.
MC_Reset a pagina 484	MAFR	Controllo assi	Blocco funzione	Trasforma lo stato dell'asse da ErrorStop a StandStill ripristinando tutti gli errori interni legati all'asse.
MC_SetPosition a pagina 486	MRP	Controllo assi	Blocco funzione	Sposta il sistema delle coordinate di un asse tramite la manipolazione della posizione effettiva.
MC_Stop a pagina 489	MAS	Controllo assi	Blocco funzione	Comanda un arresto di movimento controllato e trasferisce lo stato dell'asse su Arresto in corso.
MC_TouchProbe a pagina 492	MC_TouchProbe	Controllo assi	Blocco funzione	Registra una posizione dell'asse in corrispondenza di un evento di attivazione.
MC_WriteBoolParameter a pagina 496	MC_WriteBoolParameter	Controllo assi	Blocco funzione	Modifica il valore di un parametro specifico del fornitore di tipo BOOL.
MC_WriteParameter a pagina 499	MC_WriteParameter	Controllo assi	Blocco funzione	Modifica il valore di un parametro specifico del fornitore di tipo Real.
MID a pagina 668	MID	Manipolazione stringa	Funzione	Estrae i caratteri dal centro di una stringa.
MIN a pagina 292	MIN	Manipolazione dati	Funzione	Calcola il valore minimo tra due valori interi.
MLEN a pagina 671	MLEN	Manipolazione stringa	Funzione	Calcola la lunghezza di una stringa.
MM_INFO a pagina 377	MM_INFO	Ingresso/uscita	Blocco funzione	Legge le informazioni sull'intestazione del modulo di memoria.
MOD a pagina 89	MOD	Aritmetico	Funzione	Esegue il calcolo Modulo sui valori Integer.
MODULE_INFO a pagina 380	MODULE_INFO	Ingresso/uscita	Blocco funzione	Legge le informazioni di modulo da un modulo plug-in o di espansione.
MOV a pagina 91	MOV	Aritmetico	Operatore	Assegna il valore di ingresso in un'uscita.
MSG_CIPGENERIC a pagina 182	MSG	Comunicazioni	Funzione	Invia un messaggio CIP generico esplicito.
MSG_CIPSYMBOLIC a pagina 191	MSG_CIPSYMBOLIC	Comunicazioni	Funzione	Invia un messaggio CIP Symbolic esplicito.
MSG_MODBUS a pagina 196	MSG_MODBUS	Comunicazioni	Funzione	Invia un messaggio Modbus.
MSG_MODBUS2 a pagina 202	MSG_MODBUS2	Comunicazioni	Funzione	Invia un messaggio MODBUS/TCP su un canale Ethernet.
MUX4B a pagina 175	MUX4B	Booleano	Funzione	Multiplexer tra quattro ingressi BOOL, restituisce un valore BOOL.
MUX8B a pagina 172	MUX8B	Booleano	Funzione	Multiplexer tra otto ingressi BOOL, restituisce un valore BOOL.
Neg a pagina 94	NEG	Aritmetico	Operatore	Converte un valore in un valore negativo.
NOP a pagina 553	NOP	Controllo del programma	Funzione	Funziona come segnaposto.
NOT a pagina 166	NOT	Operazioni booleane	Operatore	Converte i valori booleani in valori negati.
NOT_MASK a pagina 144	NOT_MASK	Operazioni binarie	Funzione	Maschera negazione numero intero bit-to-bit, inverte il valore di un parametro.
OR a pagina 164	OR	Operazioni booleane	Operatore	OR booleano di due o più valori.
OR_MASK a pagina 146	OR_MASK	Operazioni binarie	Funzione	Maschera OR numero intero bit-to-bit, abilita i bit.
PID a pagina 586	PID	Controllo del processo	Blocco funzione	Un'istruzione di uscita che controlla le proprietà fisiche come la temperatura, la pressione, il livello dei liquidi o la portata tramite cicli di processo.
PLUGIN_INFO a pagina 392	PLUGIN_INFO	Ingresso/uscita	Blocco funzione	Acquisisce le informazioni del modulo da un modulo plug-in generico (escluso il Modulo di memoria).
PLUGIN_READ a pagina 394	PLUGIN_READ	Ingresso/uscita	Blocco funzione	Legge dati da un modulo plug-in generico (escluso il Modulo di memoria).
PLUGIN_RESET a pagina 397	PLUGIN_RESET	Ingresso/uscita	Blocco funzione	Ripristina un modulo plug-in generico, ripristino hardware (ad esclusione di Modulo di memoria).
PLUGIN_WRITE a pagina 398	PLUGIN_WRITE	Ingresso/uscita	Blocco funzione	Scriva dati per un modulo plug-in generico (escluso il Modulo di memoria).
POW a pagina 95	XPY	Aritmetico	Funzione	Calcola il valore di un numero Real elevato alla potenza dell'esponente Real.
PWM a pagina 533	PWM	Controllo del processo	Blocco funzione	Attiva o disattiva l'uscita del PMW (Pulse Width Modulation) per un canale PWM configurato.
R_TRIG a pagina 161	OSR	Operazioni booleane	Blocco funzione	Rileva un fronte di salita di una variabile booleana.
RAND a pagina 97	RAND	Aritmetico	Funzione	Calcola valori interi casuali da un intervallo definito.
RCP a pagina 400	RCP	Ingresso/uscita	Blocco funzione	Legge e scrive dati Ricetta su e da una scheda di memoria SD.

Istruzione	Istruzione mappata (tema Logix)	Categoria	Tipo	Descrizione
REPLACE a pagina 674	REPLACE	Manipolazione stringa	Funzione	Sostituisce le parti di una stringa con nuovi set di caratteri.
RHC a pagina 364	RHC	Ingresso/uscita	Funzione	Legge high-speed clock.
RIGHT a pagina 672	RIGHT	Manipolazione stringa	Funzione	Estrae i caratteri dal lato destro di una stringa.
ROL a pagina 147	ROL	Operazioni binarie	Funzione	Per i numeri Integer a 32 bit, ruota i bit interi a sinistra.
ROR a pagina 149	ROR	Operazioni binarie	Funzione	Per i numeri interi a 32 bit, ruota i bit dell'intero a destra.
RPC a pagina 365	RPC	Ingresso/uscita	Funzione	Legge il checksum del programma utente.
RS a pagina 163	RS	Operazioni booleane	Blocco funzione	Ripristino bistabile dominante.
RTC_READ a pagina 403	RTC_READ	Ingresso/uscita	Blocco funzione	Legge le informazioni sul modulo real-time clock (RTC).
RTC_SET a pagina 405	RTC_SET	Ingresso/uscita	Blocco funzione	Imposta dati RTC (Real-Time clock) per le informazioni sul modulo RTC.
RTO a pagina 687	RTO	Durata	Blocco funzione	Tempo di ritrasmissione. Incrementa un timer interno quando l'ingresso è attivo, ma non ripristina il timer interno quando l'ingresso diventa inattivo.
SCALER a pagina 535	SCP	Controllo del processo	Blocco funzione	Scala il valore di ingresso in base all'intervallo di uscita.
SCL a pagina 540	SCL	Controllo del processo	Blocco funzione	Converte un valore di ingresso non scalato in un valore in virgola mobile espresso in unità ingegneristiche.
SHL a pagina 151	SHL	Operazioni binarie	Funzione	Per gli integer a 32-bit, sposta gli integer a sinistra e inserisce 0 nel bit meno significativo.
SHR a pagina 153	SHR	Operazioni binarie	Funzione	Per gli Integer a 32 bit, sposta gli Integer a sinistra e colloca 0 nel bit più significativo.
SIN a pagina 99	SIN	Aritmetico	Funzione	Calcola il seno di un valore Real.
SIN_LREAL a pagina 100	SIN_LREAL	Aritmetico	Funzione	Calcola il seno di un valore Long Real.
SOCKET_ACCEPT a pagina 604	SOCKET_ACCEPT	Comunicazioni	Blocco funzione	Accetta una richiesta di connessione TCP da una destinazione remota e restituisce un'istanza di socket utilizzata per inviare e ricevere dati sulla connessione appena creata.
SOCKET_CREATE a pagina 607	SOCKET_CREATE	Comunicazioni	Blocco funzione	Crea un'istanza del socket e restituisce un numero di istanza che viene usato come ingresso nelle operazioni socket successive.
SOCKET_DELETE a pagina 612	SOCKET_DELETE	Comunicazioni	Blocco funzione	Elimina un'istanza di socket creata.
SOCKET_DELETEALL a pagina 615	SOCKET_DELETEALL	Comunicazioni	Blocco funzione	Elimina tutte le istanze socket create.
SOCKET_INFO a pagina 618	SOCKET_INFO	Comunicazioni	Blocco funzione	Restituisce le informazioni per il socket come codici di errore e lo stato di esecuzione.
SOCKET_OPEN a pagina 623	SOCKET_OPEN	Comunicazioni	Blocco funzione	Apri la connessione per l'indirizzo di destinazione specificato per le connessioni Transmission Control Protocol (TCP). Per le connessioni User Datagram Protocol (UDP), associa un indirizzo IP di destinazione e il numero di porta al socket specificato.
SOCKET_READ a pagina 628	SOCKET_READ	Comunicazioni	Blocco funzione	Legge i dati su un socket.
SOCKET_WRITE a pagina 632	SOCKET_WRITE	Comunicazioni	Blocco funzione	Invia dati su un socket.
SORT a pagina 102	SQR	Aritmetico	Funzione	Calcola la radice quadrata di un valore Real.
SR a pagina 167	SR	Operazioni booleane	Blocco funzione	Impostazione bistabile dominante.
STACKINT a pagina 537	STACKINT	Controllo del processo	Blocco funzione	Gestisce lo stack di valori interi.
STIS a pagina 413	STS	Interrupt	Funzione	Avvia il timer Selected Timed Interrupt (STI) utente dal programma di controllo invece che con l'avvio automatico.
SUS a pagina 554	SUS	Controllo del programma	Blocco funzione	Sospende l'esecuzione del <controllore M800>.
SYS_INFO a pagina 407	SYS_INFO	Ingresso/uscita	Blocco funzione	Legge il blocco dati dello stato per il controllore Micro800.
TAN a pagina 105	TAN	Aritmetico	Funzione	Calcola la tangente di un valore Real.
TAN_LREAL a pagina 107	TAN_LREAL	Aritmetico	Funzione	Calcola la tangente di un valore Long Real.
TDF a pagina 691	TDF	Durata	Funzione	Calcola la differenza di tempo tra TimeA e TimeB.
TND a pagina 547	TND	Controllo del processo	Funzione	Arresta il ciclo di scansione del programma utente in corso.
TOF a pagina 678	TOF	Durata	Blocco funzione	Tempistica ritardo spegnimento. Incrementa un timer interno fino a un valore dato.
TON a pagina 680	TON	Durata	Blocco funzione	Tempistica ritardo accensione. Incrementa un timer interno fino a un valore dato.
TONOFF a pagina 682	TONOFF	Durata	Blocco funzione	Ritardo di accensione di un'uscita in caso di ramo True, quindi ritardo di spegnimento di un'uscita in caso di ramo False.

Istruzione	Istruzione mappata (tema Logix)	Categoria	Tipo	Descrizione
TOW a pagina 693	TOW	Durata	Funzione	Attiva un'uscita se il valore del real-time clock è compreso nell'intervallo dell'impostazione Ora della settimana.
TP a pagina 685	TP	Durata	Blocco funzione	Tempistica impulsi. Sul fronte di salita, incrementa un timer interno fino a un valore dato.
TRIMPOT_READ a pagina 410	TRIMPOT_READ	Ingresso/uscita	Blocco funzione	Legge il valore TrimPot da un trimpot specifico.
TRUNC a pagina 108	TRN	Aritmetico	Funzione	Tronca valori Real, lasciando solo l'intero.
TTABLE a pagina 169	TTABLE	Booleano	Funzione	Fornisce il valore di uscita in base alla combinazione degli ingressi.
UIC a pagina 415	UIC	Interrupt	Funzione	Annulla la perdita di bit per l'interrupt utente selezionato.
UID a pagina 416	UID	Interrupt	Funzione	Disabilita un interrupt utente specifico.
UIE a pagina 418	UIE	Interrupt	Funzione	Abilita un interrupt utente specifico.
UIF a pagina 420	UIF	Interrupt	Funzione	Scarica o rimuove un ingresso utente in sospeso.
XOR a pagina 166	XOR	Operazioni booleane	Operatore	OR esclusivo booleano di due o più valori.
XOR_MASK a pagina 155	XOR_MASK	Operazioni binarie	Funzione	Maschera OR intero esclusivo bit-to-bit, restituisce valori di bit invertiti.

Vedere anche

[Ricerca di informazioni sulle istruzioni e gli elementi ladder](#) a pagina 15

Linguaggio LD (Ladder Diagram)

Un ladder diagram (LD) è una rappresentazione grafica di equazioni Booleana che combinano i contatti (argomenti ingresso) con le bobine (risultati uscita). Utilizzando simboli grafici su un diagramma di programma (organizzato come ladder diagram di cablaggio relè), il linguaggio LD descrive prove e modifiche dei dati Booleana.

Nel diagramma i simboli grafici LD sono organizzati come un diagramma dei contatti elettrici. Il termine "ladder" (scala) deriva dal concetto di pioli collegati a linee di alimentazione verticali su entrambe le estremità, in cui ciascun piolo rappresenta un singolo circuito.

In Connected Components Workbench è incluso un editor di lingue LD e sono supportati solo gli elementi e le istruzioni forniti con il software Connected Components Workbench.

Vedere anche

[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Ambiente di sviluppo del programma LD](#) a pagina 26

[Utilizzo degli elementi nell'editor di lingue LD](#) a pagina 56

Programmazione in Ladder Diagram

Un ladder diagram (LD) è una rappresentazione grafica di equazioni Booleana che combinano i contatti (argomenti ingresso) con le bobine (risultati uscita). Utilizzando simboli grafici su un diagramma di programma (organizzato come ladder diagram di cablaggio relè), il linguaggio LD descrive prove e modifiche dei dati Booleana.

Il linguaggio LD utilizza simboli grafici su un diagramma di programma, organizzato come diagramma ladder del cablaggio del relè, per descrivere le prove e le modifiche dei dati Booleani.

Connected Components Workbench fornisce un editor di lingue Diagramma ladder e supporta solo gli elementi e le istruzioni forniti con il software Connected Components Workbench.

Vedere anche

[Ambiente di sviluppo del programma LD](#) a pagina 26

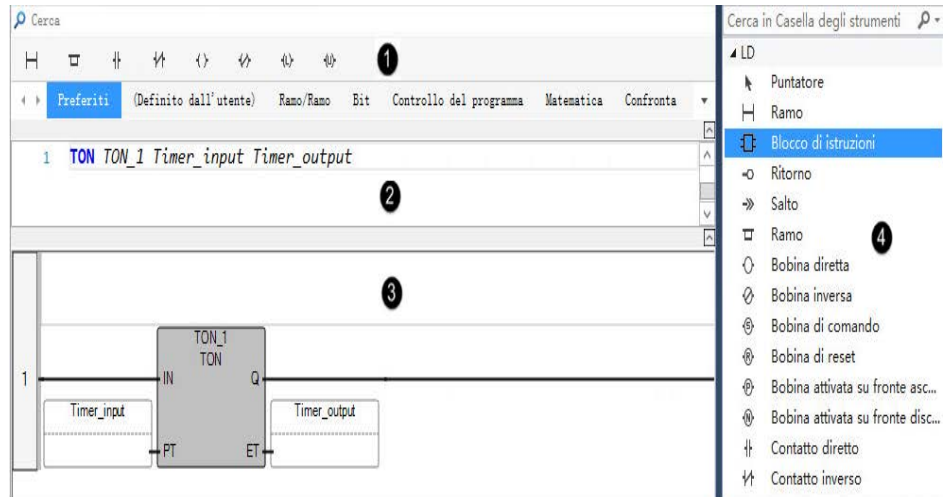
[Utilizzo degli elementi nell'editor di lingue LD](#) a pagina 56

[Elementi del ladder diagram \(LD\)](#) a pagina 29

Ambiente di sviluppo del programma LD

L'editor di lingue per un programma Diagramma ladder in cui si sviluppa una Program Organization Unit (POU).

Nella figura seguente vengono mostrate le aree principali dell'ambiente di sviluppo di programmi LD.



N.	Nome	Descrizione
1	Barra degli strumenti Istruzione	Selezionare rapidamente un elemento istruzione e collocarlo nell'editor grafico LD o fare clic per aggiungere nell'editor di testo LD.
2	Editor di testo LD	Modifica la logica mediante elementi mnemonici delle istruzioni ASCII.
3	Editor grafico LD	Modifica la logica mediante elementi delle istruzioni grafici.
4	Casella degli strumenti LD	Aggiunge elementi all'editor grafico LD.

Vedere anche

[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Barra degli strumenti Istruzione](#) a pagina 26

[Esempi di programma ladder diagram \(LD\)](#) a pagina 57

[Tasti di scelta rapida LD](#) a pagina 59

Barra degli strumenti Istruzione (LD)

Barra degli strumenti Istruzione è il nome colloquiale per il riquadro secondario nel riquadro editor di lingue, che funziona come una barra degli strumenti e viene utilizzata per aggiungere elementi lingua come

istruzioni allo Spazio di Lavoro dell'editor di lingue. È complementare alla casella degli strumenti di workbench.



Elemento	Descrizione
Schede	Elenca gli elementi istruzione per categoria. Fare clic su una categoria per visualizzare le istruzioni all'interno di tale categoria. È anche possibile utilizzare i tasti freccia per selezionare categorie diverse. Fare clic sulle frecce a sinistra delle schede per scorrere le categorie. Fare clic sulla freccia giù a destra delle schede per visualizzare un elenco di tutte le categorie.
Istruzioni	Elenca le istruzioni corrispondenti alla categoria Schede selezionata. Una categoria può anche avere elementi istruzione base appuntati all'inizio dell'elenco. Fare clic sulle frecce a sinistra delle istruzioni per scorrere le istruzioni.
Ricerca	Filtra le istruzioni per categoria non appena vengono immessi i nomi o la parola chiave descrittiva. Fare clic sul nome categoria per visualizzare le istruzioni filtrate per tale categoria.
Preferiti	Personalizza un elenco di istruzioni che è possibile individuare e aggiungere rapidamente a un programma.

Vedere anche

[Aggiungere elementi istruzione dalla barra degli strumenti Istruzione](#) a pagina 27

Aggiungere elementi istruzione dalla barra degli strumenti Istruzione

Utilizzare la barra degli strumenti Istruzione per cercare, esplorare e aggiungere istruzioni allo Spazio di Lavoro dell'editor di lingue del programma Diagramma ladder (LD).

Per aggiungere elementi istruzione dalla barra degli strumenti Istruzione all'editor di lingue:

1. Fare clic su una scheda categoria che comprende l'istruzione da aggiungere. È possibile accedere alla scheda utilizzando i tasti freccia e spostarla mediante trascinamento.
2. Selezionare l'istruzione attraverso un'operazione di trascinamento o facendo clic su di essa.
3. (facoltativo) Per individuare rapidamente l'istruzione, fare clic nel campo di ricerca, quindi digitare per trovare l'elemento istruzione in base al nome o alla parola chiave. Per chiudere la ricerca e abilitare la navigazione con tasto freccia, premere il tasto **Esc**.
4. (Opzionale) Fare clic con il pulsante destro del mouse sull'istruzione e selezionare **Aggiungi ai Preferiti** per aggiungerla alla scheda **Preferiti**. In alternativa, selezionare **Rimuovi dai Preferiti** per rimuoverla. L'impostazione Preferiti verrà salvata automaticamente.

Vedere anche

[Barra degli strumenti Istruzione](#) a pagina 26

Elementi del ladder diagram (LD)

Gli elementi del diagramma Ladder sono i componenti utilizzati per compilare una Programmazione in Ladder Diagram. Tutti gli elementi elencati nella seguente tabella possono essere aggiunti a una Programmazione in Ladder Diagram all'interno di Connected Components Workbench.

Elemento	Descrizione
Piolo a pagina 29	Rappresenta un gruppo di elementi di circuito che portano all'attivazione di una bobina.
Blocco istruzione (LD) a pagina 37	Le istruzioni comprendono operatori, funzioni e blocchi funzione, tra cui blocchi funzione definiti dall'utente.
Diramazione a pagina 33	Due o più istruzioni in parallelo.
Bobina a pagina 37	Rappresenta l'assegnazione di output o variabili interne. In un programma LD, una bobina rappresenta un'azione.
Contatto a pagina 47	Rappresenta il valore o la funzione di una variabile di input o interna.
Ritorno a pagina 53	Rappresenta la fine condizionale dell'output di un diagramma.
Salto a pagina 54	Rappresenta la logica condizionale o non condizionale nel programma LD che controlla l'esecuzione dei diagrammi.

Vedere anche

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

Piolo

I rami sono componenti grafici dei programmi Diagramma ladder che rappresentano un gruppo di elementi circuitali che conduce all'attivazione di una bobina. Usare le etichette per identificare i rami all'interno del diagramma. Le etichette, insieme ai salti, controllano l'esecuzione di un diagramma. I commenti sono testo in formato libero che è possibile aggiungere sopra il ramo a scopo di documentazione.

Vedere anche

[Aggiunta di un ramo a un programma LD](#) a pagina 30

[Aggiunta di un'etichetta a un ramo](#) a pagina 31

[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

Aggiunta di un ramo a un programma LD

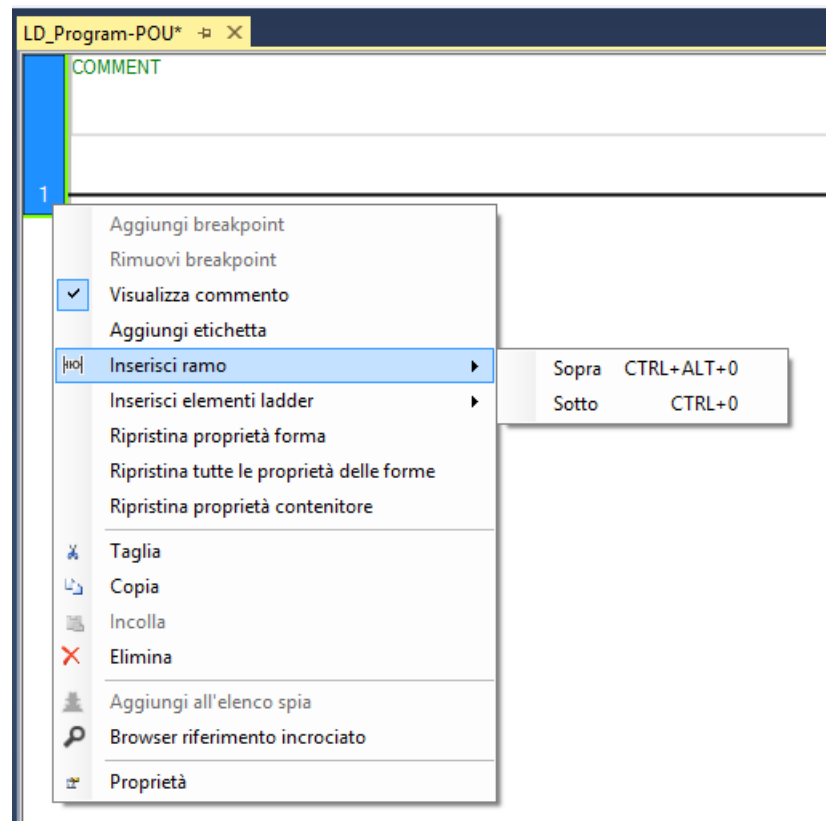
I rami sono componenti grafici del programma Diagramma ladder che rappresentano un gruppo di elementi circuitali che conduce all'attivazione di una bobina.

In Connected Components Workbench è possibile aggiungere un ramo a un programma Diagramma ladder da:

- Editor di lingue Diagramma ladder
- L'Editor multi-lingue si trova nel menu Strumenti
- Casella degli strumenti LD

Per aggiungere un elemento del ramo a un programma Diagramma ladder:

1. Nell'editor di lingue LD, fare clic con il pulsante destro del mouse su un ramo esistente, quindi:
 - Fare clic su **Copia**, quindi fare clic su **Incolla** per inserire una copia del ramo nell'editor di lingue.
 - Fare clic su **Inserisci ramo**, quindi:
 - Fare clic su **Sopra** per aggiungere il ramo sopra il ramo selezionato.
 - Fare clic su **Sotto** per aggiungere il ramo sotto il ramo selezionato.



- Selezionare un ramo o un elemento nell'editor di lingue LD e premere:
Premere **CTRL+ALT+0** per aggiungere il ramo sopra il ramo selezionato.

Premere **CTRL+0** per aggiungere il ramo sotto il ramo selezionato.

- Selezionare un ramo o un elemento nell'editor di lingue LD e premere:
 - Fare clic su **Strumenti > Editor multi-lingue > Inserisci ramo sotto** per aggiungere il ramo sotto il ramo selezionato.
 - Fare clic su **Strumenti > Editor multi-lingue > Inserisci ramo sopra** per aggiungere il ramo sopra il ramo selezionato.
1. (facoltativo) Aprire la casella degli strumenti LD per visualizzare l'elemento ramo.

- Per inserire un ramo sotto un ramo esistente, selezionare il ramo nell'editor di lingue LD, quindi fare doppio clic su **Ramo** nella casella degli strumenti LD.

Se non è selezionato un elemento prima di fare doppio clic su **Ramo** nella casella degli strumenti LD, il ramo viene inserito sotto l'ultimo ramo nell'editor di lingue LD.

- Selezionare **Ramo**, quindi trascinare l'elemento nell'editor di lingue LD.

Nell'editor di lingue LD viene mostrato un segno più (+) per visualizzare una destinazione valida. Rilasciare il pulsante del mouse per aggiungere l'elemento.



Suggerimento: Se il Diagramma ladder contiene più di 355 rami, per visualizzare rami aggiuntivi, utilizzare il menu a comparsa invece della barra di scorrimento.



Vedere anche

[Aggiunta di un'etichetta a un ramo](#) a pagina 31

[Tasti di scelta rapida LD](#) a pagina 59

[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Pioli](#) a pagina 29

Etichette ramo

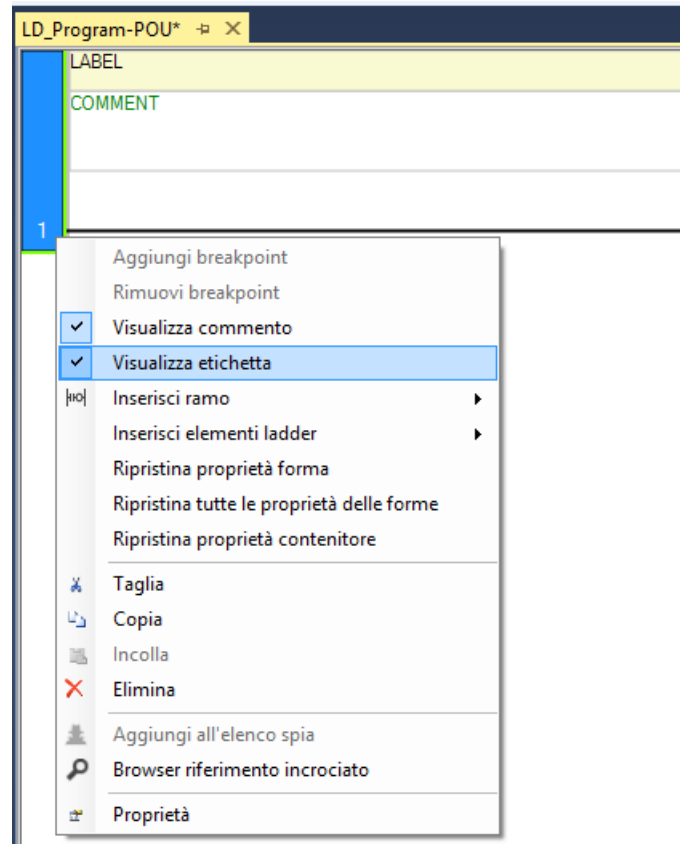
Le etichette sono aggiunte facoltative di ogni ramo nell'Editor di lingue del Diagramma ladder.

Le etichette possono avere un numero illimitato di caratteri e devono iniziare con una lettera o un carattere di sottolineatura, seguiti da lettere, numeri e caratteri di sottolineatura. Le etichette non possono contenere spazi o caratteri speciali (ad esempio +, -, o \).

Per aggiungere un'etichetta a un ramo nell'editor di lingue LD:

1. Per aggiungere un'etichetta a un ramo, fare clic con il pulsante destro del mouse sul ramo per aprire il menu Editor di lingue LD, quindi selezionare **Visualizza Etichetta**.

Il ramo viene aggiornato per includere l'etichetta e il menu Editor di lingue LD mostra un segno di spunta accanto a **Visualizza Etichetta**.



2. Selezionare l'**Etichetta** e digitare una descrizione.
3. (facoltativo) Per rimuovere l'etichetta, fare clic su **Visualizza Etichetta** nel menu Editor di lingue LD.

Vedere anche

[Pioli](#) a pagina 29

Commenti piolo

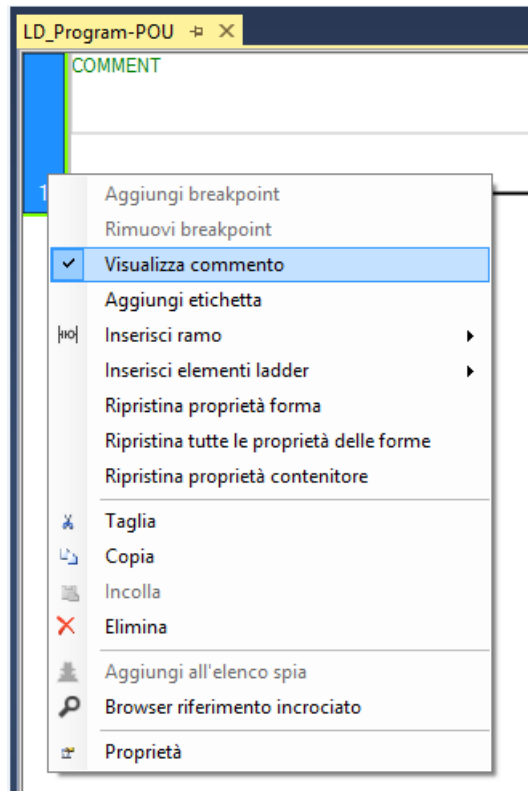
I commenti sono facoltativi per ogni ramo nell'Editor di lingue Diagramma ladder. Per impostazione predefinita, un commento è incluso quando si aggiunge un elemento ramo nell'editor di lingue LD.

I commenti sono:

- Inseriti nello spazio sopra il ramo.
- Salvati in formato RTF.
- Archiviati nel controllore.

Per aggiungere o rimuovere un commento per un ramo nell'editor di lingue LD:

1. Per rimuovere un commento, fare clic con il pulsante destro del mouse sul ramo per aprire il menu dell'editor di lingue LD.



2. Nel menu editor di lingue LD, fare clic su **Visualizza commento**. Il commento viene rimosso dal ramo e il segno di spunta accanto a **Visualizza commento** nell'editor di lingue LD viene rimosso dal menu.
3. (facoltativo) Per aggiungere un commento a un ramo, fare clic su **Visualizza commento** nel menu Editor di lingue LD.

Vedere anche

[Aggiunta di un ramo a un programma LD](#) a pagina 30

[Aggiunta di un'etichetta a un ramo](#) a pagina 31

[Elementi del ladder diagram \(LD\)](#) a pagina 29

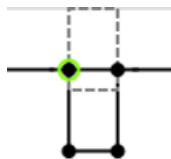
[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

[Pioli](#) a pagina 29

Diramazione

Le diramazioni creano routine alternativi per le connessioni. È possibile aggiungere rami paralleli agli elementi su un ramo in un editor di lingue Diagramma ladder.

Esempio di diramazione



Vedere anche

[Aggiunta di un ramo a un programma LD](#) a pagina 34

[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Tasti di scelta rapida LD](#) a pagina 59

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

Aggiunta di un ramo a un programma LD

I rami sono componenti grafici di programmazioni in Ladder Diagram (LD) che consentono di creare un routing alternativo per le connessioni e potrebbero includere rami paralleli.

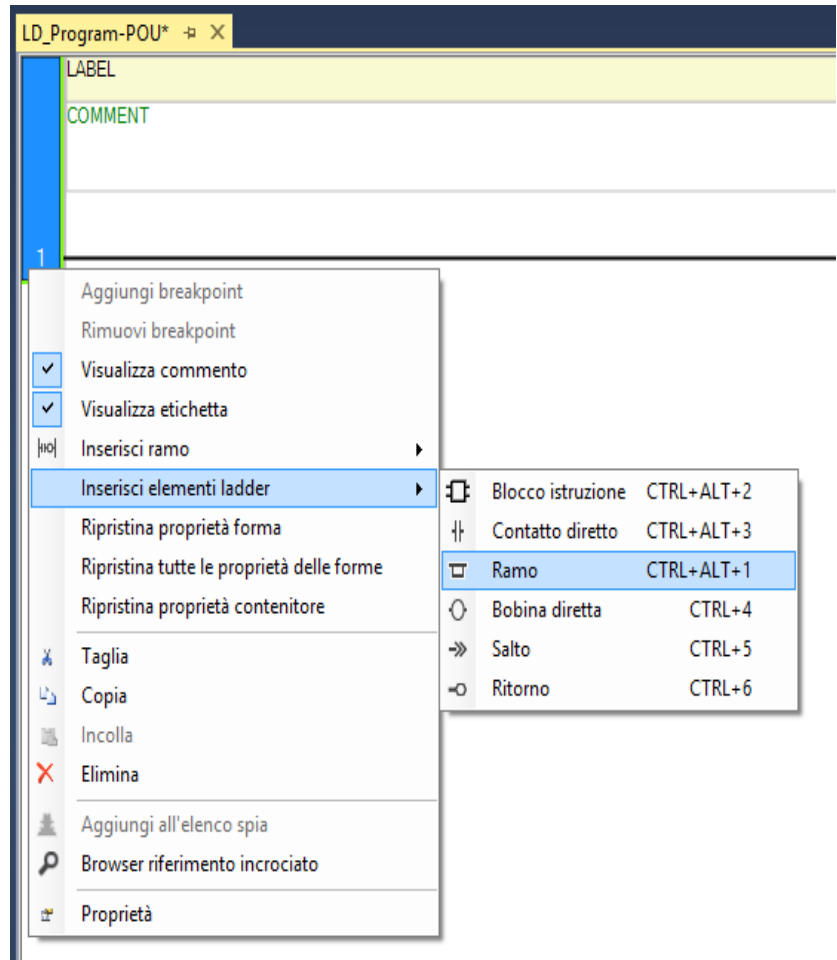
In Connected Components Workbench è possibile aggiungere un ramo a una programmazione in Ladder Diagram (LD) da:

- Editor di lingue Diagramma ladder
- L'Editor multi-lingue si trova nel menu Strumenti
- Casella degli strumenti LD
- Barra degli strumenti Istruzione (LD)

Per aggiungere un ramo a una programmazione in Ladder Diagram:

1. Nell'editor di lingue LD, verificare che la programmazione in LD disponga di un ramo definito. Quindi, effettuare una delle seguenti operazioni:
 - Fare clic con il pulsante destro del mouse su un ramo o su un elemento nell'editor di lingue LD, selezionare **Inserisci Elementi Ladder**, quindi fare clic su **Ramo**.

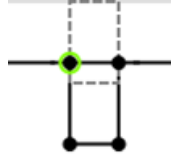
Se viene visualizzato il **selettore di variabili**, selezionare una variabile o fare clic su **OK** senza selezionare una variabile per aggiungere l'elemento **Ramo**.



- Selezionare un ramo o un elemento nell'editor di lingue LD e premere:
 - CTRL+ALT+1** per aggiungere il ramo a sinistra dell'elemento o del ramo selezionato.
 - CTRL+ALT+1** per aggiungere il ramo a sinistra dell'elemento selezionato.
- Selezionare un ramo o un elemento nell'editor di lingue LD, quindi:
 - Fare clic su **Strumenti > Editor multi-lingue > Inserisci prima ramo** per aggiungere il ramo a sinistra dell'elemento selezionato.
 - Fare clic su **Strumenti > Editor multi-lingue > Inserisci dopo ramo** per aggiungere il ramo a destra dell'elemento selezionato.
- Aprire la casella degli strumenti LD per visualizzare l'elemento ramo, quindi:
 - Fare doppio clic sull'elemento ramo per aggiungerlo all'editor di lingue LD.
 - Trascinare l'elemento ramo nell'editor di lingue LD e posizionarlo sul ramo.

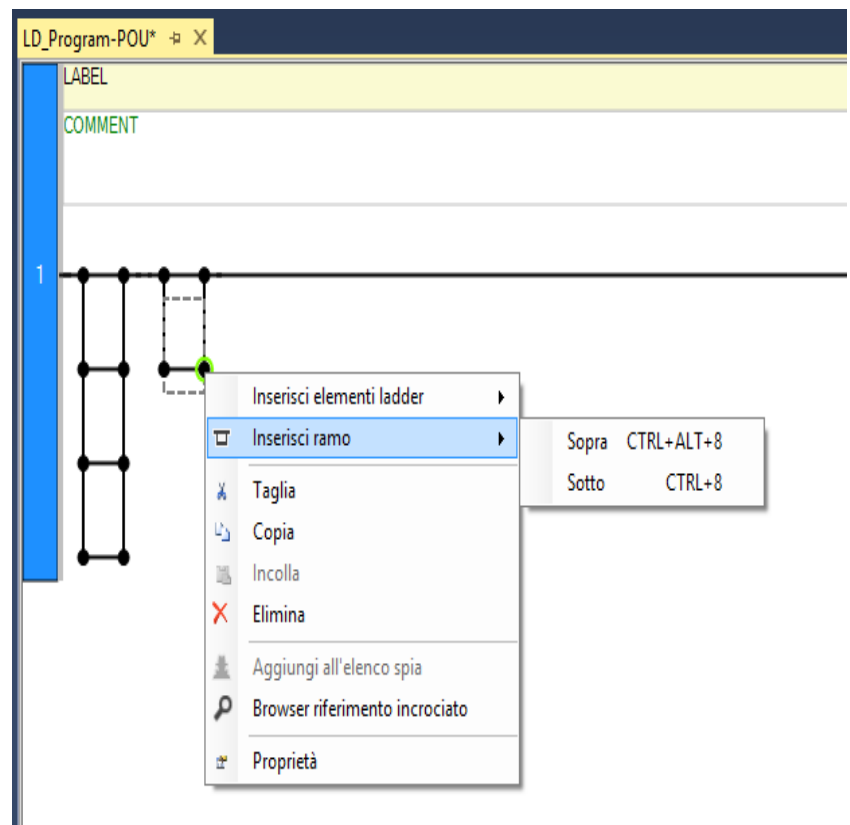
Nell'editor di lingue LD viene mostrato un segno più (+) per visualizzare una destinazione valida. Rilasciare il pulsante del mouse per aggiungere l'elemento.

1. (facoltativo) Per inserire un ramo parallelo:
 - a. Nell'editor di lingue LD, fare clic con il pulsante destro del mouse sul ramo per aprire il menu dell'editor di lingue LD.



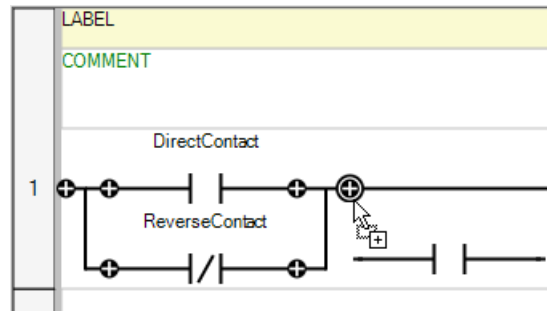
- b. Dal menu editor di lingue LD, selezionare **Inserisci ramo**, quindi fare clic su:

Sopra per inserire un ramo sopra il ramo selezionato.
Sotto per inserire un ramo sotto il ramo selezionato.



2. (facoltativo) Per spostare un elemento ramo in un'altra posizione in una programmazione in LD, selezionare l'elemento e trascinarlo in una nuova posizione nella programmazione in LD.
 Nell'editor di lingue LD viene mostrato un segno più (+) per visualizzare una destinazione valida. Rilasciare il pulsante del

mouse per inserire l'elemento nella posizione di destinazione.



Vedere anche

[Diramazione](#) a pagina 33

[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Tasti di scelta rapida LD](#) a pagina 59

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

Blocco istruzione (LD)

Un elemento Blocco istruzione Diagramma ladder è un elemento funzionale conforme a IEC 61131-3 in un programma LD che può essere un blocco funzione, una funzione, un blocco funzione definito dall'utente, una funzione definita dall'utente o un operatore.

Vedere anche

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

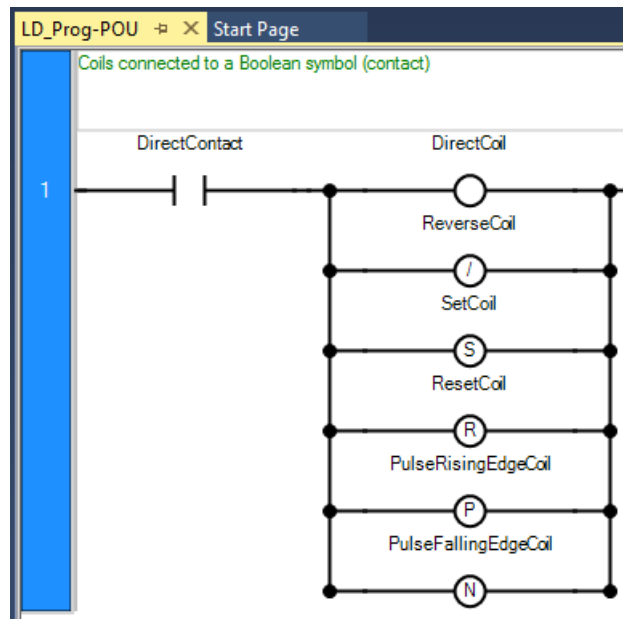
[Elementi del ladder diagram \(LD\)](#) a pagina 29

Bobina

Le bobine sono componenti grafici di Programmazione in Ladder Diagram, che rappresentano l'assegnazione di un'uscita o di una variabile interna. Nelle programmazioni LD, una bobina rappresenta un'azione. Una bobina deve essere collegata a sinistra a un simbolo booleano come un contatto o all'output booleano di un blocco di istruzioni. Le bobine possono essere aggiunte solo a un ramo definito nell'editor di lingue LD. La definizione di bobina può essere modificata dopo che la bobina è stata aggiunta al ramo.

L'esempio seguente mostra i tipi di elemento della bobina disponibili per la Programmazione in Ladder Diagram.

Esempio: Bobine



Vedere anche

[Aggiunta di elementi a bobina](#) a pagina 38

[Bobina diretta](#) a pagina 41

[Bobina invertita](#) a pagina 41

[Bobina impostata](#) a pagina 44

[Bobina ripristinata](#) a pagina 45

Aggiunta di elementi a bobina

Le bobine sono componenti grafici di programmi Diagramma ladder, che rappresentano un'azione eseguita, un'uscita o una variabile interna.

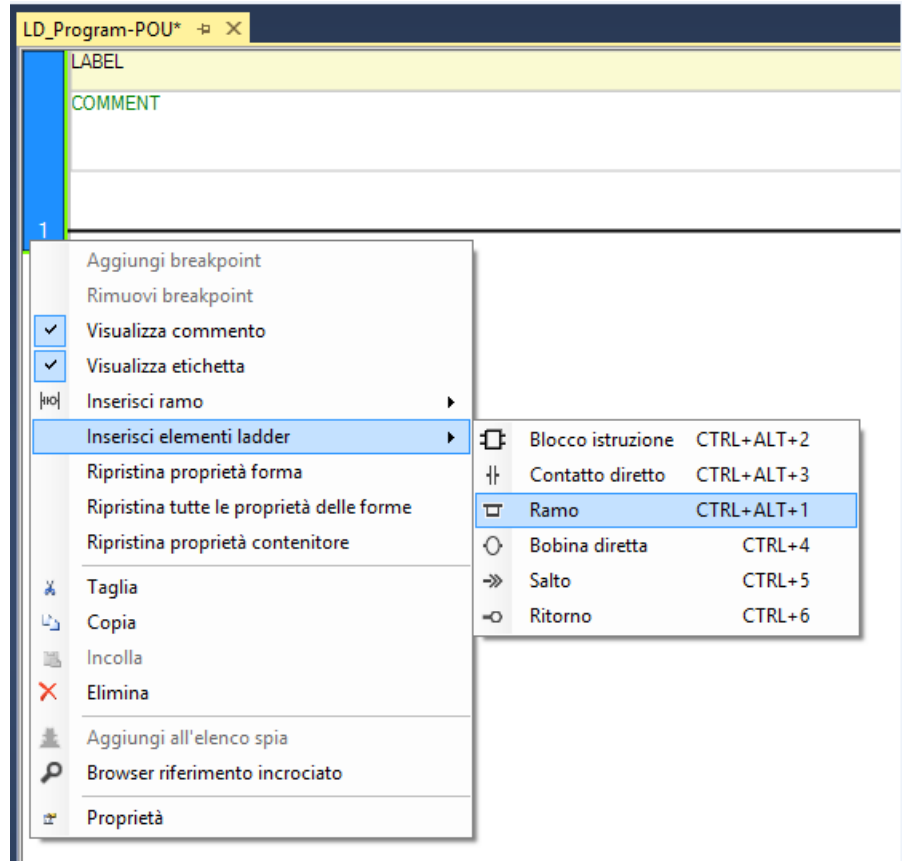
In Connected Components Workbench è possibile aggiungere una bobina a un programma Diagramma ladder da:

- Editor di lingue Diagramma ladder
- L'Editor multi-lingue si trova nel menu Strumenti
- Casella degli strumenti LD
- Barra degli strumenti Istruzione (LD)

Per aggiungere un elemento a bobina a un programma Diagramma ladder:

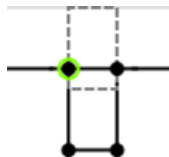
1. Nell'editor di lingue LD, verificare che il programma LD disponga di un ramo definito per la bobina. Quindi, effettuare una delle seguenti operazioni:

- Fare clic con il pulsante destro del mouse su un ramo o un elemento nell'editor di lingue LD, selezionare **Inserisci Elementi ladder**, quindi fare clic su **Bobina diretta**.
Se viene visualizzato il **selettore di variabili**, selezionare una variabile o fare clic su **OK** senza selezionare una variabile per aggiungere l'elemento **Bobina diretta**.

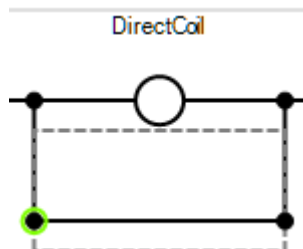


- Selezionare un ramo o un elemento nell'editor di lingue LD, quindi premere **CTRL+4** per aggiungere la **Bobina diretta** a destra del ramo.
 - Selezionare un ramo o un elemento nell'editor di lingue LD, quindi fare clic su **Strumenti > Editor multi-lingue > Inserisci bobina** per aggiungere la bobina a destra del ramo.
 - Aprire la casella degli strumenti LD per visualizzare gli elementi bobina (**Bobina diretta**, **Bobina inversa**, **Imposta bobina**, **Reimposta bobina**, **Bobina fronte di salita dell'impulso**, **Bobina fronte di discesa dell'impulso**), quindi:
 - Fare doppio clic sull'elemento a bobina per aggiungerlo all'editor di lingue LD.
 - Trascinare l'elemento a bobina nell'editor di lingue LD e posizionarlo sul ramo.
 - Nell'editor di lingue LD viene mostrato un segno più (+) per visualizzare una destinazione valida. Rilasciare il pulsante del mouse per aggiungere l'elemento.
- (Facoltativo) Per inserire una bobina parallela:
 - Nell'editor di lingue LD, verificare che il programma LD abbia una diramazione definita, quindi fare clic con il pulsante destro del

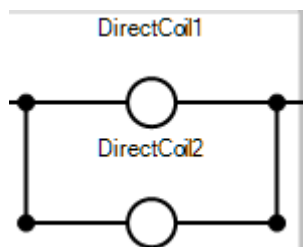
mouse sul primo livello della diramazione, per aprire il menu Editor di lingue LD.



- b. Nel menu Editor di lingue LD, selezionare **Inserisci elementi Ladder**, quindi fare clic su **Bobina diretta**. L'elemento viene inserito nel livello superiore della diramazione.



- c. Fare clic con il pulsante destro del mouse sul livello inferiore della diramazione, per aprire il menu Editor di lingue LD.
- d. Nel menu Editor di lingue LD, selezionare **Inserisci elementi Ladder**, quindi fare clic su **Bobina diretta**. L'elemento viene inserito nel livello inferiore della diramazione.



2. (Facoltativo) Per modificare il tipo di bobina, nell'editor di lingue selezionare la bobina, quindi premere la **barra spaziatrice** fino a quando il tipo di bobina viene visualizzato nell'editor di lingue.

Ogni volta che si preme la barra spaziatrice, il tipo di bobina cambia da diretto, a inverso, a impostato, a ripristinato, a fronte salita dell'impulso, a fronte discesa dell'impulso.

Vedere anche

[Diramazione](#) a pagina 33

[Bobine](#) a pagina 37

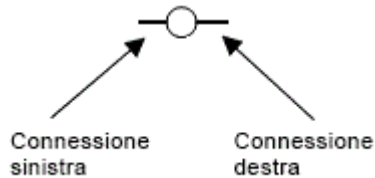
[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Tasti di scelta rapida LD](#) a pagina 59

Bobina diretta

Le bobine sono componenti grafici di Programmazione in Ladder Diagram, che rappresentano l'assegnazione di un'uscita o di una variabile interna.

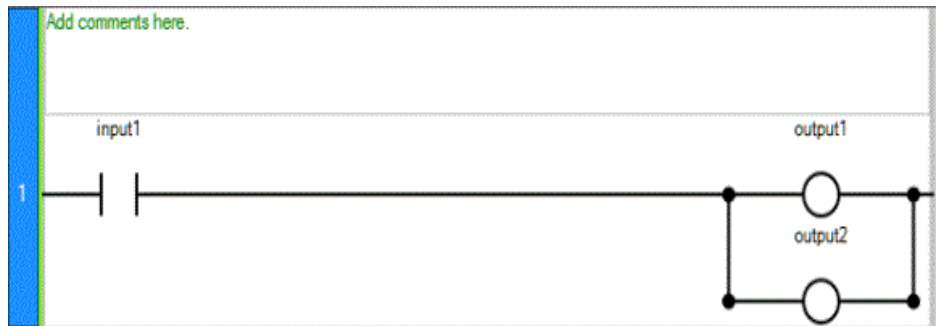
La bobina diretta supporta l'output booleano di uno stato booleano della linea di collegamento.



La variabile associata viene assegnata con lo stato booleano della connessione di sinistra. Lo stato della connessione di sinistra si propaga alla connessione di destra. La connessione di destra deve essere collegata alla rotaia di alimentazione verticale di destra (a meno che non siano presenti bobine parallele, caso in cui solo la bobina superiore deve essere collegata alla rotaia di alimentazione verticale di destra).

La variabile Booleana associata deve essere un'uscita oppure deve essere definita dall'utente.

Esempio di bobina diretta



Vedere anche

[Aggiunta di elementi a bobina](#) a pagina 38

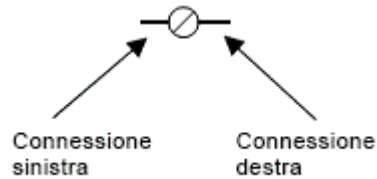
[Bobine](#) a pagina 37

[Bobina invertita](#) a pagina 41

Bobina invertita

Le bobine sono componenti grafici di Programmazione in Ladder Diagram, che rappresentano l'assegnazione di un'uscita o di una variabile interna.

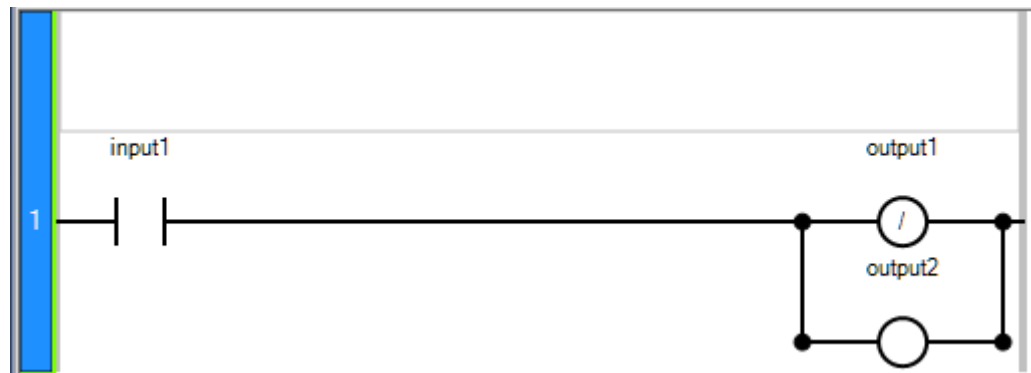
L'elemento a bobina invertito supporta un output booleano in base alla negazione booleana dello stato di una linea di collegamento.



La variabile associata viene assegnata con la negazione booleana dello stato della connessione di sinistra. Lo stato della connessione di sinistra si propaga alla connessione di destra. La connessione di destra deve essere collegata alla rotaia di alimentazione verticale di destra (a meno che non siano presenti bobine parallele, caso in cui solo la bobina superiore deve essere collegata alla rotaia di alimentazione verticale di destra).

La variabile Booleana associata deve essere un'uscita oppure deve essere definita dall'utente.

Esempio di Bobina inversa



Vedere anche

[Aggiunta di elementi a bobina](#) a pagina 38

[Bobine](#) a pagina 37

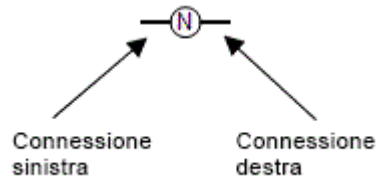
[Bobina diretta](#) a pagina 41

[Elementi del ladder diagram \(LD\)](#) a pagina 29

Bobina fronte di discesa impulso

Le bobine sono componenti grafici di Programmazione in Ladder Diagram, che rappresentano l'assegnazione di un'uscita o di una variabile interna.

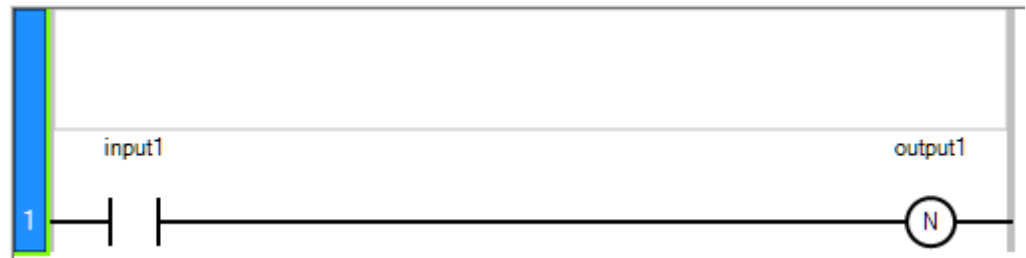
Le bobine fronte di discesa impulso (o negative) supportano l'uscita dello stato booleano di una linea di connessione.



La variabile associata viene impostata su TRUE se lo stato booleano della connessione di sinistra scende da TRUE a FALSE. La variabile di uscita viene reimpostata a FALSE in tutti gli altri casi. Lo stato della connessione di sinistra si propaga alla connessione di destra. La connessione di destra deve essere collegata alla rotaia di alimentazione verticale di destra (a meno che non siano presenti bobine parallele, caso in cui solo la bobina superiore deve essere collegata alla rotaia di alimentazione verticale di destra).

La variabile Booleana associata deve essere un'uscita oppure deve essere definita dall'utente.

Esempio di Bobina fronte di discesa impulso



Vedere anche

[Aggiunta di elementi a bobina](#) a pagina 38

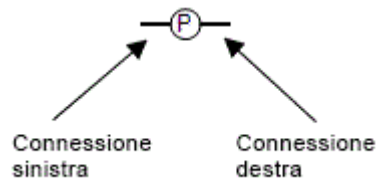
[Bobine](#) a pagina 37

[Bobina fronte di salita impulso](#) a pagina 43

Bobina fronte di salita impulso

Le bobine sono componenti grafici di Programmazione in Ladder Diagram, che rappresentano l'assegnazione di un'uscita o di una variabile interna.

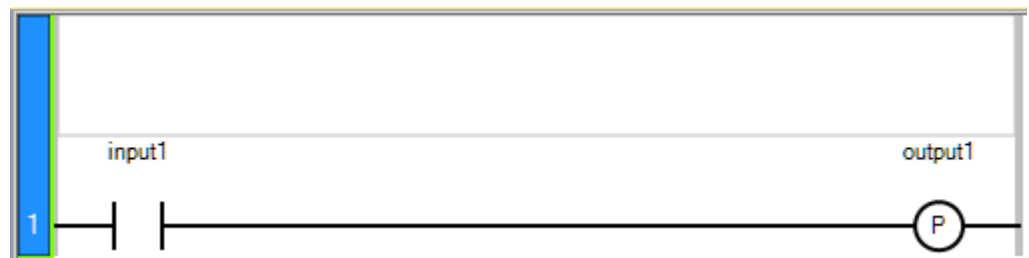
Le bobine fronte di salita impulso (o positive) supportano l'uscita dello stato booleano di una linea di connessione.



La variabile associata viene impostata su TRUE se lo stato booleano della connessione di sinistra sale da FALSE a TRUE. La variabile di uscita viene reimpostata a FALSE in tutti gli altri casi. Lo stato della connessione di sinistra si propaga alla connessione di destra. La connessione di destra deve essere collegata alla rotaia di alimentazione verticale di destra (a meno che non siano presenti bobine parallele, caso in cui solo la bobina superiore deve essere collegata alla rotaia di alimentazione verticale di destra).

La variabile booleana associata deve essere di uscita oppure deve essere definita dall'utente.

Esempio di Bobina fronte di salita impulso



Vedere anche

[Aggiunta di elementi a bobina](#) a pagina 38

[Bobine](#) a pagina 37

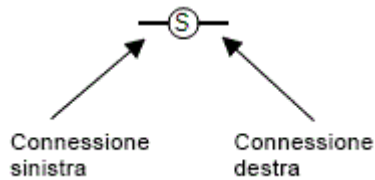
[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Bobina fronte di discesa impulso](#) a pagina 42

Bobina impostata

Le bobine sono componenti grafici di Programmazione in Ladder Diagram, che rappresentano l'assegnazione di un'uscita o di una variabile interna. Nelle programmazioni LD, una bobina rappresenta un'azione.

Le bobine impostate supportano l'uscita dello stato booleano di una linea di connessione.



La variabile associata viene impostata su TRUE quando lo stato booleano della connessione di sinistra diviene TRUE. La variabile di uscita mantiene tale valore fino a che viene generato un ordine inverso da una Bobina ripristinata. Lo stato della connessione di sinistra si propaga alla connessione di destra. La connessione di destra deve essere collegata alla rotaia di alimentazione verticale di destra (a meno che non siano presenti bobine parallele, caso in cui solo la bobina superiore deve essere collegata alla rotaia di alimentazione verticale di destra).

La variabile Booleana associata deve essere un'uscita oppure deve essere definita dall'utente.

Esempio di Imposta bobina



Vedere anche

[Aggiunta di elementi a bobina](#) a pagina 38

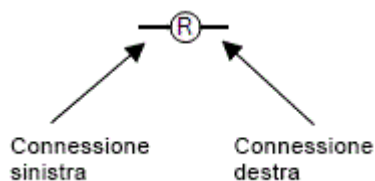
[Bobine](#) a pagina 37

[Bobina ripristinata](#) a pagina 45

Bobina ripristinata

Le bobine sono componenti grafici di Programmazione in Ladder Diagram, che rappresentano l'assegnazione di un'uscita o di una variabile interna.

Le bobine ripristinate supportano l'uscita dello stato booleano di una linea di connessione.



La variabile associata viene impostata su FALSE quando lo stato booleano della connessione di sinistra diviene TRUE. La variabile di uscita mantiene tale valore fino a che viene generato un ordine inverso da una Bobina impostata. Lo stato della connessione di sinistra si propaga alla connessione di destra. La connessione di destra deve essere collegata alla rotaia di alimentazione verticale di destra (a meno che non siano presenti bobine parallele, caso in cui solo la bobina superiore deve essere collegata alla rotaia di alimentazione verticale di destra).

La variabile booleana associata deve essere di uscita oppure deve essere definita dall'utente.

Esempio di Reimposta bobina



Vedere anche

[Bobine](#) a pagina 37

[Aggiunta di elementi a bobina](#) a pagina 38

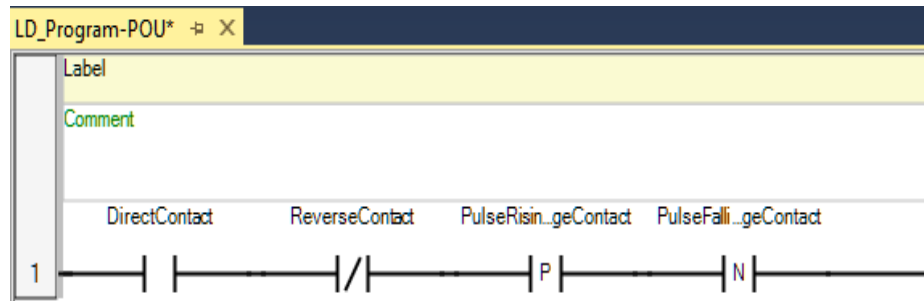
[Bobina impostata](#) a pagina 44

Contatto

I contatti sono componenti grafici di Programmazioni in Ladder Diagram. A seconda del tipo, un contatto rappresenta il valore o la funzione di un ingresso o di una variabile interna. I contatti possono essere aggiunti solo a un piolo definito nell'editor di lingue LD. Dopo aver aggiunto un contatto, è possibile modificarne la definizione.

L'esempio seguente mostra i tipi di elemento di contatto disponibili per la Programmazione in Ladder Diagram.

Esempio: Contatti



Vedere anche

[Aggiunta di elementi contatto](#) a pagina 47

[Contatto diretto](#) a pagina 50

[Contatto invertito](#) a pagina 51

[Contatto fronte di salita impulso](#) a pagina 51

[Contatto fronte di discesa impulso](#) a pagina 52

Aggiunta di un contatto a un programma LD

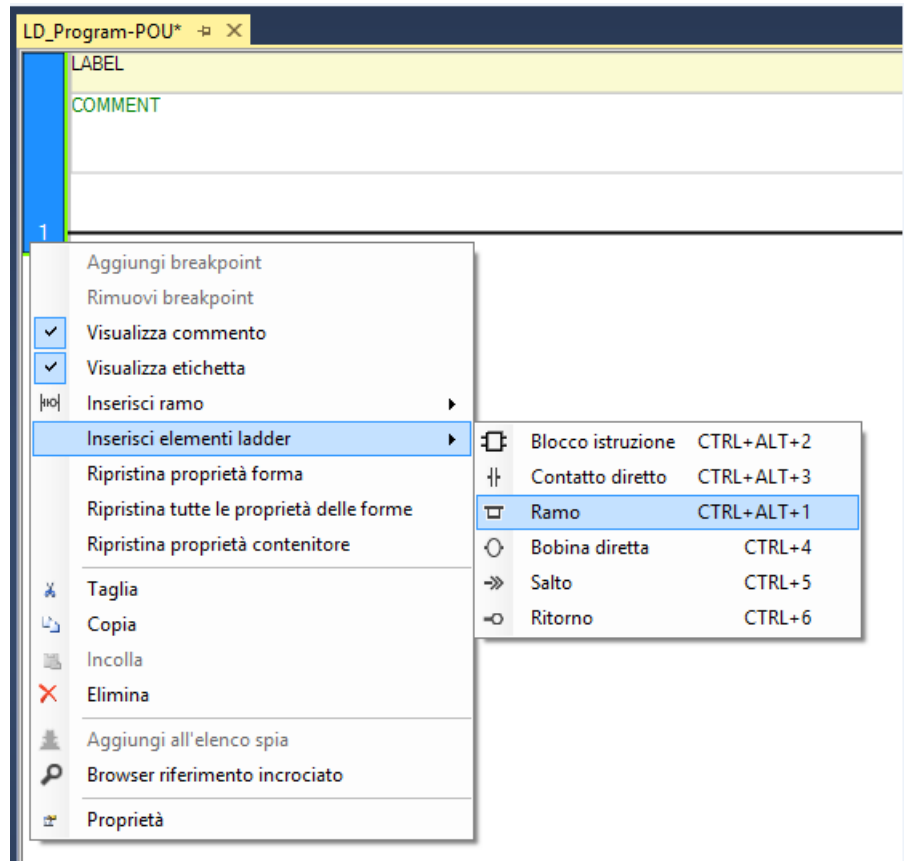
I contatti sono componenti grafici di una Programmazione in Ladder Diagram. A seconda del tipo, un contatto rappresenta il valore o la funzione di un ingresso o di una variabile interna. I contatti possono essere aggiunti solo a un piolo definito nell'editor di lingue LD.

In Connected Components Workbench è possibile aggiungere un contatto a un programma Diagramma ladder da:

- Editor di lingue Diagramma ladder
- L'Editor multi-lingue si trova nel menu Strumenti
- Casella degli strumenti LD
- Barra degli strumenti Istruzione (LD)

Per aggiungere un elemento di contatto a un programma Diagramma ladder:

1. Nell'editor di lingue LD, verificare che il programma LD disponga di un ramo definito per il contatto. Quindi, effettuare una delle seguenti operazioni:
 - Fare clic con il pulsante destro del mouse su un ramo o un elemento nell'editor di lingue LD, selezionare **Inserisci Elementi ladder**, quindi fare clic su **Contatto Diretto**.
Se viene visualizzato il **selettore di variabili**, selezionare una variabile o fare clic su **OK** senza selezionare una variabile per aggiungere l'elemento **Contatto diretto**.



- Selezionare un ramo o un elemento nell'editor di lingue LD e premere:
CTRL+ALT+3 per aggiungere l'elemento Contatto diretto a sinistra dell'elemento o del ramo selezionato.
CTRL+3 per aggiungere l'elemento Contatto diretto a destra dell'elemento o del ramo selezionato.
- Selezionare un ramo o un elemento nell'editor di lingue LD, quindi:
Fare clic su **Strumenti > Editor multi-lingue > Inserisci prima contatto** per aggiungere il contatto a sinistra dell'elemento o del ramo selezionato.
Fare clic su **Strumenti > Editor multi-lingue > Inserisci dopo contatto** per aggiungere il contatto a destra dell'elemento o del ramo selezionato.

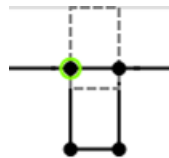
- Aprire la casella degli strumenti LD per visualizzare gli elementi di contatto (**Contatto diretto**, **Contatto inverso**, **Contatto sul fronte di salita dell'impulso**, **Contatto sul fronte di discesa dell'impulso**), quindi:

Fare doppio clic sull'elemento di contatto per aggiungerlo all'editor di lingue LD.

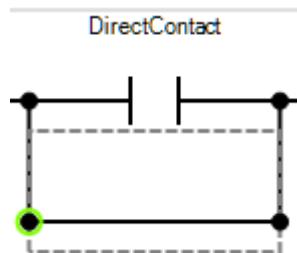
Trascinare l'elemento di contatto nell'editor di lingue LD e posizionarlo sul ramo.

Nell'editor di lingue LD viene mostrato un segno più (+) per visualizzare una destinazione valida. Rilasciare il pulsante del mouse per aggiungere l'elemento.

1. (Facoltativo) Per inserire un contatto parallelo:
 - a. Nell'editor di lingue LD, verificare che il programma LD abbia una diramazione definita, quindi fare clic con il pulsante destro del mouse sul primo livello della diramazione, per aprire il menu Editor di lingue LD.

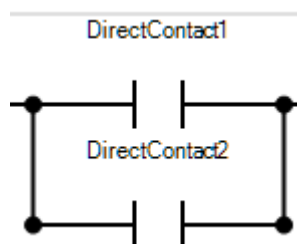


- b. Nel menu Editor di lingue LD, selezionare **Inserisci elementi Ladder**, quindi fare clic su **Contatto diretto**. L'elemento viene inserito nel livello superiore della diramazione.



- c. Fare clic con il pulsante destro del mouse sul livello inferiore della diramazione, per aprire il menu Editor di lingue LD.

- d. Nel menu Editor di lingue LD, selezionare **Inserisci elementi Ladder**, quindi fare clic su **Contatto diretto**. L'elemento viene inserito nel livello inferiore della diramazione.



2. (Facoltativo) Per modificare il tipo di contatto, nell'editor di lingue selezionare il contatto, quindi premere la **Barra spaziatrice** fino a quando il tipo di contatto viene visualizzato nell'editor di lingue.

Ogni volta che si preme la barra, il tipo di contatto cambia da diretto a inverso, a fronte di salita dell'impulso, a fronte di discesa dell'impulso.

Vedere anche

[Diramazione](#) a pagina 33

[Contatti](#) a pagina 47

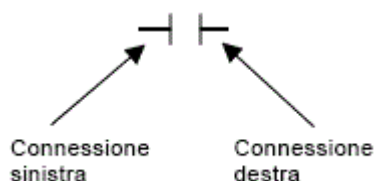
[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Tasti di scelta rapida LD](#) a pagina 59

Contatto diretto

I contatti sono componenti grafici di Programmazioni in Ladder Diagram.

I contatti diretti supportano le operazioni booleane tra uno stato della linea di connessione e una variabile booleana.



Lo stato della linea di collegamento a destra del contatto è l'AND logico tra lo stato della linea di collegamento sinistra e il valore della variabile associata al contatto.

Esempio di contatto diretto



Vedere anche

[Aggiunta di un elemento contatto](#) a pagina 47

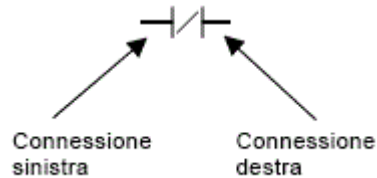
[Contatti](#) a pagina 47

[Contatto invertito](#) a pagina 51

Contatto invertito

I contatti sono componenti grafici di Programmazioni in Ladder Diagram.

I contatti invertiti consentono un'operazione booleana tra lo stato di una linea di connessione e la negazione booleana di una variabile booleana.



Lo stato della linea di collegamento a destra del contatto è l'AND logico tra lo stato della linea di collegamento sinistra e la negazione booleana del valore della variabile associata al contatto.

Esempio di Contatto inverso



Vedere anche

[Aggiunta di elementi contatto](#) a pagina 47

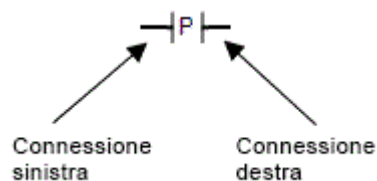
[Contatti](#) a pagina 47

[Contatto diretto](#) a pagina 50

Contatto fronte di salita impulso

I contatti sono componenti grafici di Programmazioni in Ladder Diagram.

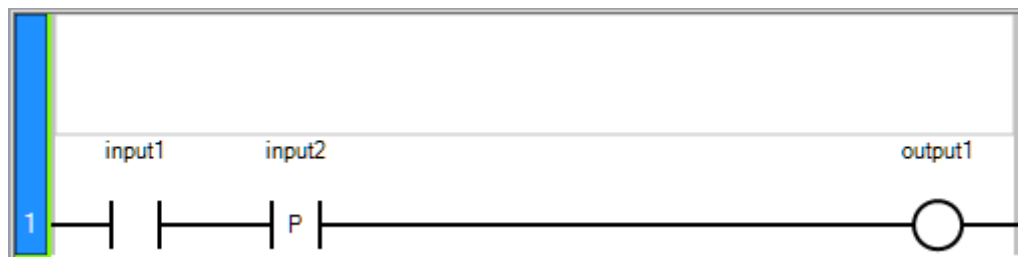
I contatti fronte di salita impulso (o positivi) consentono un'operazione booleana tra lo stato di una linea di connessione e il fronte di salita di una variabile booleana.



Lo stato della linea di connessione a destra del contatto è TRUE quando lo stato della linea di connessione a sinistra è TRUE e lo stato

della variabile associata sale da FALSE a TRUE. Lo stato viene reimpostato a FALSE in tutti gli altri casi.

Esempio di Contatto sul fronte di salita dell'impulso



Raccomandazione: limitare l'uso di variabili di uscita con contatti fronte

Si consiglia di non utilizzare output o variabili con un contatto sul fronte di salita dell'impulso (positivo) o un contatto sul fronte di discesa dell'impulso (negativo). Tali contatti sono destinati agli impulsi fisici in un ladder diagram. Per individuare il fronte di una variabile o di un output, è consigliato l'utilizzo del blocco funzione R_TRIG/F_TRIG, che è supportato e funziona in qualsiasi linguaggio e in qualsiasi punto del programma.

Vedere anche

[Aggiunta di elementi contatto](#) a pagina 47

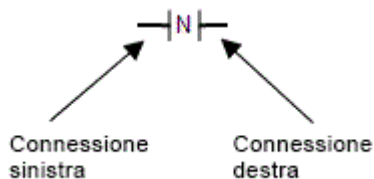
[Contatti](#) a pagina 47

[Contatto fronte di discesa impulso](#) a pagina 52

Contatto fronte di discesa impulso

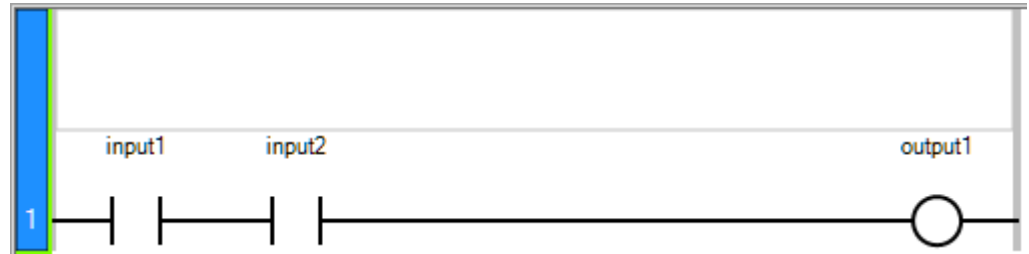
I contatti sono componenti grafici di Programmazioni in Ladder Diagram.

I contatti fronte di discesa impulso (o negativi) consentono un'operazione booleana tra lo stato di una linea di connessione e il fronte di discesa di una variabile booleana.



Lo stato della linea di connessione a destra del contatto è TRUE quando lo stato della linea di connessione a sinistra è TRUE e lo stato della variabile associata cade da TRUE a FALSE. Lo stato viene reimpostato a FALSE in tutti gli altri casi.

Esempio di Contatto sul fronte di discesa dell'impulso



Raccomandazione: limitare l'uso di variabili di uscita con contatti fronte

Si consiglia di non utilizzare output o variabili con un contatto sul fronte di salita dell'impulso (positivo) o un contatto sul fronte di discesa dell'impulso (negativo). Tali contatti sono destinati agli impulsi fisici in un ladder diagram. Per individuare il fronte di una variabile o di un output, è consigliato l'utilizzo del blocco funzione R_TRIG/F_TRIG, che è supportato e funziona in qualsiasi linguaggio e in qualsiasi punto del programma.

Vedere anche

[Aggiunta di elementi contatto](#) a pagina 47

[Contatto](#) a pagina 47

[Contatto fronte di salita impulso](#) a pagina 51

Ritorno

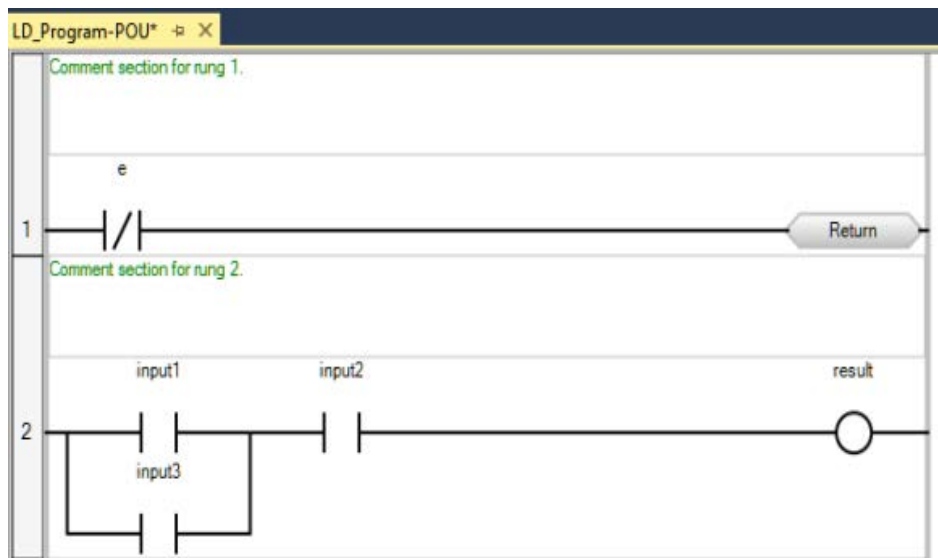
Vengono restituite uscite che rappresentano una fine condizionale di un programma Diagramma ladder.

Non è possibile posizionare connessioni a destra di un elemento ritorno.

Se la linea di collegamento di sinistra ha lo stato Booleana TRUE, il diagramma termina senza che vengano eseguite le equazioni nelle righe successive del diagramma stesso.

Se il diagramma LD ha una funzione, il suo nome viene associato a una bobina di uscita per impostare il valore di ritorno (restituito al programma che effettua la chiamata).

Esempio di valori restituiti



Per inserire un ritorno in programma Diagramma ladder:

Effettuare una delle seguenti operazioni:

- Fare clic con il pulsante destro del mouse su un ramo o su un elemento nell'editor di lingue LD, selezionare **Inserisci gli elementi Ladder**, quindi fare clic su **Restituisci**.
- Selezionare un ramo o un elemento nell'editor di lingue LD, quindi premere **CTRL+6**.
- Selezionare un ramo o un elemento nell'editor di lingue LD, quindi fare clic su **Strumenti > Editor multi-lingue > Inserisci ritorno**.
- Aprire la casella degli strumenti LD per visualizzare l'elemento di ritorno, quindi:
 - Fare doppio clic sull'elemento di ritorno per aggiungerlo all'editor di lingue LD.
 - Trascinare l'elemento di ritorno nell'editor di lingue LD e posizionarlo sul ramo.
 - Nell'editor di lingue LD viene mostrato un segno più (+) per visualizzare una destinazione valida. Rilasciare il pulsante del mouse per aggiungere l'elemento.

Vedere anche

[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

Salto

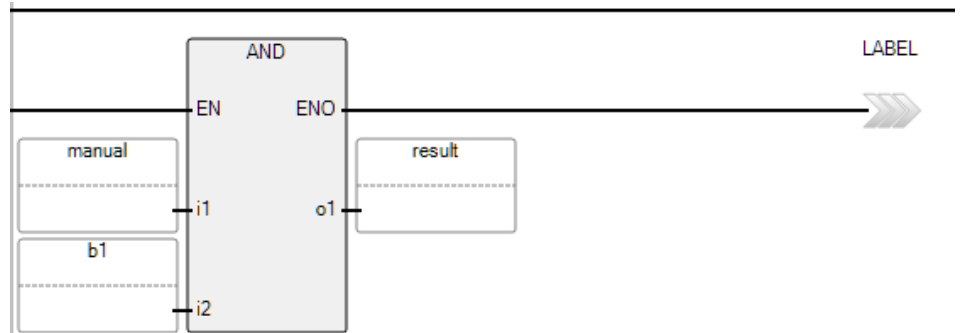
I salti sono elementi condizionali o non condizionali che controllano l'esecuzione di un programma Diagramma ladder.

Notazione di salto

La seguente notazione indica un salto a un'etichetta:

>>LABEL: salta a un'etichetta con nome "LABEL"

Esempio di salto



Per inserire un salto:

Effettuare una delle seguenti operazioni:

- Fare clic con il pulsante destro del mouse su un ramo o su un elemento nell'editor di lingue LD, selezionare **Inserisci gli elementi ladder**, quindi fare clic su **Salta**.
- Selezionare un ramo o un elemento nell'editor di lingue LD, quindi premere **CTRL+5**.
- Selezionare un ramo o un elemento nell'editor di lingue LD, quindi fare clic su **Strumenti > Editor multi-lingue > Inserisci salto**.
- Aprire la casella degli strumenti LD per visualizzare l'elemento di salto, quindi:
 - Fare doppio clic sull'elemento di salto per aggiungerlo all'editor di lingue LD.
 - Trascinare l'elemento di salto nell'editor di lingue LD e posizionarlo sul ramo.
 - Nell'editor di lingue LD viene mostrato un segno più (+) per visualizzare una destinazione valida. Rilasciare il pulsante del mouse per aggiungere l'elemento.

Vedere anche

[Elementi del ladder diagram \(LD\)](#) a pagina 29

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

[Ritorni](#) a pagina 53

Blocchi di istruzione nei programmi LD

Il set di istruzioni Connected Components Workbench include i blocchi di istruzioni IEC 61131-3 conformi. I blocchi di istruzioni comprendono i blocchi funzione, le funzioni e gli operatori. L'utente può collegare input

e output dei blocchi di istruzioni a variabili, contatti, bobine o altri input e output di blocchi di istruzione.

Convenzioni per il blocco di istruzioni

Le specifiche IEC61131-3 relative ai linguaggi di programmazione affrontano numerosi aspetti legati ai controllori programmabili, tra cui l'esecuzione del sistema operativo, le definizioni dei dati, i linguaggi di programmazione e i set di istruzioni. Di conseguenza, la specifica IEC61131-3 fornisce un set di funzionalità di base che è possibile estendere secondo le applicazioni dell'utente finale.

Nomi blocco di istruzioni

Funzioni e blocchi funzione sono rappresentati da una casella che visualizza il nome dell'istruzione e i nomi abbreviati dei parametri. Per i blocchi funzione, il nome dell'istanza viene visualizzato sopra il nome del blocco funzione.

Parametri di ritorno del blocco di istruzioni

- Il parametro di ritorno di una funzione ha lo stesso nome della funzione. Il parametro di ritorno è l'unico output.
- I parametri di ritorno di un blocco funzione possono avere qualsiasi nome. Parametri di ritorno multipli possono fornire più output.
- È possibile definire i parametri dei programmi per diversi dispositivi esplorando le schede dei singoli dispositivi visualizzate nella schermata **Parametro**.

Vedere anche

[Elementi del ladder diagram \(LD\)](#) a pagina 29

Utilizzo nell'editor di lingue LD

Quando si aggiungono elementi a un ramo nella Programmazione in Ladder Diagram, questi vengono aggiunti secondo i seguenti criteri.

- Il primo elemento di un piolo viene inserito nella posizione selezionata nel ladder diagram.
- I successivi elementi vengono inseriti a destra dell'elemento selezionato nel piolo.
- Impossibile inserire un elemento a destra di un ritorno bobina o salto.

Diversi metodi per aggiungere un elemento alla Programmazione in Ladder Diagram:

- **Editor ladder LD**

- Aggiungere elementi, eliminare elementi, copiare e incollare elementi.
- Utilizzare i tasti di scelta rapida LD per aggiungere elementi.
- **Editor di testo LD**
 - Aggiunge, modifica ed elimina gli elementi.
 - Copiare e incollare elementi da/verso RSLogix 500 e <RSLX5000>.
- L'**Editor multi-lingue** si trova nel menu **Strumenti**.
 - Aggiungere elementi.
 - Esportare un'immagine del programma LD.
 - Attivare o disattivare l'apertura automatica del **Selettore di variabili** e del **Selettore blocco istruzione**.
- **Casella degli strumenti LD**
 - Aggiungere elementi.
- **Barra degli strumenti Istruzione**
 - Aggiungere elementi.

È possibile sostituire una variabile assegnata direttamente dall'editor di lingue o dal **selettore di variabili**.

Per modificare una variabile dall'editor di lingue:

1. Nell'editor di lingue, fare clic sul nome della variabile per visualizzare un elenco a discesa delle variabili globali e locali.
2. Effettuare una delle seguenti operazioni:
 - Immettere un nome della variabile nella casella di testo:
 - Selezionare un nome variabile diverso dall'elenco a discesa.

Per modificare una variabile dal selettore di variabili:

1. Nell'editor di lingue, fare doppio clic sulla variabile per visualizzare il **selettore di variabili**.
2. Fare clic sul nome della variabile, quindi selezionare una variabile diversa dall'elenco a discesa delle variabili globali e locali.
3. Fare clic su una variabile esistente e immettere valori costanti nella casella di testo visualizzata.

Vedere anche

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

[Esempi di programma ladder diagram \(LD\)](#) a pagina 57

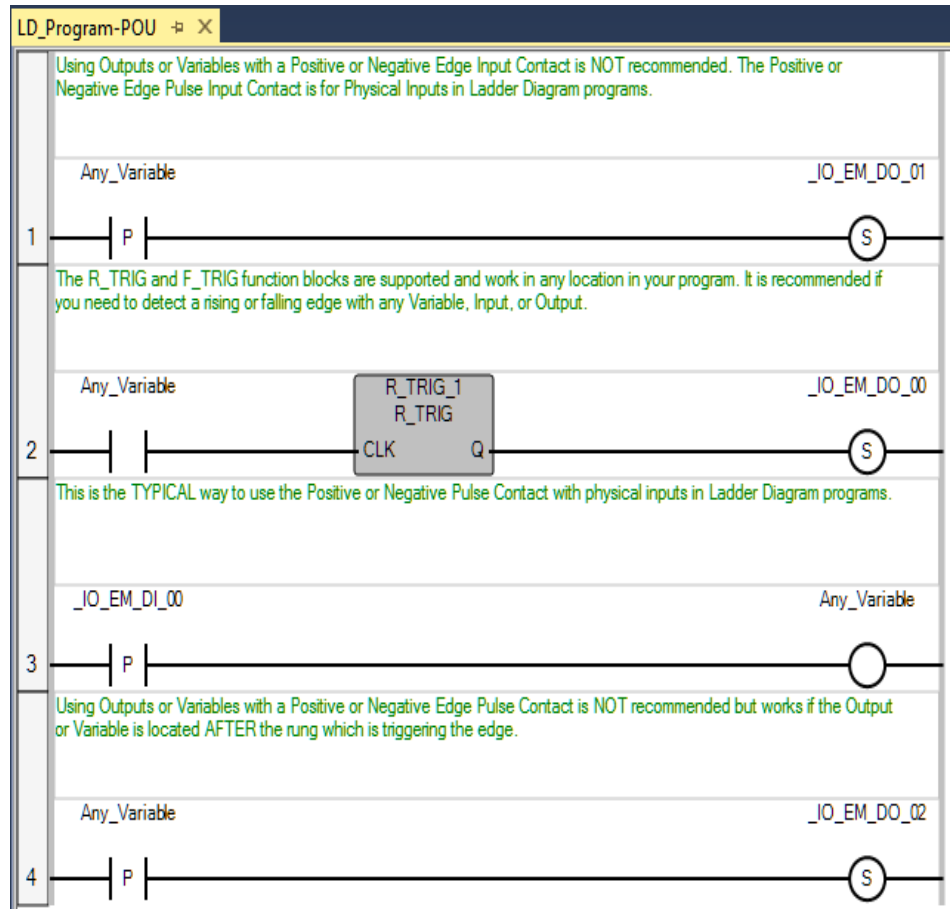
[Tasti di scelta rapida LD](#) a pagina 59

I seguenti esempi sono Programmazioni in Ladder Diagram.

Esempi di programma ladder diagram (LD)

Esempio: blocco funzione R_TRIG

Il programma di esempio seguente mostra l'uso consigliato di un blocco funzione R_TRIG utilizzato per rilevare un bordo durante la connessione con il controllore.



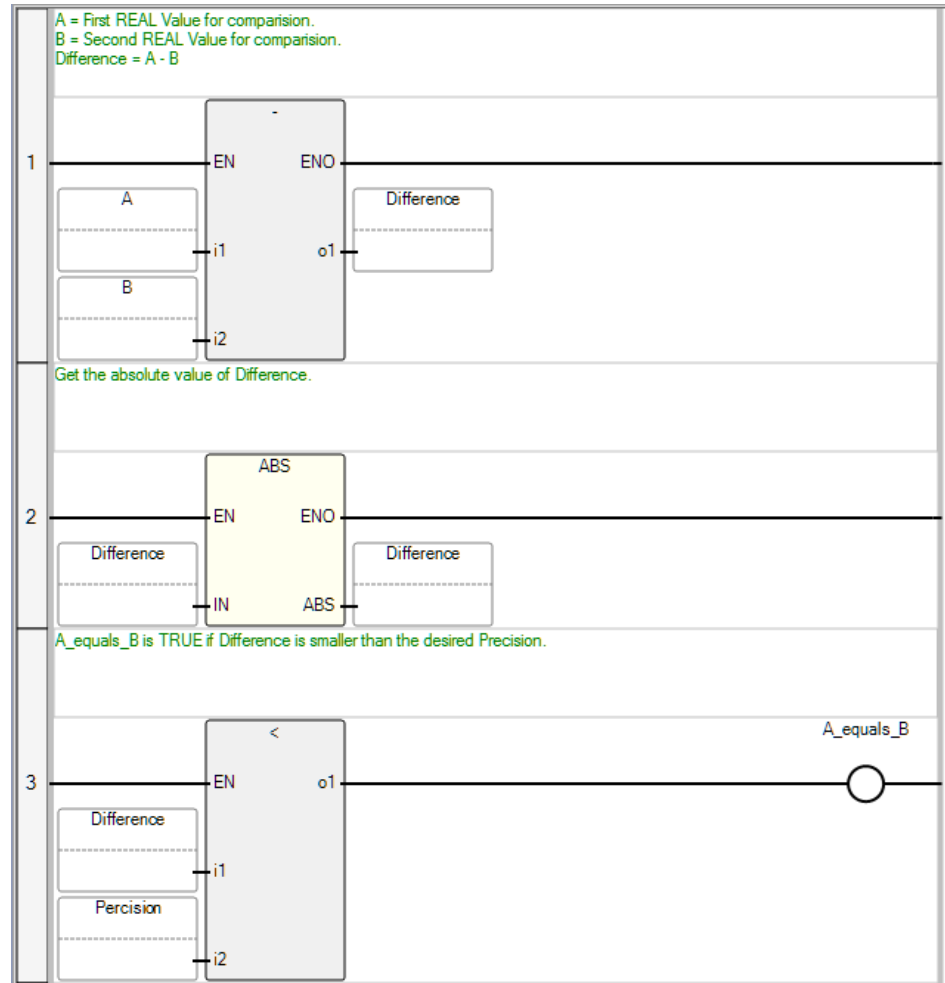
Esempio: confronto dei valori reali mediante sottrazione (-) ABS e minore di (<)

Il tipo di dati Reale non è consigliabile per confrontare l'uguaglianza dei valori a causa delle differenze nell'arrotondamento dei numeri. Due valori di uscita potrebbero sembrare uguali su un display Connected Components Workbench, ma comunque valutati come false.

Ad esempio, 23,500001 e 23,499999 saranno visualizzati entrambi come 23,5 in sul display di ingresso delle variabili, tuttavia nel controllore non saranno uguali.

Per verificare se due dati di tipo reale sono uguali, è possibile usare l'istruzione Sottrazione per ottenere la differenza tra i valori e quindi determinare se la differenza è inferiore a un valore con precisione

stabilita. Vedere l'esempio seguente di programma LD per il confronto tra i valori di due dati di tipo reale.



Vedere anche

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

[Ambiente di sviluppo del programma LD](#) a pagina 26

[Utilizzo degli elementi nell'editor di lingue LD](#) a pagina 56

Tasti di scelta rapida LD

I seguenti tasti di scelta rapida sono disponibili per l'utilizzo con il linguaggio Diagramma ladder:

Tasto di scelta rapida	Descrizione
Ctrl+0	Inserisce un piolo dopo un piolo selezionato. ¹
Ctrl+Alt+0	Inserisce un piolo prima di un piolo selezionato. ¹
Ctrl+1	Inserisce una diramazione dopo un elemento selezionato.
Ctrl+Alt+1	Inserisce una diramazione prima di un elemento selezionato.
Ctrl+2	Inserisce un blocco di istruzioni dopo un elemento selezionato. ²
Ctrl+Alt+2	Inserisce un blocco di istruzioni prima di un elemento selezionato. ²

Tasto di scelta rapida	Descrizione
Ctrl+3	Inserisce un contatto dopo un elemento selezionato. ²
Ctrl+Alt+3	Inserisce un contatto prima di un elemento selezionato. ²
Ctrl+4	Inserisce una bobina dopo un elemento selezionato.
Ctrl+5	Inserisce un salto dopo un elemento selezionato.
Ctrl+Alt+5	Inserisce un salto dopo un elemento selezionato.
Ctrl+6	Inserisce un ritorno dopo un elemento selezionato.
Ctrl+8	Inserisce una diramazione sopra la diramazione selezionata.
Ctrl+Alt+8	Inserisce una diramazione sotto la diramazione selezionata.
Elimina	Rimuove un piolo o un elemento selezionato.
Invio	Se è selezionato un ramo, premendo il tasto INVIO si seleziona il primo elemento del ramo. Se non sono presenti elementi all'interno del ramo, non viene eseguita alcuna operazione.
Barra spaziatrice	Se è selezionato un contatto o una bobina, premere la barra spaziatrice per modificare il tipo di contatto o bobina.
Maiusc+Invio	Inserisce un'interruzione di riga.
Ctrl+Invio	Apri una riga sopra la riga in uso.
Ctrl+Maiusc+Invio	Apri una riga sotto la riga corrente.
Ctrl+Maiusc+L	Rimuove la riga corrente.
Ctrl+Canc	Rimuove la parola successiva nella riga in uso.
Backspace	Rimuove il carattere a sinistra.
Ctrl+Backspace	Rimuove la parola precedente sulla riga in uso.
Ctrl+C	Copia il testo selezionato negli appunti.
Ctrl+Ins	Copia il testo selezionato negli appunti.
Ctrl+V	Incolla nel punto di inserimento il testo salvato negli appunti.
Maiusc+Ins	Incolla nel punto di inserimento il testo salvato negli appunti.
Ctrl+Z	Annula il comando precedente.
Ctrl+Y	Ripete il comando precedente.
Ctrl+Maiusc+Z	Ripete il comando precedente.
Ctrl+Sinistra	Passa all'istruzione o parola precedente.
Ctrl+Destra	Passa all'istruzione o parola successiva.
Home	Passa al primo elemento del ramo selezionato; se non sono presenti elementi all'interno del ramo, non viene eseguita alcuna operazione.
Fine	Passa all'ultimo elemento del ramo selezionato; se non sono presenti elementi all'interno del ramo, non viene eseguita alcuna operazione.
Ctrl+Home	Passa al primo elemento del primo ramo; se non sono presenti elementi all'interno del ramo viene selezionato il primo ramo.
Ctrl+Fine	Passa all'ultimo elemento dell'ultimo ramo, se non sono presenti elementi all'interno del ramo viene selezionato l'ultimo ramo.
Pagina su	Passa in alto nel codice visibile.
Pagina giù	Passa in basso nel codice visibile.
Ctrl+J	Passa alla parentesi corrispondente.
Ctrl+Giù	Scorre verso il basso.
Ctrl+Su	Scorre verso l'alto.
Maiusc+Giù	Seleziona verso il basso.
Maiusc+clic con il pulsante sinistro	Seleziona più rami. Fare clic su ogni ramo singolarmente.
Maiusc+Su	Seleziona verso l'alto.
Maiusc+Sinistra	Seleziona verso sinistra.
Maiusc+Destra	Seleziona verso destra.
Ctrl+Maiusc+Sinistra	Seleziona la parola o istruzione precedente.

Tasto di scelta rapida	Descrizione
Ctrl+Maiusc+Destra	Seleziona la parola o istruzione successiva.
Maiusc+Home	Seleziona dal punto di inserimento all'inizio della riga.
Maiusc+Fine	Seleziona dal punto di inserimento alla fine della riga.
Ctrl+Maiusc+Home	Seleziona dal punto di inserimento all'inizio del documento.
Ctrl+Maiusc+Fine	Seleziona dal punto di inserimento alla fine del documento.
MAIUSC+Pagina su	Seleziona dal punto di inserimento all'inizio del codice visibile.
MAIUSC+Pagina giù	Seleziona dal punto di inserimento alla fine del codice visibile.
CTRL+MAIUSC+Pagina su	Seleziona dal punto di inserimento all'inizio del codice visibile.
CTRL+MAIUSC+Pagina giù	Seleziona dal punto di inserimento alla fine del codice visibile.
Ctrl+A	Seleziona l'intero documento.
Ctrl+D	Se è selezionato un ramo o un elemento del ramo, dopo aver premuto CTRL+D l'utente può modificare il commento del ramo.
Ctrl+R	Attivare o disattivare la chiamata automatica del selettore. Per impostazione predefinita, quando si aggiunge un elemento a una Programmazione in Ladder Diagram viene aperta la finestra di dialogo Selettore blocco istruzione o Selettore di variabili.
Ctrl+Maiusc+W	Seleziona la parola successiva.
Ctrl+Maiusc+J	Seleziona fino alla parentesi corrispondente.
Maiusc+Alt+Giù	Seleziona la riga in uso e le successive.
Maiusc+Alt+Su	Seleziona la riga in uso e le precedenti.
Maiusc+Alt+Sinistra	Seleziona a sinistra sulla riga corrente.
Maiusc+Alt+Destra	Seleziona a destra della riga corrente.
Ctrl+Maiusc+Alt+Sinistra	Seleziona le colonne disponibili nelle righe di codice da sinistra a destra.
Ctrl+Maiusc+Alt+Destra	Seleziona le colonne disponibili nelle righe di codice da destra a sinistra.
Esc	Deseleziona il testo selezionato.
Ins	Attiva/disattiva la modalità di inserimento/sovrascrittura.

¹Se non è selezionato alcun piolo, viene aggiunto un piolo alla fine dell'elenco pioli.

²Se è selezionato un ramo, viene inserito un elemento alla fine del ramo.

Vedere anche

[Linguaggio LD \(Ladder Diagram\)](#) a pagina 25

Istruzione allarme

Utilizzare l'istruzione allarme per generare avvisi quando viene raggiunto un limite superiore o inferiore configurato.

Blocco funzione	Descrizione
LIM_ALARM a pagina 63	Isteresi di un valore reale per limiti superiore e inferiore.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

LIM_ALARM (limite allarme)

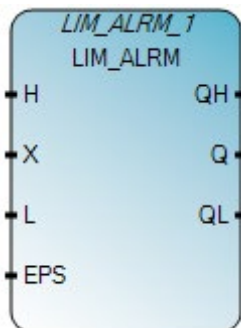
LIM_ALARM è un allarme con isteresi di un valore reale per limiti superiore e inferiore.

L'isteresi si applica ai limiti superiore e inferiore. Il delta isteresi usato per il limite superiore o inferiore è pari alla metà del parametro EPS.

L'allarme di processo si verifica quando il controllore riceve ed elabora un errore. Gli allarmi a livello di processo segnalano che il modulo ha superato i limiti alti o bassi configurati per ciascun canale.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

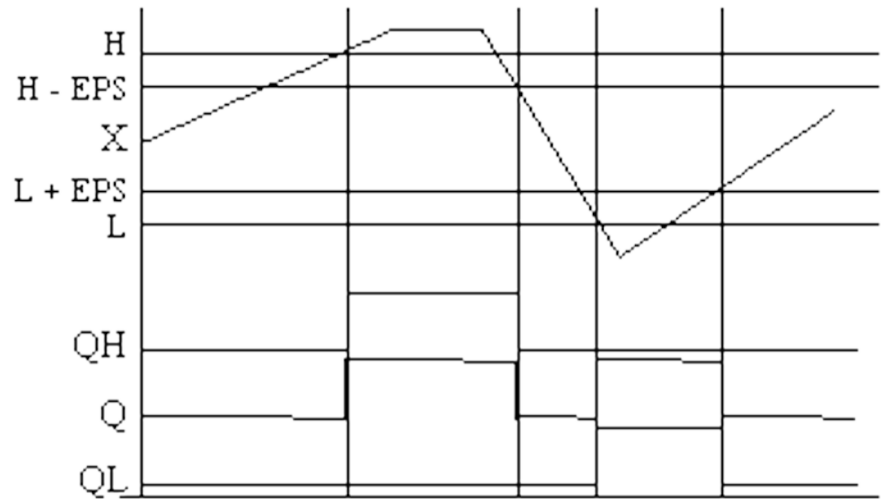


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

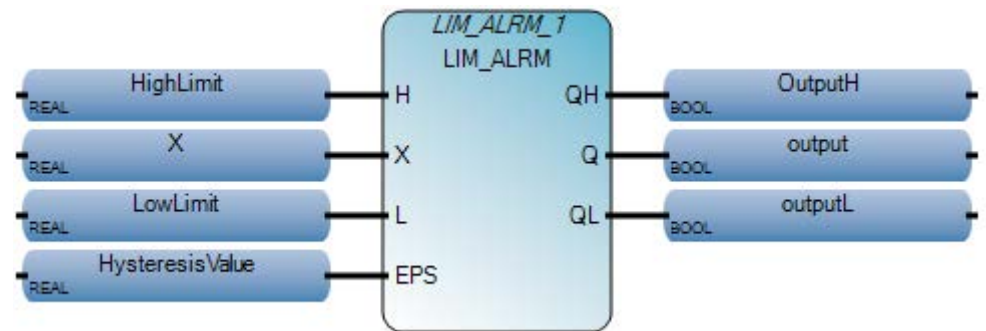
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Se TRUE, abilita il blocco istruzione. TRUE: esegue il calcolo corrente LIM_ALARM. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.

H	Ingresso	REAL	Valore limite superiore.
X	Ingresso	REAL	L'ingresso è qualsiasi valore reale.
L	Ingresso	REAL	Valore limite inferiore.
EPS	Ingresso	REAL	Valore di isteresi (deve essere maggiore di zero).
QH	Uscita	BOOL	Allarme superiore: TRUE se X supera limite superiore H.
Q	Uscita	BOOL	Uscita allarme: TRUE se X è oltre i limiti.
QL	Uscita	BOOL	Allarme inferiore: TRUE se X minore del limite inferiore L.
ENO	Uscita	BOOL	Abilita uscite. Applicabile solo alle programmazioni in Ladder Diagram.

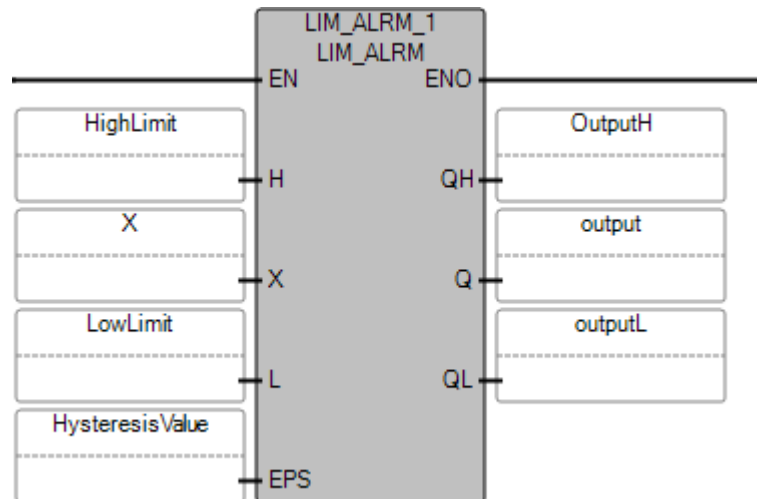
Esempio di diagramma di temporizzazione LIM_ALARM



Esempio di programmazione a blocchi funzionali LIM_ALARM



Esempio di diagramma ladder LIM_ALARM



Esempio di testo strutturato LIM_ALARM

```
LIM_ALARM_1 {
void LIM_ALARM_1(REAL H, REAL X, REAL L, REAL EPS)
Tipo: LIM_ALARM, Allarme di limite superiore/inferiore con isteresi
```

```
1 HighLimit := 10.0;
2 X := 15.0;
3 LowLimit := 5.0;
4 HysteresisValue := 2.0;
5 LIM_ALARM_1(HighLimit, X, LowLimit, HysteresisValue);
6 OutputH := LIM_ALARM_1.QH;
7 OutputL := LIM_ALARM_1.QL;
8 output := LIM_ALARM_1.Q;
```

Risultati

Monitoraggio della variabile

Nome	Alias	Valorelogico	Valorefisico	Valoreiniziale	Blocca	Tip
X		15.0	N/D		<input type="checkbox"/>	REAL
HysteresisValue		2.0	N/D		<input type="checkbox"/>	REAL
LIM_ALARM_1		<input type="checkbox"/>	LIM_A
OutputH		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
output		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
LowLimit		5.0	N/D		<input type="checkbox"/>	REAL
HighLimit		10.0	N/D		<input type="checkbox"/>	REAL

Chiudi

Vedere anche

[Istruzioni allarme](#) a pagina 63

Istruzioni aritmetiche

Usare le istruzioni aritmetiche per eseguire calcoli matematici.

Funzione	Descrizione
ABS a pagina 67	Restituisce il valore assoluto di un valore Real.
ACOS a pagina 69	Calcola l'arcocoseno di un valore Real.
ACOS_LREAL a pagina 71	Calcola l'arcocoseno di un valore Long Real.
Addition a pagina 72	Somma due o più valori Integer, Real, Time o String.
ASIN a pagina 74	Calcola l'arcoseno di un valore Real.
ASIN_LREAL a pagina 76	Calcola l'arcoseno di un valore Long Real.
ATAN a pagina 77	Calcola l'arcotangente di un valore Real.
ATAN_LREAL a pagina 79	Calcola l'arcotangente di un valore Long Real.
COS a pagina 81	Calcola il coseno di un valore Real.
COS_LREAL a pagina 82	Calcola il coseno di un valore Long Real.
Division a pagina 84	Divisione di due valori Integer o Real.
EXPT a pagina 86	Calcola il valore Real di un numero di base elevato alla potenza dell'esponente intero.
LOG a pagina 87	Calcola il logaritmo (base 10) di un valore Real.
MOD a pagina 89	Esegue il calcolo Modulo sui valori Integer.
MOV a pagina 91	Copia un valore di ingresso in un'uscita.
Multiplication a pagina 92	Moltiplica due o più valori Integer o Real.
Neg a pagina 94	Converte un valore in un valore negativo.
POW a pagina 95	Calcola il valore di un numero Real elevato alla potenza dell'esponente Real.
RAND a pagina 97	Calcola valori interi casuali da un intervallo definito.
SIN a pagina 99	Calcola il seno di un valore Real.
SIN_LREAL a pagina 100	Calcola il seno di un valore Long Real.
SORT a pagina 102	Calcola la radice quadrata di un valore Real.
Subtraction a pagina 103	Sottrarre un valore Integer, Real o Time da un altro valore Integer, Real o Time.
TAN a pagina 105	Calcola la tangente di un valore Real.
TAN_LREAL a pagina 107	Calcola la tangente di un valore Long Real.
TRUNC a pagina 108	Tronca valori Real, lasciando solo l'intero.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

ABS (valore assoluto)

Restituisce il valore assoluto (positivo) di un valore Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



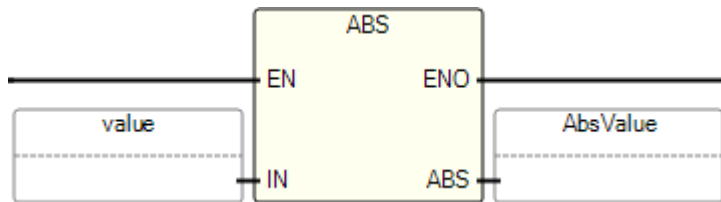
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo assoluto corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Qualunque valore reale con segno.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.
ABS	Uscita	REAL	Valore assoluto (sempre positivo).

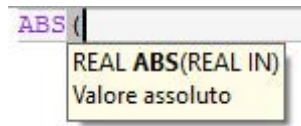
Esempio di programmazione a blocchi funzionali ABS



Esempio di Diagramma ladder ABS



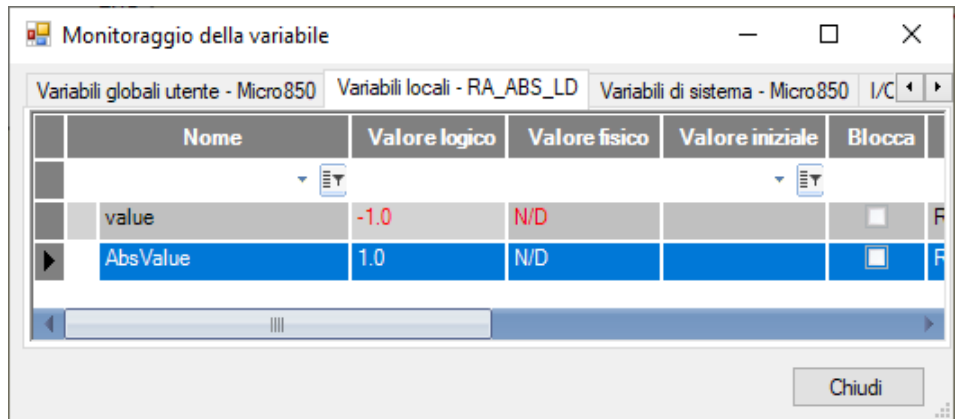
Esempio di diagramma di Testo strutturato ABS



```

1 value := -1.0;
2 AbsValue := ABS (value);

```



(* Equivalenza ST: *)

over := (ABS (delta) > range);

Risultati

Vedere anche

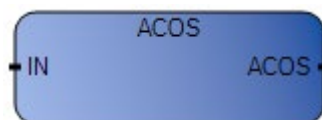
[Istruzioni aritmetiche](#) a pagina 67

ACOS (arcocoseno di origine)

Calcola l'arcocoseno di un valore Real. I valori di input e output sono espressi in radianti.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

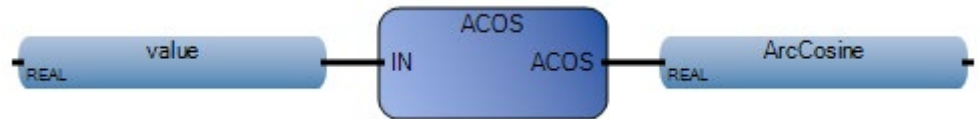
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



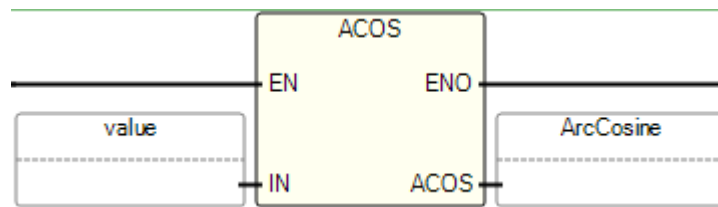
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo dell'arcocoseno corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Deve essere compreso nell'intervallo [-1,0 .. +1,0].
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.
ACOS	Uscita	REAL	Arcoseno del valore di ingresso (nel set [-p1/2..+p1/2])=0 per ingresso non valido.

Esempio di programmazione a blocchi funzionali ACOS



Esempio di Diagramma ladder ACOS

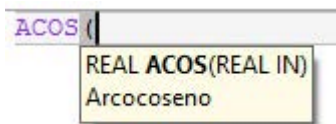


Esempio di Testo strutturato ACOS

```

1 value := 0.5;
2 ArcCosine := ACOS(value);

```

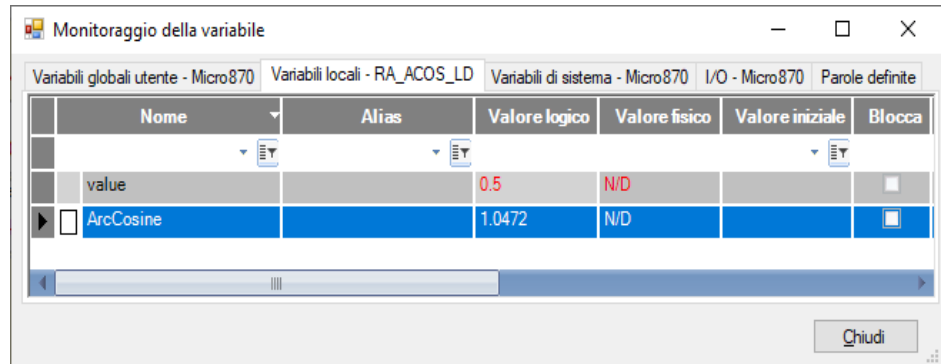


(* Equivalenza ST: *)

coseno := COS (angolo);

risultato := ACOS (coseno); (* il risultato è uguale all'angolo *)

Risultati



Vedere anche

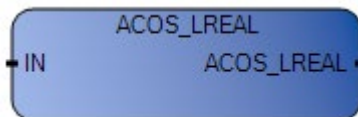
[Istruzioni aritmetiche](#) a pagina 67

ACOS_LREAL (arcocoseno Long Real)

Calcola l'arcocoseno di un valore Long Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

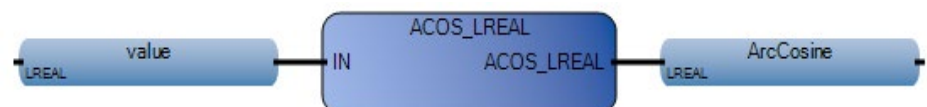
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



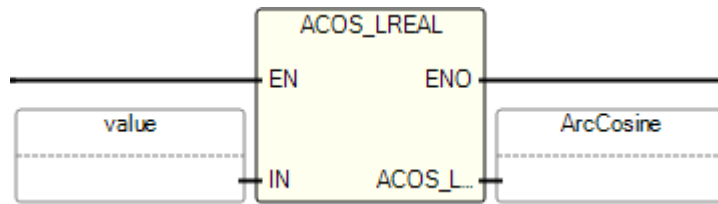
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	LREAL	Deve essere compreso nell'intervallo [-1,0 .. +1,0].
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.
ACOS_LREAL	Uscita	LREAL	Arcocoseno del valore in ingresso (in set [0,0 .. PI]) = 0,0 per ingresso non valido.

Esempio di Programmazione a blocchi funzionali ACOS_LREAL



Esempio di Diagramma ladder ACOS_LREAL



Esempio di Testo strutturato ACOS_LREAL

```
ACOS_LREAL (
    LREAL ACOS_LREAL(LREAL IN)
    Eseguire il calcolo dell'arcocoseno reale a 64 bit.
```

```
1 value := 0.5;
2 ArcCosine := ACOS_LREAL(value);
```

(* Equivalenza ST: *)

coseno := COS_LREAL (angolo);

risultato := ACOS_LREAL (coseno); (* il risultato è uguale all'angolo *)

Risultati

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca
value		0.5	N/D		<input type="checkbox"/>
ArcCosine		1.047197551196	N/D		<input type="checkbox"/>

Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

Addition

Somma due o più valori Integer, Real, Time o String.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo dell'addizione corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME STRING	Addendo in tipo di dati Real, Time o String. Tutti gli ingressi devono essere dello stesso tipo di dati.
i2	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME STRING	Addendo in tipo di dati Real, Time o String. Tutti gli ingressi devono essere dello stesso tipo di dati.

o1	Uscita	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME STRING	Somma dei valori di input in formato Real, Time o String. Input e output devono utilizzare lo stesso tipo di dati.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato Addition

(* Equivalenza ST: *)

```
ao10 := ai101 + ai102;
ao5 := (ai51 + ai52) + ai53;
```

Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

ASIN (arcoseno)

Calcola l'arcoseno di un valore Real. I valori di input e output sono espressi in radianti.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

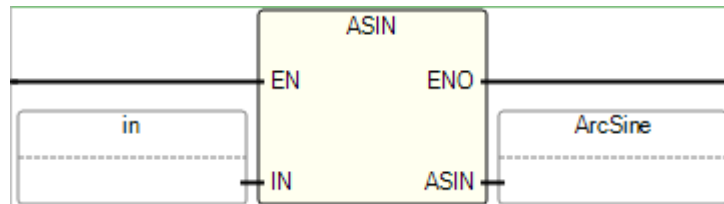
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo dell'arcoseno corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Deve essere compreso nell'intervallo [-1,0 .. +1,0].

ASIN	Uscita	REAL	Arcoseno del valore di ingresso (nel set [-p1/2..+p1/2])=0 per ingresso non valido.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali ASIN



Esempio di Diagramma ladder ASIN



Esempio di Testo strutturato ASIN

```

1 in := 0.5;
2 ArcSine := ASIN(in);

```

ASIN (

REAL ASIN(REAL IN)
Arcoseno

(* Equivalenza ST: *)

sine := SIN (angle);

result := ASIN (sine); (* il risultato è uguale all'angolo *)

Risultati

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dato
in		0.5	N/D		<input type="checkbox"/>	REAL
ArcSine		0.5235988	N/D		<input type="checkbox"/>	REAL

Vedere anche

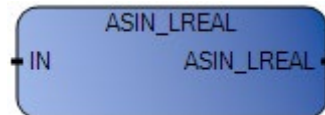
[Istruzioni aritmetiche](#) a pagina 67

ASIN_LREAL (arcoseno Long Real)

Calcola l'arcoseno di un valore Long Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

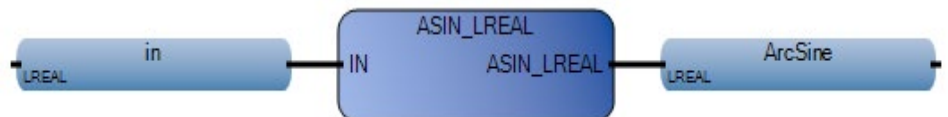
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



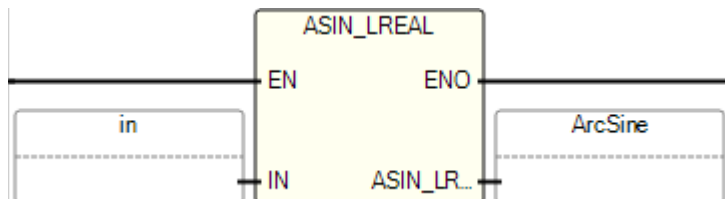
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	LREAL	Deve essere compreso nell'intervallo [-1,0 .. +1,0].
ASIN_LREAL	Uscita	LREAL	Arcoseno del valore in ingresso (in set [-PI/2 .. +PI/2]) = 0,0 per ingresso non valido.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali ASIN_LREAL



Esempio di diagramma ladder ASIN_LREAL



Esempio di testo strutturato ASIN_LREAL

```

1 | in := 0.5;
2 | ArcSine := ASIN_LREAL(in);
    
```

ASIN_LREAL (

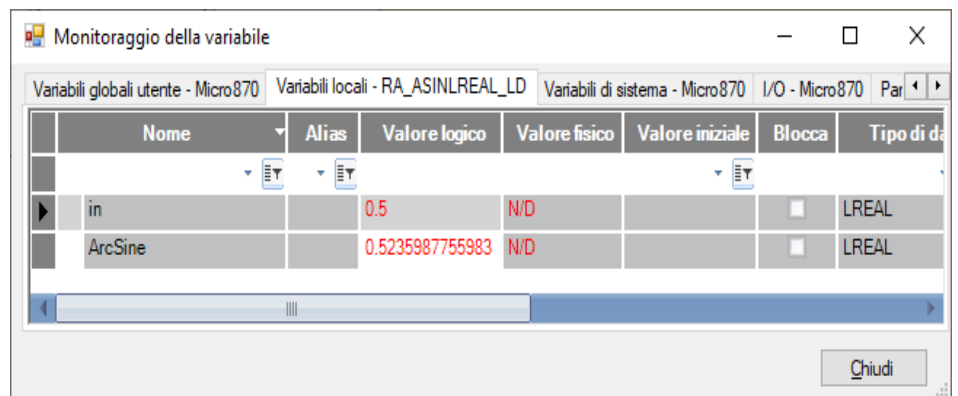
LREAL ASIN_LREAL(LREAL IN)
Eeguire il calcolo dell'arcoseno reale a 64 bit.

(* Equivalenza ST: *)

seno := SIN_LREAL (angolo);

risultato := ASIN_LREAL (seno); (* il risultato è uguale all'angolo *)

Risultati



Vedere anche

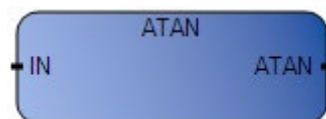
[Istruzioni aritmetiche](#) a pagina 67

ATAN (arcotangente)

Calcola l'arcotangente di un valore Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

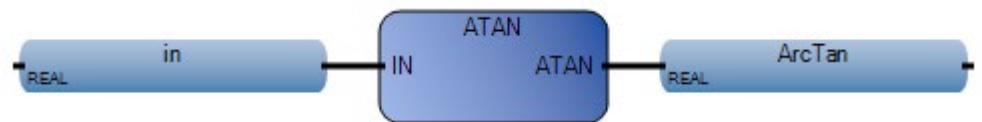


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

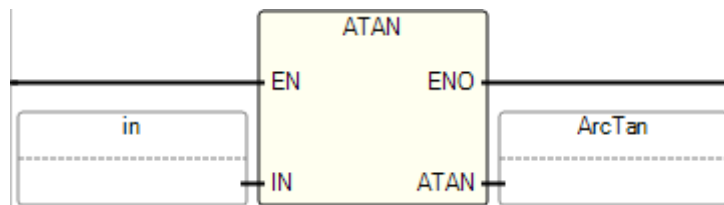
Parametro	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo dell'arcotangente corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Qualsiasi valore Real.
ATAN	Uscita	REAL	Arcotangente del valore di ingresso (nel set $[-\pi/2 .. +\pi/2]$) = 0,0 per ingresso non valido.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali ATAN



Esempio di Diagramma ladder ATAN



Esempio di Testo strutturato ATAN

```

1 in := 0.5;
2 ArcTan := ATAN(in);

```

ATAN (

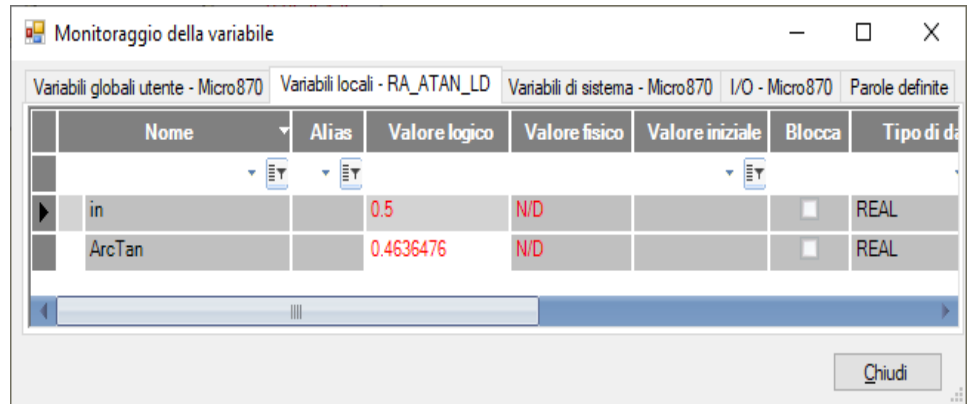
REAL ATAN(REAL IN)
Arcotangente

(* Equivalenza ST: *)

tangent := TAN (angle);

result := ATAN (tangent); (* il risultato è uguale all'angolo*)

Risultati



Vedere anche

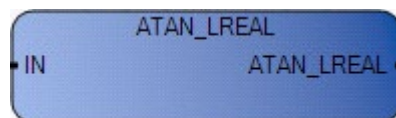
[Istruzioni aritmetiche](#) a pagina 67

ATAN_LREAL (arcotangente Long Real)

Calcola l'arcotangente di un valore Long Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

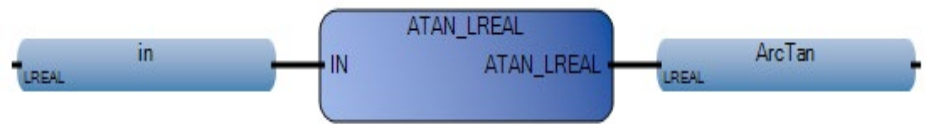
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



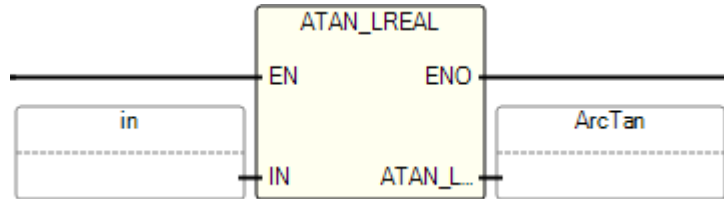
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	LREAL	Qualsiasi valore Long Real.
ATAN_LREAL	Uscita	LREAL	Arcotangente del valore di ingresso (nel set $[-\pi/2 .. +\pi/2]$) = 0,0 per ingresso non valido.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di diagramma blocco funzione ATAN_LREAL



Esempio di diagramma ladder ATAN_LREAL



Esempio di testo strutturato ATAN_LREAL

```

1 | in := 0.5;
2 | ArcTan := ATAN_LREAL(in);

```

ATAN_LREAL (

LREAL ATAN_LREAL(LREAL IN)
 Eseguire il calcolo dell'arcotangente reale a 64 bit.

(* Equivalenza ST: *)

tangente := TAN_LREAL (angolo);

risultato := ATAN_LREAL (tangente); (* il risultato è uguale all'angolo *)

Risultati

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dato
in		0.5	N/D		<input type="checkbox"/>	LREAL
ArcTan		0.4636476090006	N/D		<input type="checkbox"/>	LREAL

Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

COS (coseno)

Calcola il coseno di un valore Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

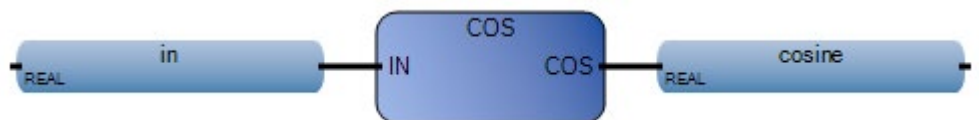
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



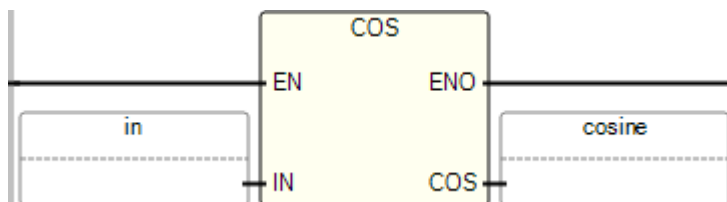
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo del coseno corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Qualsiasi valore Real.
COS	Uscita	REAL	Coseno del valore di input (nel set [-1,0 .. +1,0]).
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali COS



Esempio di Diagramma ladder COS



Esempio di Testo strutturato COS

```
COS (
  REAL COS(REAL IN)
  Coseno
```

```
1 | in := 10.0;
2 | cosine := COS(in);
```

(* Equivalenza ST: *)

coseno := COS (angolo);

risultato := ACOS (coseno); (* il risultato è uguale all'angolo *)

Risultati

Name	Logical Value	Physical Value	Initial Value	Lock	Data Type
in	10.0	N/A		<input type="checkbox"/>	REAL
Cosine	-0.8390715	N/A		<input checked="" type="checkbox"/>	REAL

Vedere anche

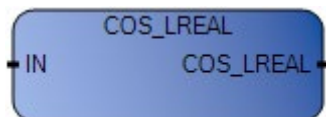
[Istruzioni aritmetiche](#) a pagina 67

COS_LREAL (coseno Long Real)

Calcola il coseno di un valore Long Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

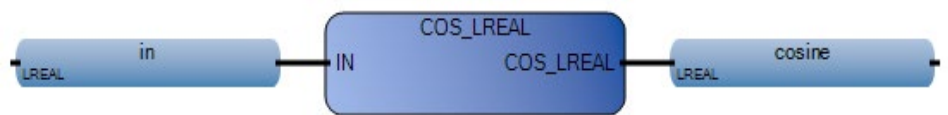
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



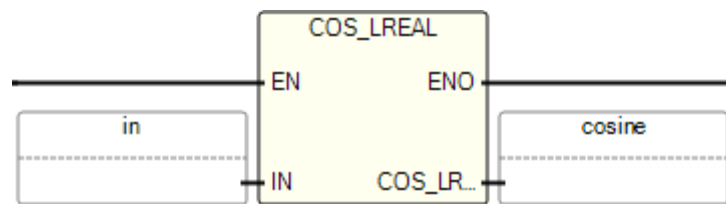
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo del coseno corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	LREAL	Qualsiasi valore Long Real.
COS_LREAL	Uscita	LREAL	Coseno del valore di input (nel set [-1,0 .. +1,0]).
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali COS_LREAL



Esempio di diagramma ladder COS_LREAL



Esempio di testo strutturato COS_LREAL

```
COS_LREAL (
  LREAL COS_LREAL(LREAL IN)
  Eseguire il calcolo del coseno reale a 64 bit.
```

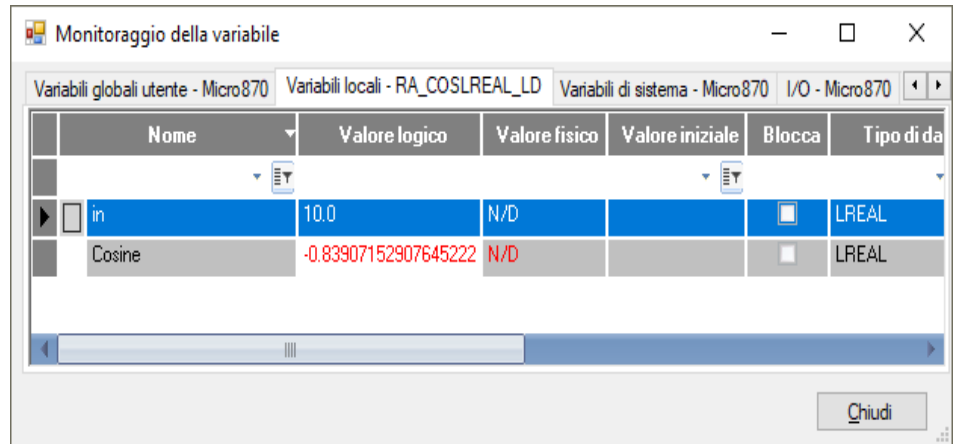
```
1 | in := 10.0;
2 | cosine := COS_LREAL(in);
```

(* Equivalenza ST: *)

coseno := COS_LREAL (angolo);

risultato := ACOS_LREAL (coseno); (* il risultato è uguale all'angolo *)

Risultati



Vedere anche

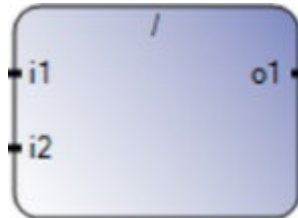
[Istruzioni aritmetiche](#) a pagina 67

Division

Divide il primo valore di input Integer o Real per il secondo valore di input Integer o Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della divisione corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.

i1	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL	dividendo nel tipo di dati Integer o Real diverso da zero. Tutti gli ingressi devono essere dello stesso tipo di dati.
i2	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL	Divisore nel tipo di dati Integer o Real diverso da zero. Tutti gli ingressi devono essere dello stesso tipo di dati.
o1	Uscita	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL	Quoziente degli input nel tipo di dati Integer o Real diverso da zero. Input e output devono utilizzare lo stesso tipo di dati.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato DIVISION

(* Equivalenza ST: *)

```
ao10 := ai101 / ai102;
ao5  := (ai5 / 2) / ai53;
```

Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

EXPT (esponente)

Aumenta il valore di IN (base) alla potenza di EXP (esponente) e restituisce il risultato Real dell'operazione.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



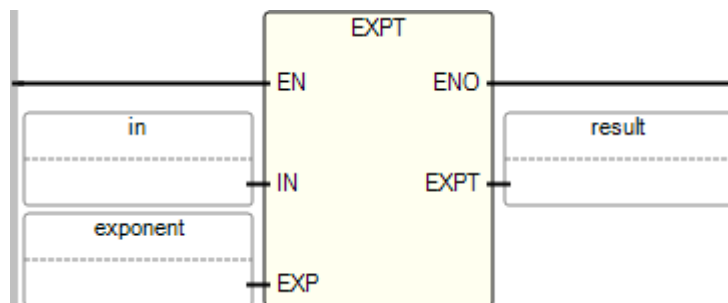
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo dell'esponente corrente. FALSE: il calcolo non avviene.
IN	Ingresso	REAL	Qualunque valore reale con segno.
EXP	Ingresso	DINT	Esponente intero.
EXPT	Uscita	REAL	Il valore Real di IN alla potenza di EXP.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali EXPT



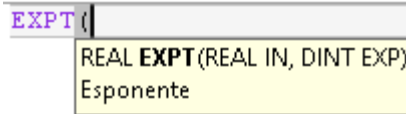
Esempio di Diagramma ladder EXPT



Esempio di Testo strutturato EXPT

```

1  in := 2.0;
2  exponent := 3;
3  result := EXPT(in, exponent);
    
```



(* Equivalenza ST: *)

dimensioni_tb := ANY_TO_DINT (EXPT (2,0; intervallo));

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
exponent	3	N/D		<input type="checkbox"/>	DINT
in	2.0	N/D		<input type="checkbox"/>	REAL
result	8.0	N/D		<input type="checkbox"/>	REAL

Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

LOG (log base 10)

Calcola il logaritmo (base 10) di un valore Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



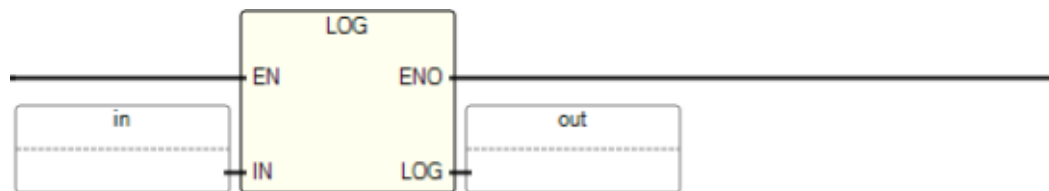
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo del logaritmo corrente. FALSE: il calcolo non avviene.
IN	Ingresso	REAL	Deve essere maggiore di zero.
LOG	Uscita	REAL	Logaritmo (base 10) del valore in ingresso. Il valore restituito è -3.4E+38 per valori IN pari a zero e negativi.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali LOG



Esempio di Diagramma ladder LOG

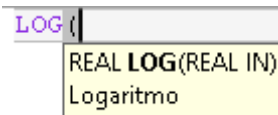


Esempio di Testo strutturato LOG

```

1 in := 10.0;
2 output := LOG(in);

```

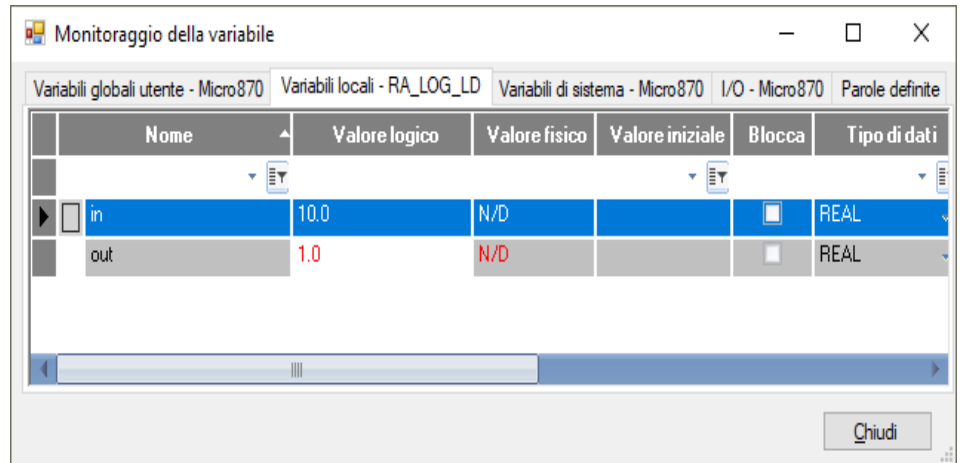


(* Equivalenza ST: *)

xpos := ABS (xval);

xlog := LOG (xpos);

Risultati



Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

MOD (modulo)

Divide l'ingresso IN per l'ingresso Base e porta il resto nell'uscita MOD.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



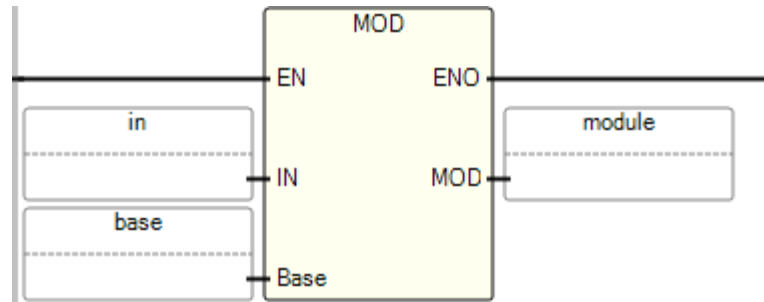
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo del modulo. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	DINT	Qualunque valore intero con segno.
Base	Ingresso	DINT	Deve essere maggiore di zero.
MOD	Uscita	DINT	Calcolo del modulo (input MOD base) / risultato -1 se base <= 0.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali MOD



Esempio di Diagramma ladder MOD



Esempio di Testo strutturato MOD

```
MOD (
  DINT MOD(DINT IN, DINT Base)
  Modulo
```

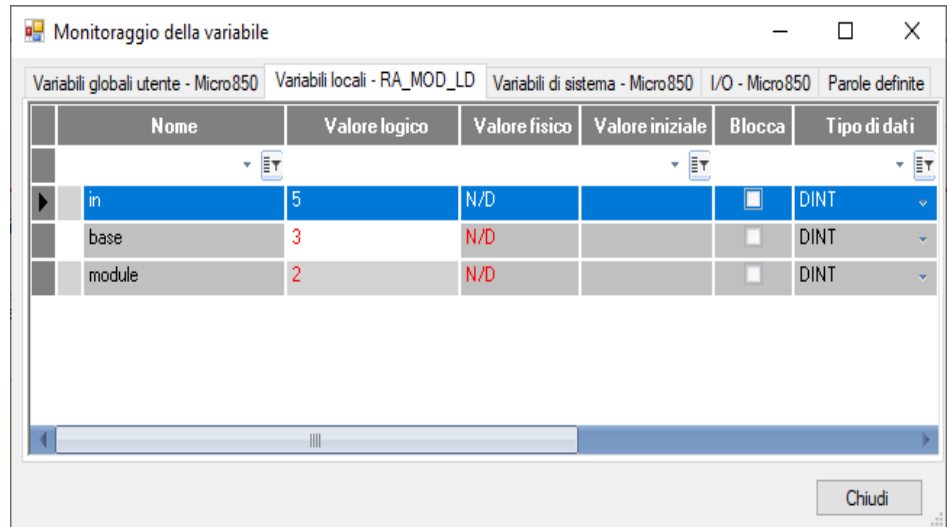
```
1 in := 5;
2 base := 3;
3 module := MOD(in, base);
```

(* Equivalenza ST: *)

risultato_divisione := (valore / divisore); (* divisione intera *)

resto_divisione := MOD (valore, divisore); (* resto della divisione *)

Risultati



Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

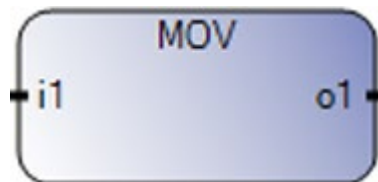
MOV (sposta)

Assegna il valore di ingresso (i1) a un'uscita (o1).

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Per i programmi Testo strutturato, utilizzare l'operatore Equal (=) anziché MOV.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue collegamento diretto a un calcolo di uscita. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.

i1	Ingresso	BOOL DINT REAL TIME STRING SINT USINT INT UINT UDINT LINT ULINT DATE LREAL BYTE WORD DWORD LWORD	Input e output devono utilizzare lo stesso tipo di dati.
o1	Uscita	BOOL DINT REAL TIME STRING SINT USINT INT UINT UDINT LINT ULINT DATE LREAL BYTE WORD DWORD LWORD	Input e output devono utilizzare lo stesso tipo di dati.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Testo strutturato

(* Equivalenza ST: *)

```
ao23 := ai10;
```

Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

Multiplication

Moltiplica due o più valori Integer o Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della moltiplicazione corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL	Fattore in tipo di dati Integer o Real. Tutti gli ingressi devono essere dello stesso tipo di dati.
i2	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT LINT LWORD REAL LREAL	Fattore in tipo di dati Integer o Real. Tutti gli ingressi devono essere dello stesso tipo di dati.

o1	Uscita	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL	Prodotto degli ingressi nel tipo di dati Integer o Real. Input e output devono utilizzare lo stesso tipo di dati.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Testo strutturato Multiplication

(* Equivalenza ST *)

```
ao10 := ai101 * ai102;
ao5 := (ai51 * ai52) * ai53;
```

Vedere anche

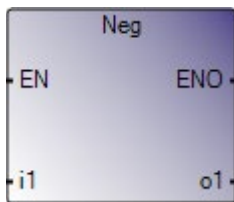
[Istruzioni aritmetiche](#) a pagina 67

Neg (negazione)

Converte un valore in un valore negato.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione corrente in negativo. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.

i1	Ingresso	SINT INT DINT LINT REAL LREAL	Input e output devono essere dello stesso tipo di dati.
o1	Uscita	SINT INT DINT LINT REAL LREAL	Input e output devono essere dello stesso tipo di dati.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato Neg

(* Equivalenza ST: *)

```
ao23 := - (ai10);
ro100 := - (ril + ri2);
```

Vedere anche

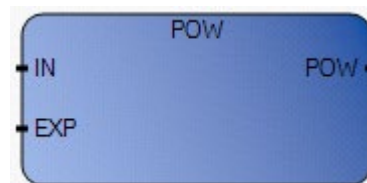
[Istruzioni aritmetiche](#) a pagina 67

POW (incremento alimentazione)

Quando il primo argomento è 'base' e il secondo è 'exponent', calcolare il risultato Real di (base exponent):

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

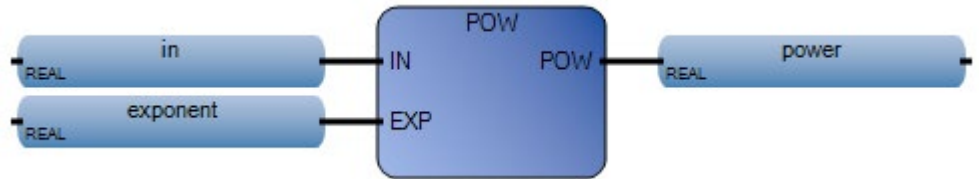


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

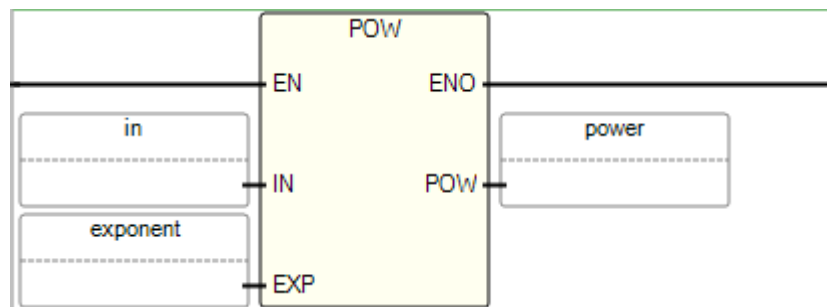
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo dell'esponente corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Numero Real da elevare.
EXP	Ingresso	REAL	Potenza (esponente).

POW	Uscita	REAL	(IN EXP) 1,0 se IN è diverso da 0,0 e EXP è 0,0 0,0 se IN è 0,0 mentre EXP è negativo 0,0 se IN ed EXP sono 0,0 0,0 se IN è negativo ed EXP non corrisponde a un numero intero.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali POW



Esempio di Diagramma ladder POW



Esempio di testo strutturato POW

```

POW (
    REAL POW(REAL IN, REAL EXP)
    Calcolo alimentazione

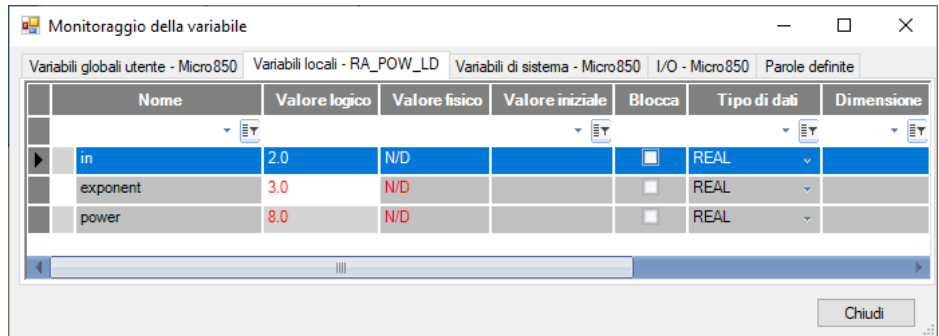
1   in := 2.0;
2   exponent := 3.0;
3   power := POW(in, exponent);

```

(* Equivalenza ST: *)

result := POW (xval, power);

Risultati



Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

RAND (valore casuale)

Calcola valori interi casuali da un intervallo definito.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

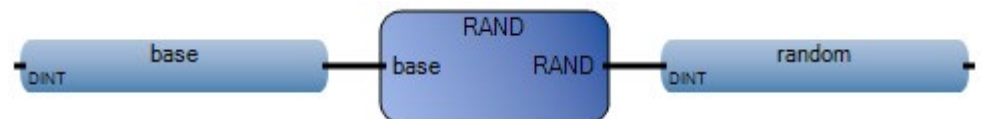
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



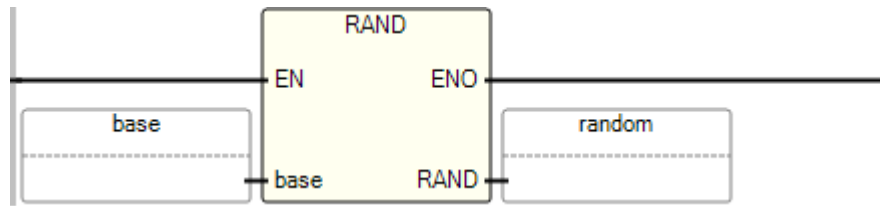
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo del valore intero casuale. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
base	Ingresso	DINT	Definisce il set di numeri supportato.
RAND	Uscita	DINT	Valore casuale nel set [0..base-1].
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali RAND



Esempio di Diagramma ladder RAND



Esempio di Testo strutturato RAND

```
RAND (
  DINT RAND(DINT base)
  Valore random
```

```
1 base := 10;
2 random := RAND (base);
```

(* Equivalenza ST: *)

selected := MUX4 (RAND (4), 1, 4, 8, 16);

(*

selezione casuale di 1 di 4 valori predefiniti

il valore emesso della chiamata RAND è nel set [0..3],

quindi 'selected' emesso da MUX4, ottiene il valore 'randomly'

1 se 0 è emesso da RAND,

o 4 se 1 è emesso da RAND,

o 8 se 2 è emesso da RAND,

o 16 se 3 è emesso da RAND,

*)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
base	10	N/D		<input checked="" type="checkbox"/>	DINT	
random	6	N/D		<input type="checkbox"/>	DINT	

Vedere anche

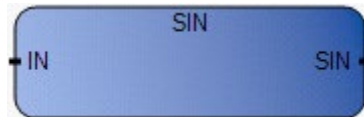
[Istruzioni aritmetiche](#) a pagina 67

SIN (seno)

Calcola il seno di un valore Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



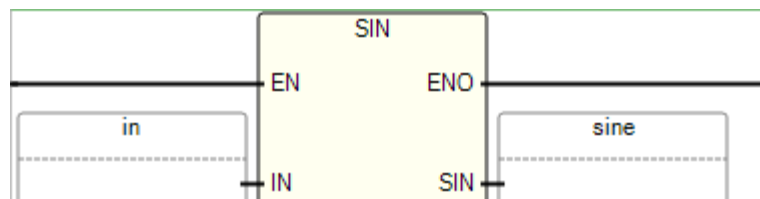
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo del seno corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Qualsiasi valore Real.
SIN	Uscita	REAL	Seno del valore di input (nel set [-1,0 .. +1,0]).
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali SIN



Esempio di Diagramma ladder SIN



Esempio di Testo strutturato SIN

```
SIN (
  REAL SIN(REAL IN)
  Seno
```

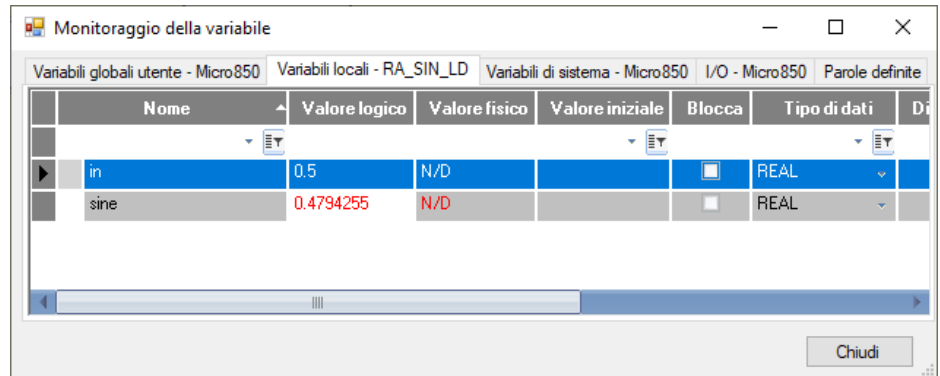
```
1 | in := 0.5;
2 | sine := SIN(in);
```

(* Equivalenza ST: *)

sine := SIN (angle);

result := ASIN (sine); (* il risultato è uguale all'angolo *)

Risultati



Vedere anche

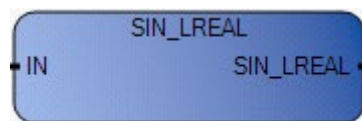
[Istruzioni aritmetiche](#) a pagina 67

SIN_LREAL (seno Long Real)

Calcola il seno di un valore Long Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

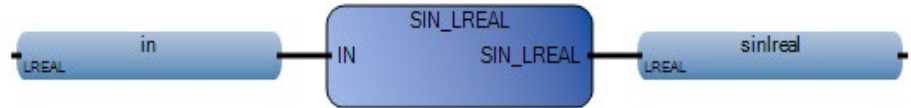


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

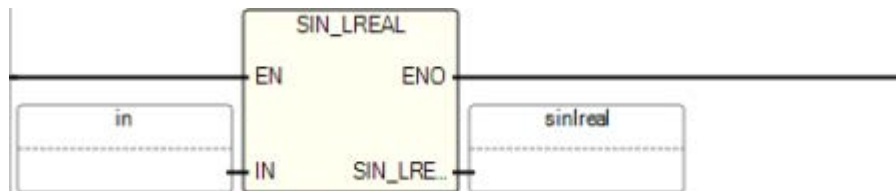
Parametro	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	LREAL	Qualsiasi valore Long Real.
SIN_LREAL	Uscita	LREAL	Seno del valore di input (nel set [-1,0 .. +1,0]).
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali SIN_LREAL



Esempio di diagramma ladder SIN_LREAL



Esempio di testo strutturato SIN_LREAL

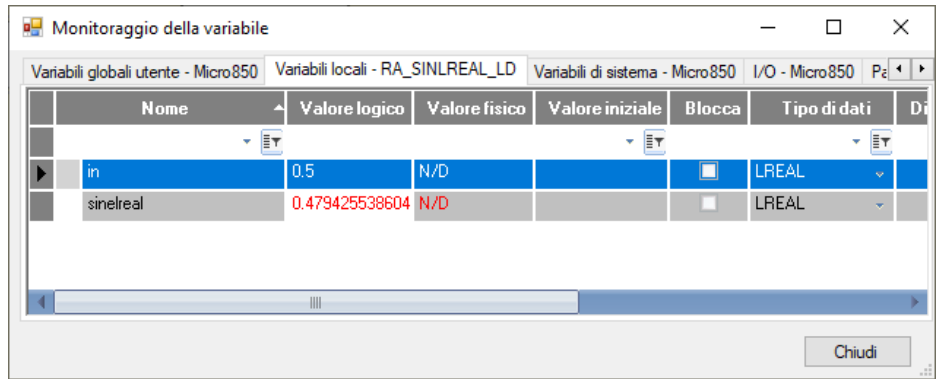
```
ASIN_LREAL {
  LREAL ASIN_LREAL(LREAL IN)
  Eseguire il calcolo dell'arcoseno reale a 64 bit.
```

```
1 | in := 0.5;
2 | sinreal := SIN_LREAL(in);
```

(* Equivalenza ST: *)

```
TESTOUTPUT1 := SIN_LREAL(TESTINPUT1);
```

Risultati



Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

SQRT (radice quadrata)

Calcola la radice quadrata di un valore Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

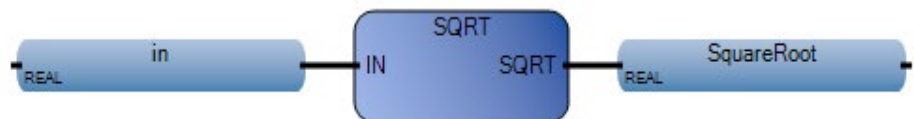
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



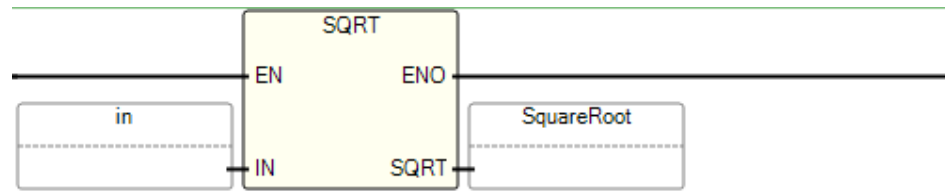
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della radice quadrata corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Deve essere maggiore o uguale a zero.
SQRT	Uscita	REAL	Radice quadrata del valore di input. Il risultato restituito è 0 per un valore IN negativo.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali SQRT



Esempio di diagramma ladder SQRT

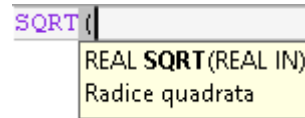


Esempio di Testo strutturato SQRT

```

1 | in := 16.0;
2 | SquareRoot := SQRT(in);

```



(* Equivalenza ST: *)

xpos := ABS (xval);

xroot := SQRT (xpos);

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
in	16.0	N/D		<input type="checkbox"/>	REAL
SquareRoot	4.0	N/D		<input type="checkbox"/>	REAL

Vedere anche

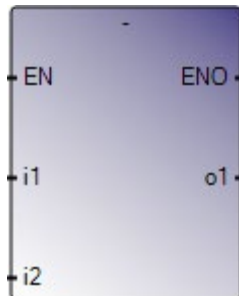
[Istruzioni aritmetiche](#) a pagina 67

Subtraction

Sottrae un valore Integer, Real o Time da un altro valore Integer, Real o Time.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo dell'addizione corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME	Minuendo in qualsiasi tipo di dati Integer, Real o Time. Tutti gli ingressi devono essere dello stesso tipo di dati.
i2	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME	Sottraendo in qualsiasi tipo di dati Integer, Real o Time. Tutti gli ingressi devono essere dello stesso tipo di dati.

o1	Uscita	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME	Differenza tra il minuendo e il sottraendo in qualsiasi tipo di dati Integer, Real o Time. L'output deve essere dello stesso tipo di dati degli input.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato Sottrazione

(* Equivalenza ST: *)

```
ao10 := ai101 - ai102;
ao5 := (ai51 - 1) - ai53;
```

Vedere anche

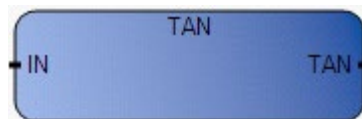
[Istruzioni aritmetiche](#) a pagina 67

TAN (tangente)

Calcola la tangente di un valore Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

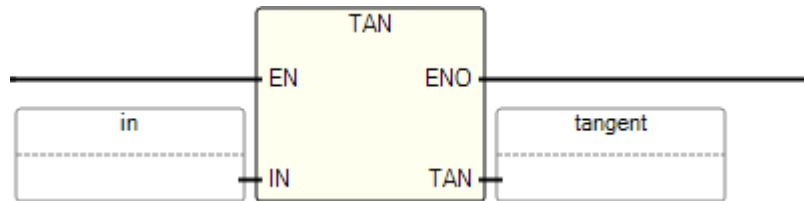
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della tangente corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Non può essere uguale a PI/2 modulo PI.
TAN	Uscita	REAL	Tangente del valore in ingresso = 1E+38 per ingresso valido.

ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.
-----	--------	------	---

Esempio di programmazione a blocchi funzionali TAN



Esempio di Diagramma ladder TAN



Esempio di Testo strutturato TAN

```
TAN(
REAL TAN(REAL IN)
Tangente
```

```
1 in := 0.5;
2 tangent := TAN(in);
```

(* Equivalenza ST: *)

tangent := TAN (angle);

result := ATAN (tangent); (* il risultato è uguale all'angolo*)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
in	0.5	N/D		<input type="checkbox"/>	REAL
tangent	0.5463025	N/D		<input type="checkbox"/>	REAL

Vedere anche

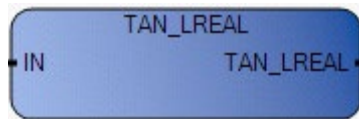
[Istruzioni aritmetiche](#) a pagina 67

TAN_LREAL (tangente Long Real)

Calcola la tangente di un valore Long Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

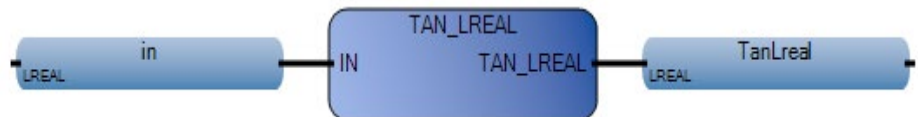
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



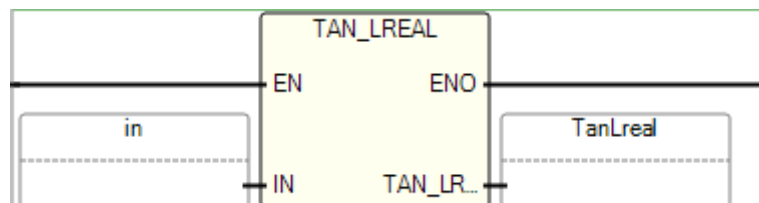
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	LREAL	Non può essere uguale a PI/2 modulo PI.
TAN_LREAL	Uscita	LREAL	Tangente del valore in ingresso = 1E+38 per ingresso valido.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali TAN_LREAL



Esempio di diagramma ladder TAN_LREAL



Esempio di testo strutturato TAN_LREAL

```
TAN_LREAL (
    LREAL TAN_LREAL(LREAL IN)
    Eseguire il calcolo della tangente reale a 64 bit.
```

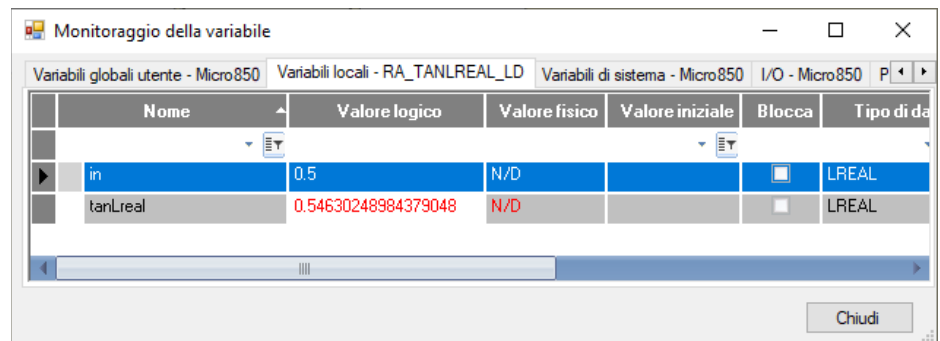
```
1 in := 0.5;
2 TanLreal := TAN_LREAL(in);
```

(* Equivalenza ST: *)

tangente := TAN_LREAL (angolo);

risultato := ATAN_LREAL (tangente); (* il risultato è uguale all'angolo *)

Risultati



Vedere anche

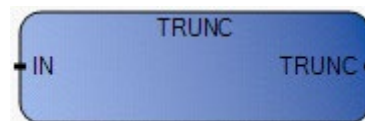
[Istruzioni aritmetiche](#) a pagina 67

TRUNC (troncamento)

Tronca valori Real, lasciando solo l'intero.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

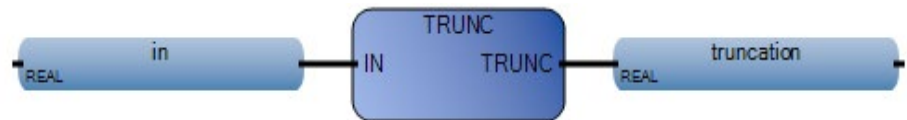


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

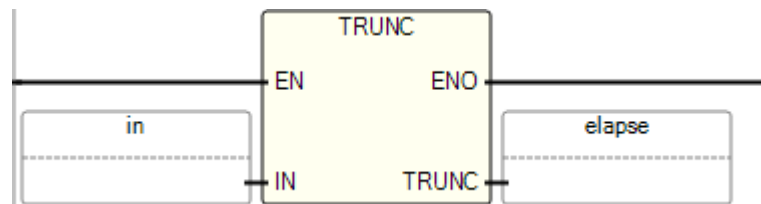
Parametro	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo del troncamento del valore Real. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	REAL	Qualsiasi valore Real.
TRUNC	Uscita	REAL	Se IN>0, il numero intero più grande minore o uguale all'ingresso. Se IN<0, il numero intero più piccolo maggiore o uguale all'ingresso.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali TRUNC



Esempio di Diagramma ladder TRUNC



Esempio di Testo strutturato TRUNC

```
TRUNC {
  REAL TRUNC(REAL IN)
  Tronca parte decimale
```

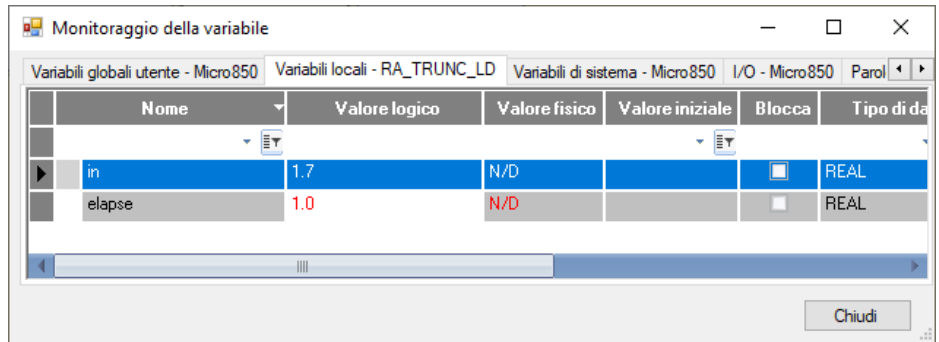
```
1 in := 1.7;
2 truncation := TRUNC(in);
```

(* Equivalenza ST: *)

result := TRUNC (+2.67) + TRUNC (-2.0891);

(* significa risultato := 2,0 + (-2,0) := 0,0; *)

Risultati



The screenshot shows a window titled "Monitoraggio della variabile" with a tabbed interface. The active tab is "Variabili locali - RA_TRUNC_LD". Below the tabs is a table with the following columns: "Nome", "Valore logico", "Valore fisico", "Valore iniziale", "Blocca", and "Tipo di da". Two rows are visible: "in" with a logical value of 1.7 and a physical value of N/D, and "elapsed" with a logical value of 1.0 and a physical value of N/D. A "Chiudi" button is located at the bottom right of the window.

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di da
in	1.7	N/D		<input type="checkbox"/>	REAL
elapsed	1.0	N/D		<input type="checkbox"/>	REAL

Vedere anche

[Istruzioni aritmetiche](#) a pagina 67

Istruzioni porta seriale ASCII

Utilizza le istruzioni della porta seriale ASCII per utilizzare o modificare il canale di comunicazione per la ricezione o la trasmissione dei dati.

Blocco funzione	Descrizione
ABL a pagina 111	Conta il numero di caratteri nel buffer fino al carattere di fine riga incluso.
ACB a pagina 118	Conta il numero totale dei caratteri nel buffer.
ACL a pagina 113	Elimina i buffer di ricezione e trasmissione.
AHL a pagina 115	Impostare o reimpostare linee di arbitraggio modem.
ARD a pagina 120	Legge i caratteri dal buffer di ingresso e posiziona gli stessi in una stringa.
ARL a pagina 123	Legge una riga di caratteri dal buffer di ingresso e posiziona gli stessi in una stringa.
AWA a pagina 125	Scrivere una stringa con due caratteri aggiunti (configurati dall'utente) su un dispositivo esterno.
AWT a pagina 128	Scrive i caratteri da una stringa sorgente a un dispositivo esterno.

Vedere anche

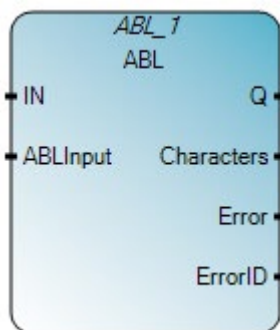
[Istruzione impostate in ordine alfabetico](#) a pagina 18

ABL (prova ASCII per riga del buffer)

Conta il numero dei caratteri ASCII nel buffer di ingresso fino al carattere di terminazione incluso.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

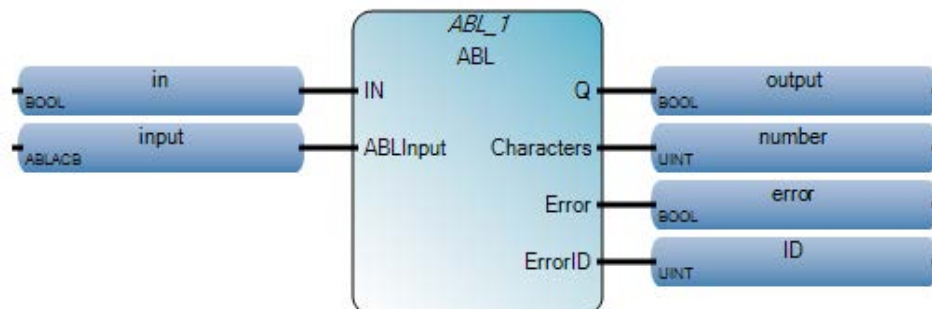
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco funzione ammesso che l'ultima operazione sia stata completata. FALSE: il blocco istruzione non è attivo.
ABLInput	Ingresso	ABLACB	Canale su cui operare. Utilizzare il tipo di dati ABLACB per definire i parametri Canale, Tipo di trigger e Annulla per l'ingresso ABL.
Q	Uscita	BOOL	Indica quando il conteggio dei caratteri è in corso o è stato completato. Le uscite vengono aggiornate in modo asincrono dalla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: il blocco funzione è completo. FALSE: il blocco funzione non è completo.
Characters	Uscita	UINT	Il numero di caratteri nel buffer. Il limite di buffer è di 82 caratteri.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore ABL.

Codici di errore ABL

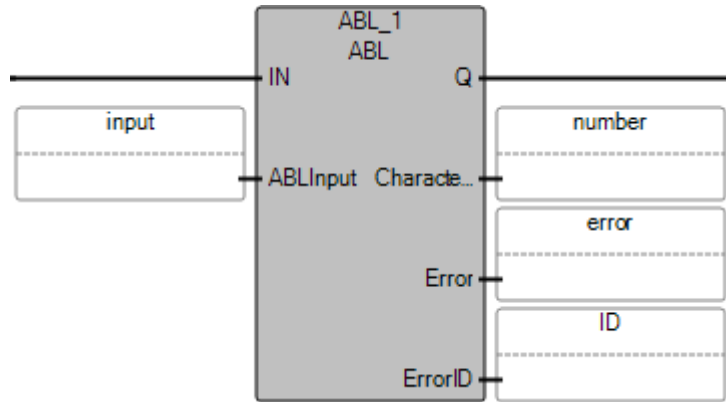
Utilizzare questa tabella per determinare i codici di errore ABL e le descrizioni.

Codice errore	Descrizione errore
03	Impossibile completare la trasmissione perché il segnale Clear-to-Send è stato perso.
06	È stato rilevato un parametro non valido.
07	Impossibile completare la ricezione o la trasmissione ASCII perché la configurazione del canale è stata disattivata nella finestra di dialogo di configurazione del canale.
08	Impossibile completare la scrittura ASCII perché è in corso un'altra trasmissione ASCII.
09	La richiesta di comunicazione ASCII non è supportata dalla configurazione del canale in uso.
10	L'esecuzione dell'istruzione è stata arrestata in seguito all'invio del comando Annulla. Nessuna azione richiesta.
11	La lunghezza richiesta per la stringa non è valida o è un numero negativo, maggiore di 82 o uguale a 0. Si applica ai blocchi funzione ARD e ARL.
13	Il valore richiesto (LEN) nel blocco di funzione è un numero negativo o un valore maggiore della dimensione della stringa memorizzata con la stringa sorgente. Si applica ai blocchi funzione AWA e AWT.
14	Il blocco funzione ACL è stato interrotto.
16	Le linee di controllo RTS o CTS non sono supportate dalla porta seriale.

Esempio di programmazione a blocchi funzionali ABL



Esempio di Diagramma ladder ABL



Esempio di Testo strutturato ABL

```

ABL_1 (
void ABL_1(BOOL IN, ABLACB ABLInput)
Tipo: ABL, Specificare il numero di caratteri nel buffer (incluso il carattere di fine riga).

1  ABL_1(in, input);
2  output := ABL_1.Q;
3  number := ABL_1.Characters;
4  error := ABL_1.Error;
5  ID := ABL_1.ErrorID;

```

Vedere anche

[Tipo di dati ABLACB](#) a pagina 131

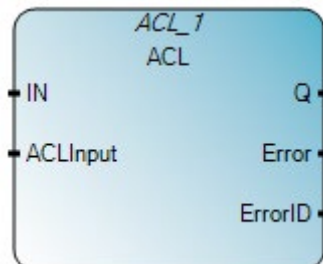
[Istruzioni porta seriale ASCII](#) a pagina 111

ACL (buffer cancellazione ASCII)

Cancella i buffer di ricezione e trasmissione e rimuove le istruzioni dalla coda ASCII.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

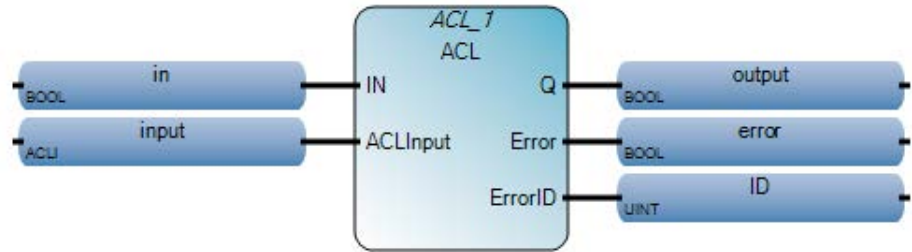
Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco funzione ammesso che l'ultima operazione sia stata completata. FALSE: il blocco istruzione non è attivo.
ACLInput	Ingresso	ACLI	Il canale su cui operare e lo stato dei buffer di trasmissione e ricezione. Se RXBuffer, elimina il buffer di ricezione e rimuove i blocchi funzione ASCII di ricezione (ARL e ARD) dalla coda ASCII. Se TXBuffer, elimina il buffer di trasmissione e rimuove i blocchi funzione ASCII di trasmissione (AWA e AWT) dalla coda ASCII. Utilizzare il tipo di dati ACLI per definire i parametri Canale, RXBuffer e TXBuffer per ACLInput.
Q	Uscita	BOOL	Indica se il processo di cancellazione della coda ASCII è in corso o completo. TRUE: il blocco funzione è completo. FALSE: il blocco funzione non è completo.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore ABL.

Codici di errore ABL

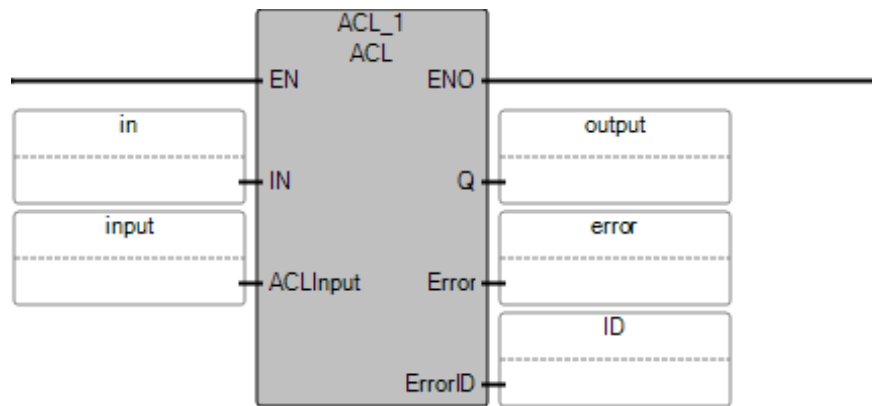
Utilizzare questa tabella per determinare i codici di errore ABL e le descrizioni.

Codice errore	Descrizione errore
03	Impossibile completare la trasmissione perché il segnale Clear-to-Send è stato perso.
06	È stato rilevato un parametro non valido.
07	Impossibile completare la ricezione o la trasmissione ASCII perché la configurazione del canale è stata disattivata nella finestra di dialogo di configurazione del canale.
08	Impossibile completare la scrittura ASCII perché è in corso un'altra trasmissione ASCII.
09	La richiesta di comunicazione ASCII non è supportata dalla configurazione del canale in uso.
10	L'esecuzione dell'istruzione è stata arrestata in seguito all'invio del comando Annulla. Nessuna azione richiesta.
11	La lunghezza richiesta per la stringa non è valida o è un numero negativo, maggiore di 82 o uguale a 0. Si applica ai blocchi funzione ARD e ARL.
13	Il valore richiesto (.LEN) nel blocco di funzione è un numero negativo o un valore maggiore della dimensione della stringa memorizzata con la stringa sorgente. Si applica ai blocchi funzione AWA e AWT.
14	Il blocco funzione ACL è stato interrotto.
16	Le linee di controllo RTS o CTS non sono supportate dalla porta seriale.

Esempio di Programmazione a blocchi funzionali ACL



Esempio di Diagramma ladder ACL



Esempio di Testo strutturato ACL

```

ACL_1 (
void ACL_1(BOOL IN, ACLI ACLInput)
Tipo: ACL, Eliminare i buffer di ricezione e/o trasmissione.

1  ACL_1(in, input);
2  output := ACL_1.Q;
3  error := ACL_1.Error;
4  ID := ACL_1.ErrorID;

```

Vedere anche

[Tipo di dati ACL](#) a pagina 131

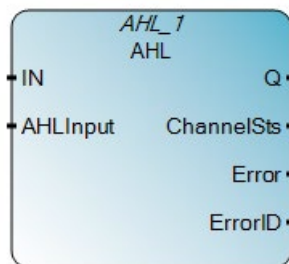
[Istruzioni porta seriale ASCII](#) a pagina 111

AHL (righe di arbitraggio ASCII)

Imposta o ripristina le linee di controllo di arbitraggio Request to Send (RTS) RS-232 del modem.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

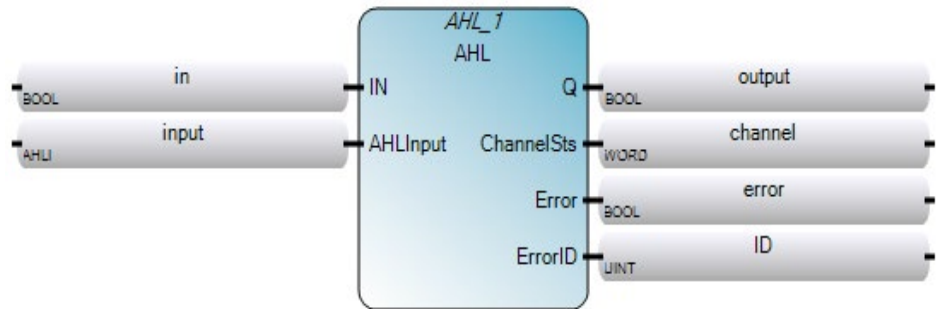
Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: il blocco istruzione non è attivo.
AHLInput	Ingresso	AHLI	Il canale su cui operare e l'impostazione o reset della linea di controllo RTS per il modem. Utilizzare il tipo di dati AHLI per definire i parametri Canale, SetRts, ClrRts, e Annulla per AHLInput.
Q	Uscita	BOOL	Indica se l'impostazione o il reset sono stati completati. Gli output di questa istruzione sono aggiornati in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: il blocco funzione è completo. FALSE: il blocco funzione non è completo.
ChannelSts	Uscita	WORD	Visualizza lo stato corrente (da 0000 a 001F) delle righe di arbitraggio per il canale specificato.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore ABL.

Codici di errore ABL

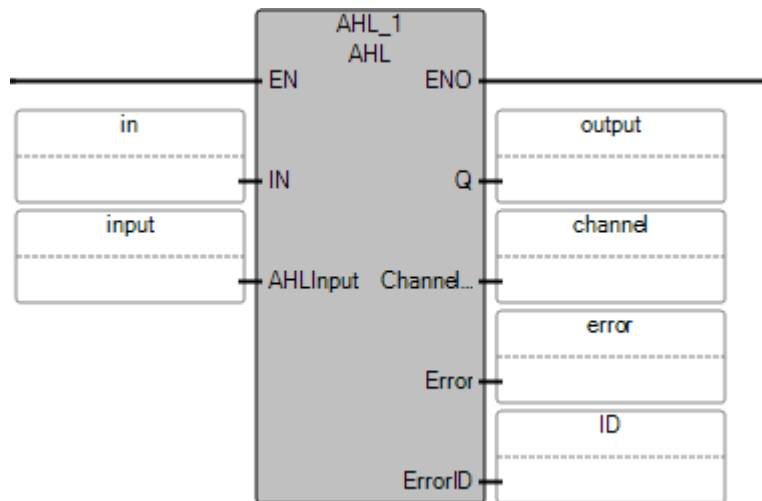
Utilizzare questa tabella per determinare i codici di errore ABL e le descrizioni.

Codice errore	Descrizione errore
03	Impossibile completare la trasmissione perché il segnale Clear-to-Send è stato perso.
06	È stato rilevato un parametro non valido.
07	Impossibile completare la ricezione o la trasmissione ASCII perché la configurazione del canale è stata disattivata nella finestra di dialogo di configurazione del canale.
08	Impossibile completare la scrittura ASCII perché è in corso un'altra trasmissione ASCII.
09	La richiesta di comunicazione ASCII non è supportata dalla configurazione del canale in uso.
10	L'esecuzione dell'istruzione è stata arrestata in seguito all'invio del comando Annulla. Nessuna azione richiesta.
11	La lunghezza richiesta per la stringa non è valida o è un numero negativo, maggiore di 82 o uguale a 0. Si applica ai blocchi funzione ARD e ARL.
13	Il valore richiesto (.LEN) nel blocco di funzione è un numero negativo o un valore maggiore della dimensione della stringa memorizzata con la stringa sorgente. Si applica ai blocchi funzione AWA e AWT.
14	Il blocco funzione ACL è stato interrotto.
16	Le linee di controllo RTS o CTS non sono supportate dalla porta seriale.

Esempio di programmazione a blocchi funzionali AHL



Esempio di Diagramma ladder AHL



Esempio di testo strutturato AHL

```
AHL_1 (
void AHL_1(BOOL IN, AHLI AHLInput)
Tipo: AHL, Impostare o reimpostare linee di arbitraggio modem.
```

```
1 AHL_1(in, input);
2 output := AHL_1.Q;
3 channel := AHL_1.ChannelSts;
4 error := AHL_1.Error;
5 ID := AHL_1.ErrorID;
```

Vedere anche

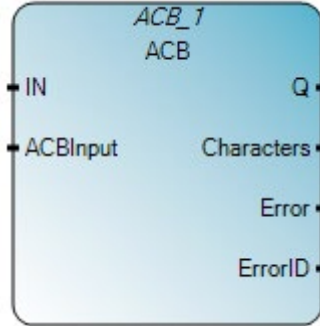
- [Tipo di dati AHLI](#) a pagina 132
- [Tipo di dati AHL ChannelSts](#) a pagina 131
- [Istruzioni porta seriale ASCII](#) a pagina 111

ACB (caratteri ASCII nel buffer)

Conta il numero totale di caratteri ASCII nel buffer, incluso il carattere di fine riga.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

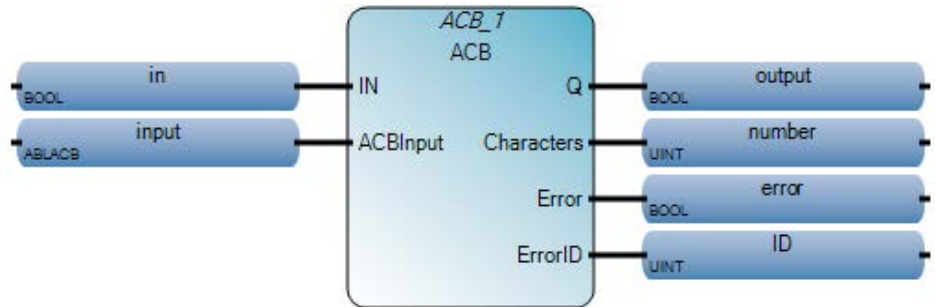
Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: il blocco istruzione non è attivo.
ACBInput	Ingresso	ABLACB	Canale su cui operare. Utilizzare il tipo di dati ABLACB per definire i parametri Canale, Tipo di trigger e Annulla per l'ingresso ACB.
Q	Uscita	BOOL	Indica se il conteggio dei caratteri è in corso o è stato completato. Le uscite di questo blocco funzione vengono aggiornate in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: il conteggio è completo. FALSE: il conteggio è in corso.
Characters	Uscita	UINT	Il numero di caratteri nel buffer.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. FALSE - Nessun errore. TRUE - È stato rilevato un errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore ABL.

Codici di errore ACB

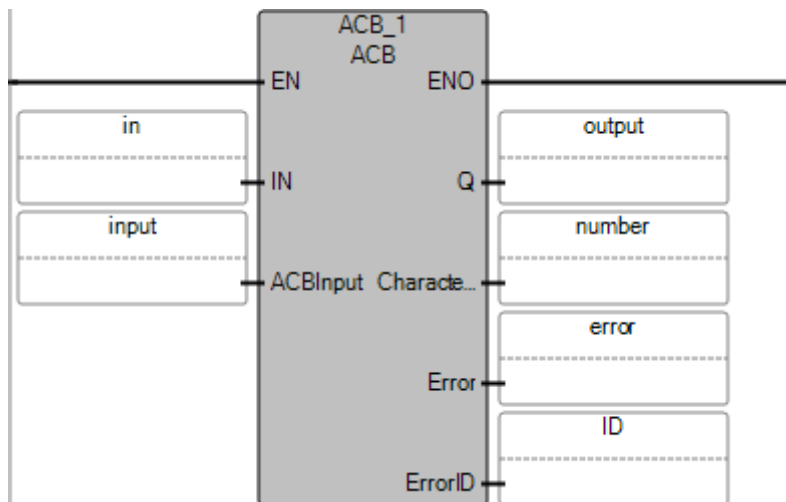
Utilizzare questa tabella per determinare i codici di errore ABL e le descrizioni.

Codice errore	Descrizione errore
03	Impossibile completare la trasmissione perché il segnale Clear-to-Send è stato perso.
06	È stato rilevato un parametro non valido.
07	Impossibile completare la ricezione o la trasmissione ASCII perché la configurazione del canale è stata disattivata nella finestra di dialogo di configurazione del canale.
08	Impossibile completare la scrittura ASCII perché è in corso un'altra trasmissione ASCII.
09	La richiesta di comunicazione ASCII non è supportata dalla configurazione del canale in uso.
10	L'esecuzione dell'istruzione è stata arrestata in seguito all'invio del comando Annulla. Nessuna azione richiesta.
11	La lunghezza richiesta per la stringa non è valida o è un numero negativo, maggiore di 82 o uguale a 0. Si applica ai blocchi funzione ARD e ARL.
13	Il valore richiesto (.LEN) nel blocco di funzione è un numero negativo o un valore maggiore della dimensione della stringa memorizzata con la stringa sorgente. Si applica ai blocchi funzione AWA e AWT.
14	Il blocco funzione ACL è stato interrotto.
16	Le linee di controllo RTS o CTS non sono supportate dalla porta seriale.

Esempio di Programmazione a blocchi funzionali ACB



Esempio di Diagramma ladder ACB



Esempio di Testo strutturato ACB

```
ACB_1(
void ACB_1(BOOL IN, ABLACB ACBInput)
Tipo: ACB, Determinare il numero totale di caratteri nel buffer.
```

```
1 ACB_1(in, input);
2 output := ACB_1.Q;
3 number := ACB_1.Characters;
4 error := ACB_1.Error;
5 ID := ACB_1.ErrorID;
```

Vedere anche

[Tipo di dati ABLACB](#) a pagina 131

[Istruzioni porta seriale ASCII](#) a pagina 111

ARD (lettura ASCII)

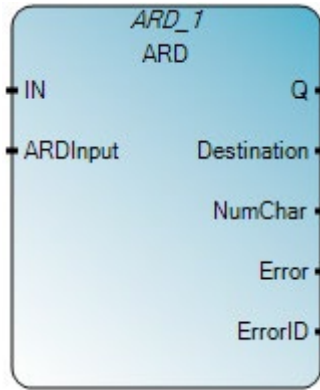
Legge i caratteri ASCII dal buffer di ingresso e li memorizza in una stringa.

Dettagli operazione:

- L'istruzione ARD viene eseguita finché non verranno ricevuti tutti i caratteri nel buffer di ASCII. Eventuali viene eseguita un'istruzione ASCII, viene messa in coda fino al completamento di ARD.
- Per interrompere l'istruzione ARD, eseguire un'istruzione ACL.
- Per evitare che l'istruzione ARD ritardi la coda ASCII mentre è in attesa del numero necessario di caratteri, usare i risultati di un'istruzione ACB per attivare l'istruzione ARD.
- Lo status dell'istruzione può essere estratto dal bit di controllo dell'istanza di istruzione (per esempio, ARD_1.controlbit). Questo mostra se l'istruzione sta bloccando la coda delle istruzioni ASCII in attesa di altri caratteri:
 - 7° bit = Istruzione abilitata.
 - 6° bit = Istruzione in coda.
 - 5° bit = Istruzione eseguita.
 - 3° bit = Istruzione con errore.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

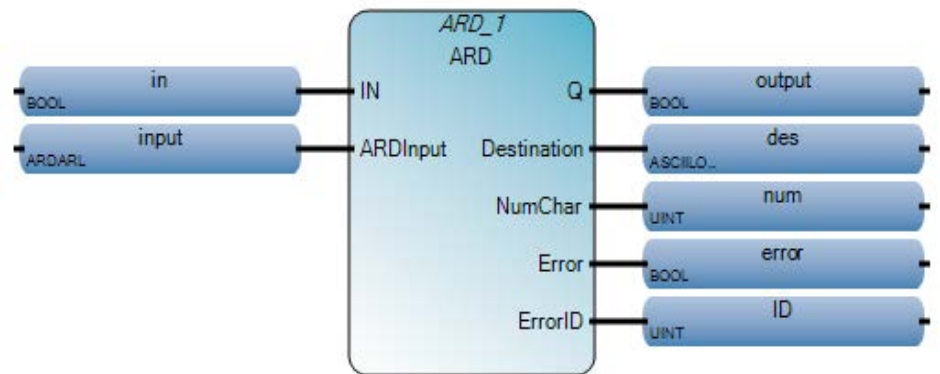
Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: il blocco istruzione non è attivo.
ARDInput	Ingresso	ARDARL	Legge i caratteri dal buffer. Il massimo è 82. Utilizzare il tipo di dati ARDARL per definire i parametri Canale, Lunghezza e Annulla per ARDInput.
Q	Uscita	BOOL	Indica quando la lettura del buffer è in corso o completata. Gli output di questa istruzione sono aggiornati in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: il blocco funzione è completo. FALSE: il blocco funzione non è completo.
Destinazione	Uscita	ASCIILOCADDR	Elemento della stringa in cui vengono memorizzati i caratteri.
NumChar	Uscita	UINT	Il numero di caratteri.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore ABL.

Codici di errore ABL

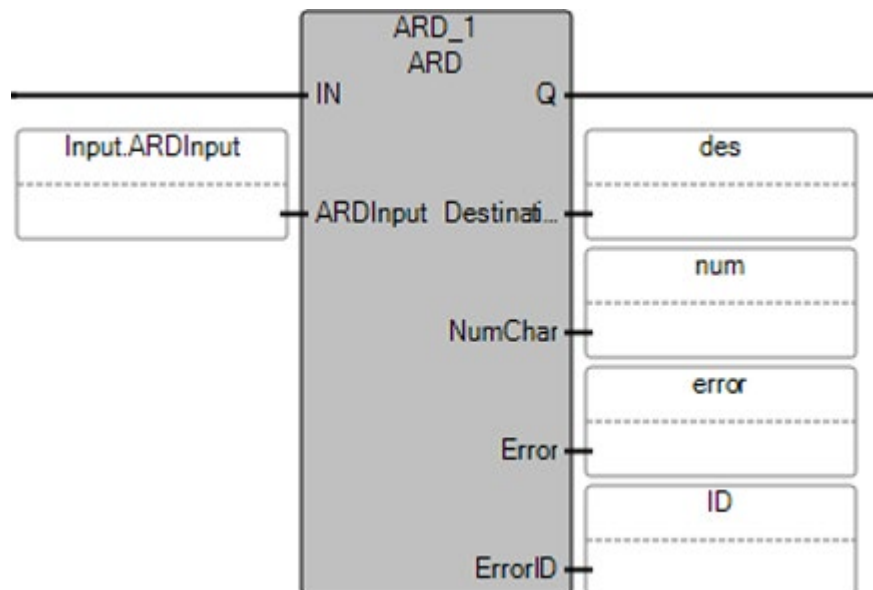
Utilizzare questa tabella per determinare i codici di errore ABL e le descrizioni.

Codice errore	Descrizione errore
03	Impossibile completare la trasmissione perché il segnale Clear-to-Send è stato perso.
06	È stato rilevato un parametro non valido.
07	Impossibile completare la ricezione o la trasmissione ASCII perché la configurazione del canale è stata disattivata nella finestra di dialogo di configurazione del canale.
08	Impossibile completare la scrittura ASCII perché è in corso un'altra trasmissione ASCII.
09	La richiesta di comunicazione ASCII non è supportata dalla configurazione del canale in uso.
10	L'esecuzione dell'istruzione è stata arrestata in seguito all'invio del comando Annulla. Nessuna azione richiesta.
11	La lunghezza richiesta per la stringa non è valida o è un numero negativo, maggiore di 82 o uguale a 0. Si applica ai blocchi funzione ARD e ARL.
13	Il valore richiesto (.LEN) nel blocco di funzione è un numero negativo o un valore maggiore della dimensione della stringa memorizzata con la stringa sorgente. Si applica ai blocchi funzione AWA e AWT.
14	Il blocco funzione ACL è stato interrotto.
16	Le linee di controllo RTS o CTS non sono supportate dalla porta seriale.

Esempio di programmazione a blocchi funzionali ARD



Esempio di diagramma ladder ARD



Esempio di testo strutturato ARD

```
ARD_1(
void ARD_1(BOOL IN, ARDARL ARDInput)
Tipo: ARD, Leggere i caratteri dal buffer di ingresso e inserirli in una stringa.
```

```
1 ARD_1(in, input);
2 output := ARD_1.Q;
3 des := ARD_1.Destination;
4 num := ARD_1.NumChar;
5 error := ARD_1.Error;
6 ID := ARD_1.ErrorID;
```

Vedere anche

[Tipo di dati ARDARL](#) a pagina 132

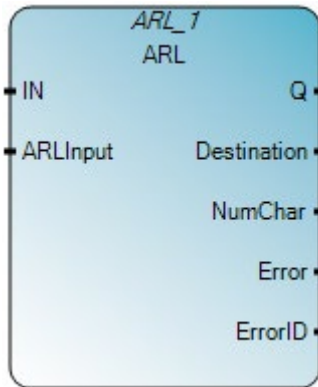
[Istruzioni porta seriale ASCII](#) a pagina 111

ARL (riga di lettura ASCII)

Legge una riga di caratteri ASCII dal buffer fino ai caratteri di terminazione inclusi e li memorizza in una stringa.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: il blocco istruzione non è attivo.
ARDInput	Ingresso	ARDARL	Leggere una riga di caratteri ASCII dal buffer. Il massimo è 82. Utilizzare il tipo di dati ARDARL per definire i parametri Canale, Lunghezza e Annulla per ARDLInput.

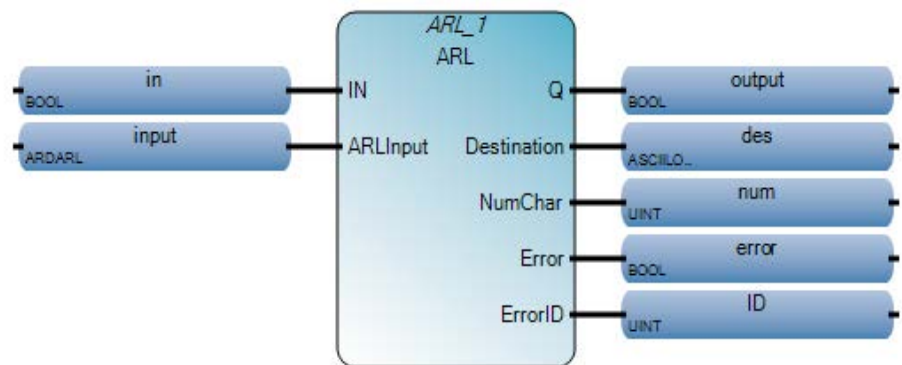
Q	Uscita	BOOL	Indica quando la linea di lettura dal buffer di ingresso è in corso o è stata completata. Gli output di questa istruzione sono aggiornati in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: il blocco funzione è completo. FALSE: il blocco funzione non è completo.
Destinazione	Uscita	ASCIILOCADDR	Elemento della stringa in cui vengono memorizzati i caratteri.
NumChar	Uscita	UINT	Il numero di caratteri nella riga, incluso il carattere di terminazione.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore ABL.

Codici di errore ABL

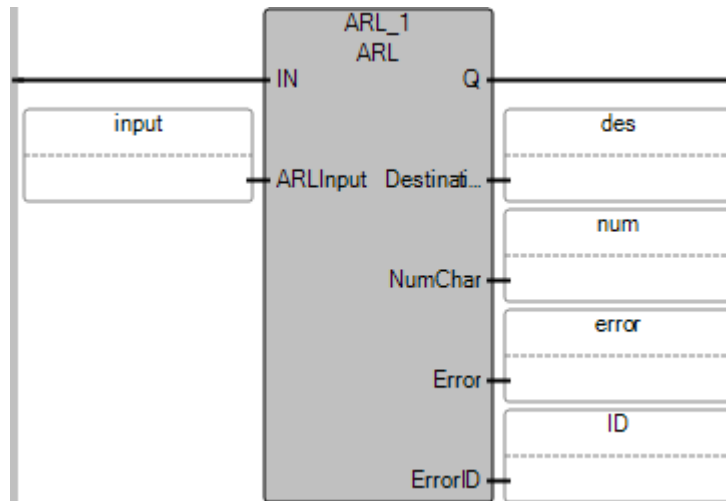
Utilizzare questa tabella per determinare i codici di errore ABL e le descrizioni.

Codice errore	Descrizione errore
03	Impossibile completare la trasmissione perché il segnale Clear-to-Send è stato perso.
06	È stato rilevato un parametro non valido.
07	Impossibile completare la ricezione o la trasmissione ASCII perché la configurazione del canale è stata disattivata nella finestra di dialogo di configurazione del canale.
08	Impossibile completare la scrittura ASCII perché è in corso un'altra trasmissione ASCII.
09	La richiesta di comunicazione ASCII non è supportata dalla configurazione del canale in uso.
10	L'esecuzione dell'istruzione è stata arrestata in seguito all'invio del comando Annulla. Nessuna azione richiesta.
11	La lunghezza richiesta per la stringa non è valida o è un numero negativo, maggiore di 82 o uguale a 0. Si applica ai blocchi funzione ARD e ARL.
13	Il valore richiesto (.LEN) nel blocco di funzione è un numero negativo o un valore maggiore della dimensione della stringa memorizzata con la stringa sorgente. Si applica ai blocchi funzione AWA e AWT.
14	Il blocco funzione ACL è stato interrotto.
16	Le linee di controllo RTS o CTS non sono supportate dalla porta seriale.

Esempio di diagramma di blocco funzione ARL



Esempio di diagramma ladder ARL



Esempio di testo strutturato ARL

```
ARL_1(
void ARL_1(BOOL IN, ARDARL ARLInput)
Tipo: ARL, Leggere la riga dal buffer di ingresso e inserire i caratteri in una stringa.
```

```
1 ARL_1(in, input);
2 output := ARL_1.Q;
3 des := ARL_1.Destination;
4 num := ARL_1.NumChar;
5 error := ARL_1.Error;
6 ID := ARL_1.ErrorID;
```

Vedere anche

[Tipo di dati ARDARL](#) a pagina 132

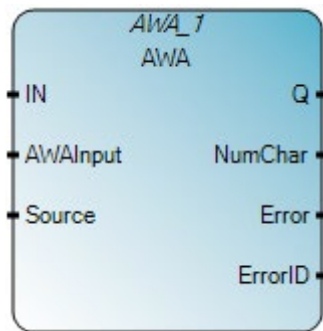
[Istruzioni porta seriale ASCII](#) a pagina 111

AWA (scrittura ASCII aggiunta)

Scrivere una stringa con due caratteri aggiunti (configurati dall'utente) su un dispositivo esterno.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

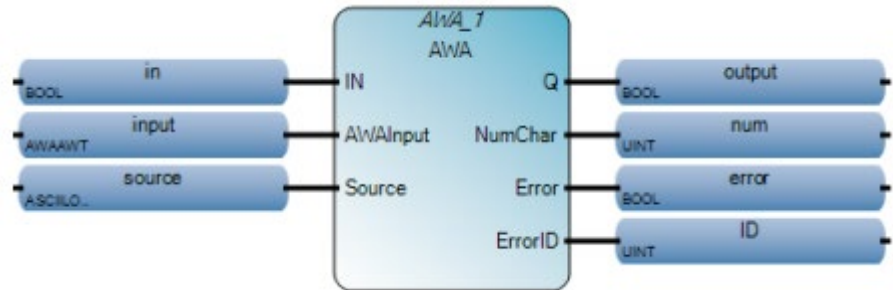
Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione, ammesso che l'ultima operazione sia stata completata. FALSE: il blocco istruzione non è attivo.
AWAInput	Ingresso	AWAAWT	Il canale e il numero (lunghezza) dei caratteri da scrivere nel buffer. Il valore massimo è 82. Utilizzare il tipo di dati AWAAWT per definire i parametri Canale, Lunghezza e Annulla per AWAInput.
Origine	Ingresso	ASCIILOCADDR	La stringa sorgente uscita come vettore di caratteri secondo l'istruzione ARD o ARL.
Q	Uscita	BOOL	Indica quando la scrittura è in corso o completata. Gli output di questa istruzione sono aggiornati in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: il blocco funzione è completo. FALSE: il blocco funzione non è completo.
NumChar	Uscita	UINT	Il numero di caratteri. NumChar può essere inferiore alla lunghezza richiesta da trasmettere se la lunghezza della stringa Source String è inferiore alla lunghezza richiesta. Aggiorna quando la trasmissione è completa e Q è TRUE.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore ABL.

Codici di errore ABL

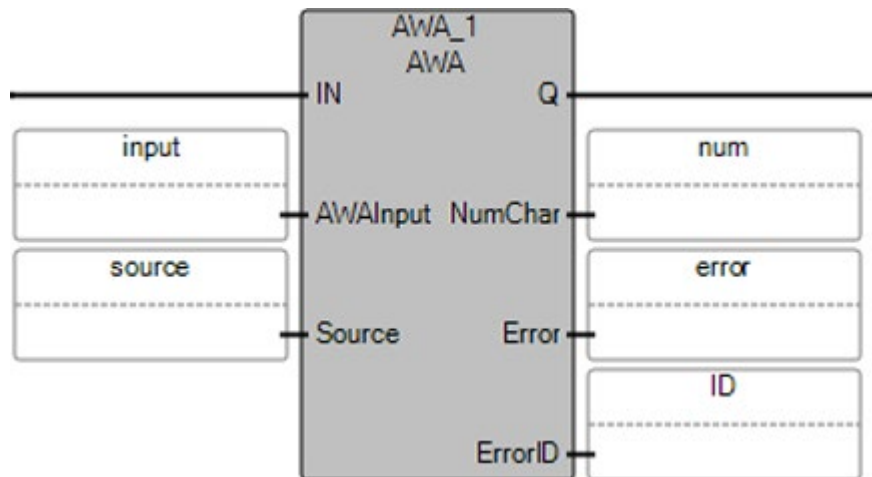
Utilizzare questa tabella per determinare i codici di errore ABL e le descrizioni.

Codice errore	Descrizione errore
03	Impossibile completare la trasmissione perché il segnale Clear-to-Send è stato perso.
06	È stato rilevato un parametro non valido.
07	Impossibile completare la ricezione o la trasmissione ASCII perché la configurazione del canale è stata disattivata nella finestra di dialogo di configurazione del canale.
08	Impossibile completare la scrittura ASCII perché è in corso un'altra trasmissione ASCII.
09	La richiesta di comunicazione ASCII non è supportata dalla configurazione del canale in uso.
10	L'esecuzione dell'istruzione è stata arrestata in seguito all'invio del comando Annulla. Nessuna azione richiesta.
11	La lunghezza richiesta per la stringa non è valida o è un numero negativo, maggiore di 82 o uguale a 0. Si applica ai blocchi funzione ARD e ARL.
13	Il valore richiesto (.LEN) nel blocco di funzione è un numero negativo o un valore maggiore della dimensione della stringa memorizzata con la stringa sorgente. Si applica ai blocchi funzione AWA e AWT.
14	Il blocco funzione ACL è stato interrotto.
16	Le linee di controllo RTS o CTS non sono supportate dalla porta seriale.

Esempio di programmazione a blocchi funzionali AWA



Esempio di diagramma ladder AWA



Esempio di testo strutturato AWA

```
AWA_1 (
void AWA_1(BOOL IN, AWAAWT AWAInput, ASCIILOCADDR Source)
Tipo: AWA, Scrivere una stringa con caratteri aggiunti a un dispositivo esterno.
```

```
1 AWA_1(in, input, source);
2 output := AWA_1.Q;
3 num := AWA_1.NumChar;
4 error := AWA_1.Error;
5 ID := AWA_1.ErrorID;
```

Vedere anche

[Istruzioni porta seriale ASCII](#) a pagina 111

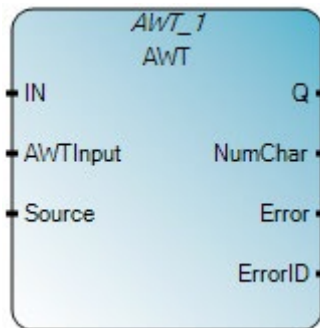
[Tipo di dati AWAAWT](#) a pagina 133

AWT (scrittura ASCII)

Scrive caratteri ASCII da una stringa sorgente a un dispositivo esterno

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: il blocco istruzione non è attivo.
AWTInput	Ingresso	AWAAWT	Il canale e il numero (lunghezza) dei caratteri da scrivere nel buffer. Il valore massimo è 82. Utilizzare il tipo di dati AWAAWT per definire i parametri Canale, Lunghezza e Annulla per AWTInput.
Origine	Ingresso	ASCIILOCADDR	La stringa sorgente uscita come vettore di caratteri secondo l'istruzione ARD o ARL.

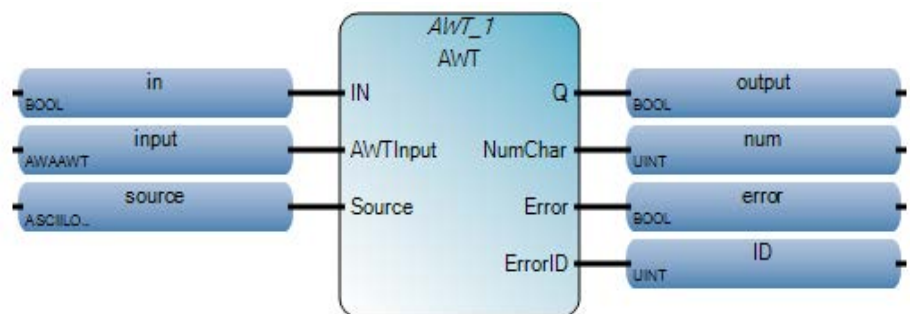
Q	Uscita	BOOL	Indica quando la scrittura è in corso o completata. Gli output di questa istruzione sono aggiornati in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: il blocco funzione è completo. FALSE: il blocco funzione non è completo.
NumChar	Uscita	UINT	Il numero di caratteri. NumChar può essere inferiore alla lunghezza richiesta da trasmettere se la lunghezza della stringa Source String è inferiore alla lunghezza richiesta. Aggiorna quando la trasmissione è completa e Q è TRUE.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore ABL.

Codici di errore ABL

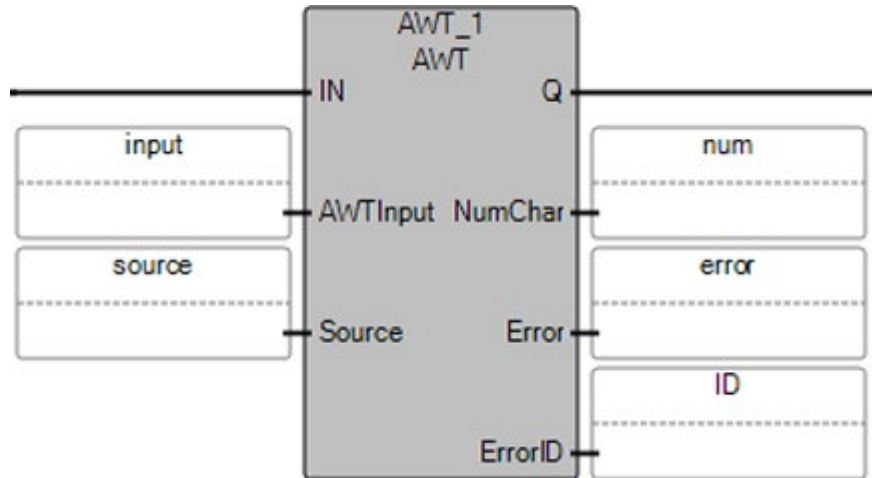
Utilizzare questa tabella per determinare i codici di errore ABL e le descrizioni.

Codice errore	Descrizione errore
03	Impossibile completare la trasmissione perché il segnale Clear-to-Send è stato perso.
06	È stato rilevato un parametro non valido.
07	Impossibile completare la ricezione o la trasmissione ASCII perché la configurazione del canale è stata disattivata nella finestra di dialogo di configurazione del canale.
08	Impossibile completare la scrittura ASCII perché è in corso un'altra trasmissione ASCII.
09	La richiesta di comunicazione ASCII non è supportata dalla configurazione del canale in uso.
10	L'esecuzione dell'istruzione è stata arrestata in seguito all'invio del comando Annulla. Nessuna azione richiesta.
11	La lunghezza richiesta per la stringa non è valida o è un numero negativo, maggiore di 82 o uguale a 0. Si applica ai blocchi funzione ARD e ARL.
13	Il valore richiesto (.LEN) nel blocco di funzione è un numero negativo o un valore maggiore della dimensione della stringa memorizzata con la stringa sorgente. Si applica ai blocchi funzione AWA e AWT.
14	Il blocco funzione ACL è stato interrotto.
16	Le linee di controllo RTS o CTS non sono supportate dalla porta seriale.

Esempio di programmazione a blocchi funzionali AWT



Esempio di diagramma ladder AWT



Esempio di testo strutturato AWT

```

AWT_1 (
void AWT_1(BOOL IN, AWAAWT AWTInput, ASCIILOCADDR Source)
Tipo: AWT, Scrivere caratteri da una stringa di origine a un dispositivo esterno.

1  AWT_1(in, input, source);
2  output := AWT_1.Q;
3  num := AWT_1.NumChar;
4  error := AWT_1.Error;
5  ID := AWT_1.ErrorID;

```

Vedere anche

[Istruzioni porta seriale ASCII](#) a pagina 111

[Tipo di dati AWAAWT](#) a pagina 133

Dettagli dei parametri ASCII

I seguenti argomenti forniscono maggiori dettagli sui parametri ASCII e i tipi di dati strutturati.

- Tipo di dati ABLACB
- Tipo di dati ACL
- Tipo di dati AHL ChannelSts
- Tipo di dati AHLI
- Tipo di dati ARDARL
- Tipo di dati AWAAWT

Vedere anche

[Istruzioni porta seriale ASCII](#) a pagina 111

Tipo di dati ABLACB

Utilizzare questa tabella per determinare i valori dei parametri per il tipo di dati ABLACB.

Parametri	Tipo di dati	Descrizione
Canale	UINT	Numero seriale porta: <ul style="list-style-type: none"> • 2 per la porta seriale integrata o • 5-9 per i plug-in della porta seriale installati negli slot 1 - 5: • 5 per lo slot 1 • 6 per lo slot 2 • 7 per lo slot 3 • 8 per lo slot 4 • 9 per lo slot 5
TriggerType	USINT	Rappresenta uno dei seguenti: <ul style="list-style-type: none"> • 0: Msg attivato una volta (quando IN passa da False a True) • 1: Msg attivato di continuo quando IN è True • Altro valore: Riservato
Annulla	BOOL	Quando questo input è impostato a TRUE, questo blocco funzione non viene eseguito.

Vedere anche

[Dettagli dei parametri ASCII](#) a pagina 130

Tipo di dati ACL

Utilizzare questa tabella per determinare i valori dei parametri per il tipo di dati ABL.

Parametri	Tipo di dati	Descrizione
Canale	UINT	Numero seriale porta: <ul style="list-style-type: none"> • 2 per la porta seriale integrata o • 5-9 per i plug-in della porta seriale installati negli slot 1 - 5: • 5 per lo slot 1 • 6 per lo slot 2 • 7 per lo slot 3 • 8 per lo slot 4 • 9 per lo slot 5
RXBuffer	BOOL	Se TRUE, elimina il buffer di ricezione e rimuove i blocchi funzione ASCII di ricezione (ARL e ARD) dalla coda ASCII.
TXBuffer	BOOL	Se TRUE, elimina il buffer di trasmissione e rimuove i blocchi funzione ASCII di trasmissione (ARL e ARD) dalla coda ASCII.

Vedere anche

[Dettagli dei parametri ASCII](#) a pagina 130

Tipo di dati AHL ChannelSts

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati AHL ChannelSts.

Parametri	Tipo di dati	Descrizione
DTRstatus	UINT	Utilizzato per il segnale DTR (riservato)
DCDstatus	UINT	Utilizzato per il segnale DCD (bit 3 di parola) 1 indica attivo
DSRstatus	UINT	Utilizzato per il segnale DSR (riservato)
RTSstatus	UINT	Utilizzato per il segnale RTS (bit 1 di parola) 1 indica attivo
CTSstatus	UINT	Utilizzato per il segnale CTS (bit 0 di parola) 1 indica attivo

Vedere anche

[Dettagli dei parametri ASCII](#) a pagina 130

[Tipo di dati AHLI](#) a pagina 132

Tipo di dati AHLI

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati AHL.

Parametri	Tipo di dati	Descrizione
Canale	UINT	Numero seriale porta: <ul style="list-style-type: none"> • 2 per la porta seriale integrata o • 5-9 per i plug-in della porta seriale installati negli slot 1 - 5: • 5 per lo slot 1 • 6 per lo slot 2 • 7 per lo slot 3 • 8 per lo slot 4 • 9 per lo slot 5
ClrRts	BOOL	Utilizzato per ripristinare la linea di controllo RTS.
SetRts	BOOL	Utilizzato per impostare la linea di controllo RTS.
Annulla	BOOL	Quando questo input è impostato a TRUE, questo blocco funzione non viene eseguito.

Vedere anche

[Dettagli dei parametri ASCII](#) a pagina 130

[Tipo di dati AHL ChannelSts](#) a pagina 131

Tipo di dati ARDARL

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati ARDARL.

Parametri	Tipo di dati	Descrizione
Canale	UINT	Numero seriale porta: <ul style="list-style-type: none"> • 2 per la porta seriale integrata o • 5-9 per i plug-in della porta seriale installati negli slot 1 - 5: • 5 per lo slot 1 • 6 per lo slot 2 • 7 per lo slot 3 • 8 per lo slot 4 • 9 per lo slot 5
Lunghezza	UINT	Il numero di caratteri che si vuole leggere dal buffer (massimo 82).
Annulla	BOOL	Quando questo input è impostato a TRUE, questo blocco funzione non viene eseguito. Se già in esecuzione, l'operazione termina.

Vedere anche

[Dettagli dei parametri ASCII](#) a pagina 130

Tipo di dati AWAAT

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati AWAAT.

Parametri	Tipo di dati	Descrizione
Canale	UINT	Numero seriale porta: <ul style="list-style-type: none"> • 2 per la porta seriale integrata o • 5-9 per i plug-in della porta seriale installati negli slot 1 - 5: • 5 per lo slot 1 • 6 per lo slot 2 • 7 per lo slot 3 • 8 per lo slot 4 • 9 per lo slot 5
Lunghezza	UINT	Definisce il numero di caratteri che da scrivere nel buffer (massimo 82). Se si imposta la lunghezza a 0, AWA invia 0 byte di dati utente e 2 byte di caratteri aggiunti al buffer.
Annulla	BOOL	Se TRUE, il blocco funzione non viene eseguito. Se già in esecuzione, l'operazione termina.

Vedere anche

[Dettagli dei parametri ASCII](#) a pagina 130

Istruzioni binarie

Usare le istruzioni binarie per eseguire operazioni matematiche.

Operatore	Descrizione
AND_MASK a pagina 135	Esegue AND bit-to-bit tra due valori Integer.
NOT_MASK a pagina 144	Maschera negazione numero intero bit-to-bit, inverte il valore di un parametro.
BSL a pagina 137	Sposta un bit in un elemento vettore verso sinistra.
BSR a pagina 141	Sposta un bit in un elemento vettore verso destra.
OR_MASK a pagina 146	Maschera OR numero intero bit-to-bit, abilita i bit.
ROL a pagina 147	Per i numeri Integer a 32 bit, ruota i bit interi a sinistra.
ROR a pagina 149	Per i numeri Integer a 32 bit, ruota i bit interi a sinistra.
SHL a pagina 151	Per gli integer a 32-bit, sposta gli integer a sinistra e inserisce 0 nel bit meno significativo.
SHR a pagina 153	Per gli Integer a 32 bit, sposta gli Integer a sinistra e colloca 0 nel bit più significativo.
XOR_MASK a pagina 155	Maschera OR intero esclusivo bit-to-bit, restituisce valori di bit invertiti.

Vedere anche

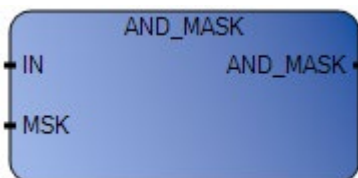
[Istruzione impostate in ordine alfabetico](#) a pagina 18

AND_MASK (maschera AND)

Esegue AND bit-to-bit tra due valori Integer.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

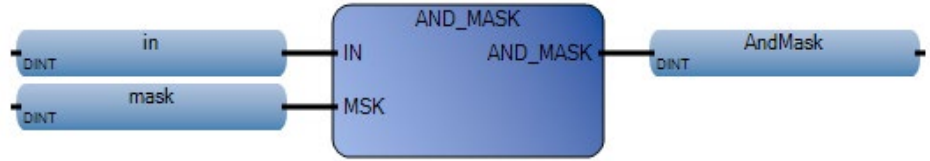


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

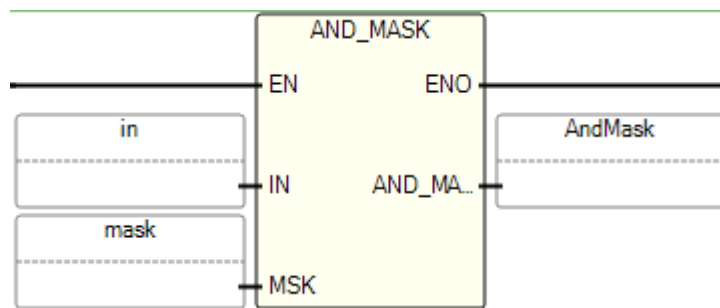
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della maschera per Integer AND bit-to-bit FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	DINT	Deve avere il formato Integer.
MSK	Ingresso	DINT	Deve avere il formato Integer.

AND_MASK	Uscita	DINT	Bit-to-bit logico AND tra IN e MSK.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali AND_MASK



Esempio di diagramma ladder AND_MASK



Esempio di testo strutturato AND_MASK

```
AND_MASK (
    DINT AND_MASK(DINT IN, DINT MSK)
    Mascheramento analogico AND bit a bit
```

```
1 in := 5;
2 mask := 6;
3 AndMask := AND_MASK(in, mask);
```

(* Equivalenza ST: *)

parity := AND_MASK (xvalue, 1); (* 1 se xvalue è dispari *)

result := AND_MASK (16#abc, 16#0f); (* uguale a 16#a0c *)

Risultati

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca
mask		6	N/D		<input type="checkbox"/>
in		5	N/D		<input type="checkbox"/>
AndMask		ATTESA	N/D		<input type="checkbox"/>

Vedere anche

[Istruzioni binarie](#) a pagina 135

BSL (spostamento del bit a sinistra)

Sposta un bit in un elemento vettore verso sinistra.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

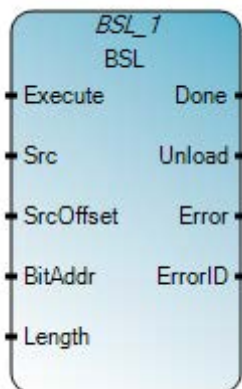
Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

Dettagli operazione:

L'istruzione BSL è un processo immediato nella transizione del ramo da false a true e negli aggiornamenti in uscita in modo sincrono. Quando Execute è TRUE, il bit più a sinistra (Src + SrcOffset e Length) viene copiato nel bit Unload e tutti i bit nel vettore o meno vengono spostati a sinistra di un bit. Length e limite di 16 bit vengono considerati tranne che per i tipi di dati BOOL. Il bit esterno viene quindi spostato sul bit 0 (Src + SrcOffset) del primo elemento.

Per le operazioni di testo intorno, impostare la posizione di BitAddr sull'ultima posizione del bit o sul bit Unload. Possibile utilizzo

dell'istruzione BSL, tracciare bottiglie in una linea di imbottigliamento dove ogni bit rappresenta una bottiglia.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

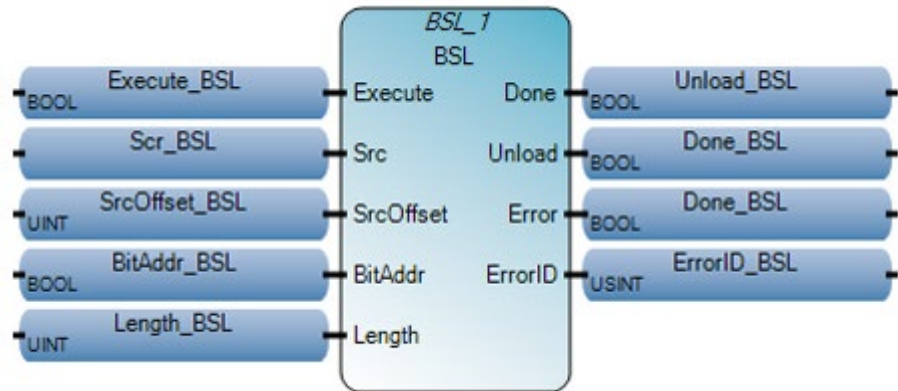
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	<p>Abilitazione istruzioni.</p> <p>TRUE: fronte di salita rilevato, sposta il bit di una posizione a sinistra.</p> <ul style="list-style-type: none"> Innanzitutto vengono verificate le condizioni di errore. Se Length = 0, il bit esterno viene spostato nel bit Unload. Nessun spostamento di bit su Scr. Viene eseguito il reset dei bit Error ed ErrorID. Viene impostato il bit Done. Se Length > 0 e Length ≤ 2048, viene eseguito il reset dei bit Error ed ErrorID. Una volta completato lo spostamento dei bit, il bit Done viene impostato. Se Length > 0 e Length ≤ 2048, il bit più a sinistra (indirizzato da Src + SrcOffset e Length) viene copiato nel bit Scarica e tutti i bit che fanno parte del vettore o no vengono spostati a sinistra di un bit (fino al bit Length e al limite di 16 bit, ad eccezione di BOOL). Il bit esterno viene spostato sul bit 0 (Src + SrcOffset) del primo elemento. <p>FALSE: fronte di salita non rilevato, non attivare l'operazione BSL.</p>
Scr	Ingresso	ANY_ELEMENTARY	<p>L'indirizzo del (bit) Src da spostare. Tipi di dati supportati: BOOL, DWORD, INT, UINT, WORD, DINT e UDINT.</p> <ul style="list-style-type: none"> Vettori: Impostare Scr su un indirizzo a base variabile come: Source1, Source1[0] o Source1[1]. Non vettori: Impostare Scr su un indirizzo variabile come Source1.
SrcOffset	Ingresso	UINT	<p>Se SrcOffset è 0, iniziare dal primo elemento.</p> <ul style="list-style-type: none"> Vettori: Impostare SrcOffset su 0. Se impostato su Source1[0] o Source1[1], si verifica un errore: "L'offset di origine supera le dimensioni dell'array". Non vettori: Impostare SrcOffset su 0; in caso contrario si verifica un errore: "L'offset di origine supera le dimensioni dell'array".
BitAddr	Ingresso	BOOL	<p>Posizione del bit spostato in Scr.</p>
Lunghezza	Ingresso	UINT	<p>Lunghezza contiene il numero di bit presenti in Src da spostare. Supporta lo spostamento tra elementi dell'array.</p> <ul style="list-style-type: none"> Per il tipo di dati BOOL, il numero di Booleani presenti nel vettore da spostare. Per i tipi di dati a 16 e 32 bit, i bit vengono spostati in multipli di 16, ad esempio 16, 32 e 64. Se Length non è un multiplo esatto di 16, il numero di bit spostati viene inviato al successivo limite di 16 bit. Length si basa sulla dimensione del tipo di dati. Se si supera l'intervallo viene generato un errore con il seguente messaggio: "L'offset di origine supera le dimensioni del vettore". <p>Valori Lunghezza:</p> <ul style="list-style-type: none"> BOOL: 1 Parola da 16 bit: 1-16 Parola da 32 bit: 1-32 Parola da 64 bit: 1-64
Done	Uscita	BOOL	<p>Se TRUE, l'operazione è stata completata correttamente.</p> <p>Se FALSE, durante l'operazione è stata rilevata una condizione di errore.</p>
Scarica	Uscita	BOOL	<p>Bit spostato fuori dall'indirizzo Src.</p>

Errore	Uscita	BOOL	Se si verifica un errore, Error viene impostato su true.
ErrorID	Uscita	USINT	Se si verifica un errore, ErrorID contiene il codice dell'errore.

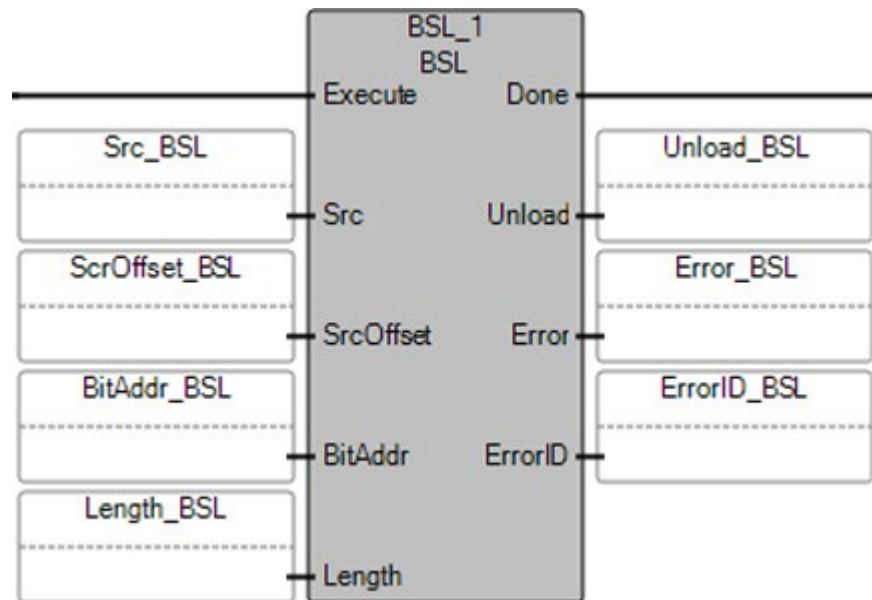
Codici di errore BSL

Codice errore	Descrizione errore
01	Dimensioni non supportate.
02	Tipo di dati non supportato.
03	La lunghezza dei bit supera 2048.
04	L'offset di origine supera la dimensione dell'array.
05	La lunghezza dei bit supera la dimensione dell'array.
07	Parametro non valido.

Esempio di programmazione a blocchi funzionali BSL



Esempio di Diagramma ladder BSL



Esempio di Testo strutturato BSL

```
BSL_1(
void BSL_1(BOOL Execute, ANY_ELEMENTARY[1..1] Src, UINT SrcOffset, BOOL BitAddr, UINT Length)
Tipo: BSL, Esegue l'operazione "Scorri bit a sinistra".
```

```
1 BSL_1(Execute, Src, SrcOffset, BitAddr, Length);
2 Done_BSL := BSL_1.Done;
3 Unload_BSL := BSL_1.Unload;
4 Error_BSL := BSL_1.Error;
5 ErrorID_BSL := BSL_1.ErrorID;
```


Risultati

	Nome	Alias	Valore logico	Valore fisico	Valore inizia	Blocca	Tipo di dati	Dimensione
	BSL_1		<input type="checkbox"/>	BSL	
	Execute		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
	Src		<input type="checkbox"/>	BOOL	[1..4]
	SrcOffset		0	N/D		<input type="checkbox"/>	UINT	
	BitAddr		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
	Length		0	N/D		<input type="checkbox"/>	UINT	
	Done_BSL		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
	Upload_BSL		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
	Error_BSL		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
	ErrorID_BSL		0	N/D		<input type="checkbox"/>	USINT	

Vedere anche

[Istruzioni binarie](#) a pagina 135

BSR (spostamento del bit a destra)

Sposta un bit in un elemento vettore verso destra.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

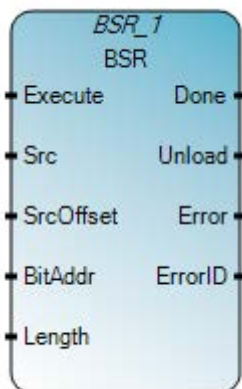
Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

Dettagli operazione:

L'istruzione BSR è un processo immediato nella transizione del ramo da false a true e negli aggiornamenti in uscita in modo sincrono. Se Execute è TRUE, il bit più a destra (bit 0 dell'elemento indirizzato da Src + SrcOffset) viene copiato nel bit Unload e tutti i bit nel vettore o all'esterno del vettore vengono spostati a destra di un bit. Length e limite di 16 bit vengono considerati tranne che per i tipi di dati BOOL. Il bit esterno viene quindi spostato sul bit 0 (Src + SrcOffset) del primo elemento.

Per le operazioni di testo intorno, impostare la posizione di BitAddr sull'ultima posizione del bit o sul bit Unload. Possibile utilizzo

dell'istruzione BSL, tracciare bottiglie in una linea di imbottigliamento dove ogni bit rappresenta una bottiglia.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione istruzioni. TRUE: fronte di salita rilevato, sposta il bit di una posizione a destra. FALSE: fronte di salita non rilevato, non attivare l'operazione BSR.
Src	Ingresso	ANY_ELEMENTARY	L'indirizzo del (bit) Src da spostare. Tipi di dati supportati: BOOL, DWORD, INT, UINT, WORD, DINT e UDINT. <ul style="list-style-type: none"> • Vettori: Impostare Src su un indirizzo a base variabile come: Source1, Source1[0] o Source1[1]. • Non vettori: Impostare Src su un indirizzo variabile come Source1.
SrcOffset	Ingresso	UINT	Se SrcOffset è 0, iniziare dal primo elemento. <ul style="list-style-type: none"> • Vettori: Impostare SrcOffset su 0. Se impostato su Source1[0] o Source1[1], si verifica un errore: "L'offset di origine supera le dimensioni dell'array". • Non vettori: Impostare SrcOffset su 0; in caso contrario si verifica un errore: "L'offset di origine supera le dimensioni dell'array".
BitAddr	Ingresso	BOOL	Posizione del bit spostato in Src.
Lunghezza	Ingresso	UINT	Lunghezza contiene il numero di bit presenti in Src da spostare. Supporta lo spostamento tra elementi dell'array. <ul style="list-style-type: none"> • Per il tipo di dati BOOL, il numero di Booleani presenti nel vettore da spostare. • Per i tipi di dati a 16 e 32 bit, i bit vengono spostati in multipli di 16, ad esempio 16, 32 e 64. Se Lunghezza non è un multiplo esatto di 16, il numero di bit spostati è il limite successivo di 16 bit. • Length si basa sulla dimensione del tipo di dati. Se si supera l'intervallo viene generato un errore con il seguente messaggio: "L'offset di origine supera le dimensioni del vettore". Valori Lunghezza: <ul style="list-style-type: none"> • BOOL: 1 • Parola da 16 bit: 1-16 • Parola da 32 bit: 1-32 • Parola da 64 bit: 1-64
Finito	Uscita	BOOL	Se TRUE, l'operazione è stata completata correttamente. Se FALSE, durante l'operazione è stata rilevata una condizione di errore.
Scarica	Uscita	BOOL	Bit spostato fuori dall'indirizzo Src.
Errore	Uscita	BOOL	Se si verifica un errore, Error viene impostato su true.
ErrorID	Uscita	USINT	Se si verifica un errore, ErrorID contiene il codice dell'errore.

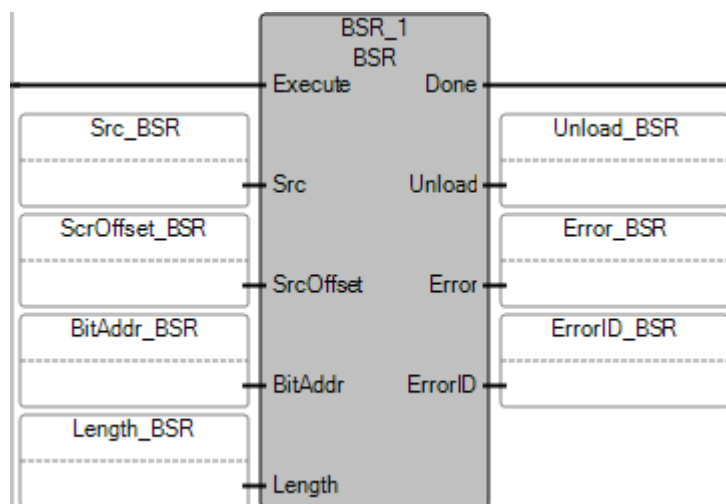
Codici di errore BSR

Codice errore	Descrizione errore
01	Dimensioni non supportate.
02	Tipo di dati non supportato.
03	La lunghezza dei bit supera 2048.
04	L'offset di origine supera la dimensione dell'array.
05	La lunghezza dei bit supera la dimensione dell'array.
07	Parametro non valido.

Esempio di programmazione a blocchi funzionali BSR



Esempio di diagramma ladder BSR



Esempio di testo strutturato BSR

```

BSR_1 (
void BSR_1(BOOL Execute, ANY_ELEMENTARY[1..1] Src, UINT SrcOffset, BOOL BitAddr, UINT Length)
Tipo: BSR, Esegue l'operazione "Scorri bit a destra".

1 BSR_1(Execute, Src, SrcOffset, BitAddr, Length);
2 Done_BSR := BSR_1.Done;
3 Unload_BSR := BSR_1.Unload;
4 Error_BSR := BSR_1.Error;
5 ErrorID_BSR := BSR_1.ErrorID;
    
```

Risultati

	Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
+	BSR_1		<input type="checkbox"/>	BSR	
	Execute		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
+	Src		<input type="checkbox"/>	BOOL	[1..4]
	SrcOffset		0	N/D		<input type="checkbox"/>	UINT	
	BitAddr		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
	Length		0	N/D		<input type="checkbox"/>	UINT	
	Done_BSR		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
	Upload_BSR		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
	Error_BSR		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
	ErrorID_BSR		0	N/D		<input type="checkbox"/>	USINT	

Vedere anche

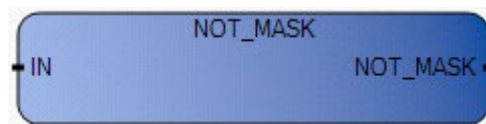
[Istruzioni binarie](#) a pagina 135

NOT_MASK (maschera NOT bit-to-bit)

Maschera negazione numero intero bit-to-bit, inverte il valore di un parametro.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

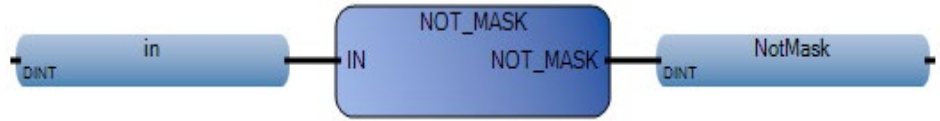


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

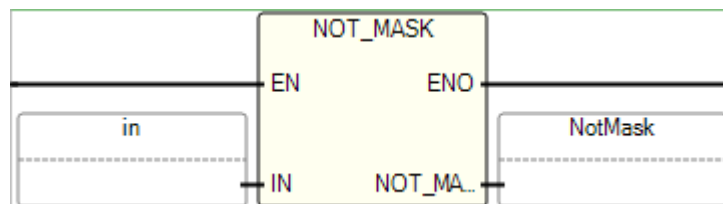
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della maschera negazione bit-to-bit. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.

IN	Ingresso	DINT	Deve avere il formato Integer.
NOT_MASK	Uscita	DINT	Negazione bit-to-bit su 32 bit di IN.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali NOT_MASK



Esempio di diagramma ladder NOT_MASK



Esempio di testo strutturato NOT_MASK

```
NOT_MASK(
  DINT NOT_MASK(DINT IN)
  negazione bit a bit
```

```
1 in := 6;
2 NotMask := NOT_MASK(in);
```

(* Equivalenza ST: *)

result := NOT_MASK (16#1234);

(* il risultato è 16#FFFF_EDCB *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
in	6	N/D		<input type="checkbox"/>	DINT	
NotMask	-7	N/D		<input type="checkbox"/>	DINT	

Vedere anche

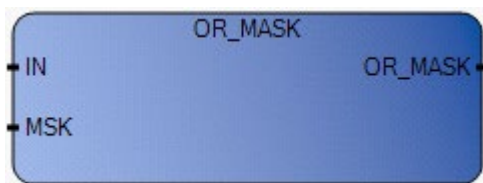
[Istruzioni binarie](#) a pagina 135

OR_MASK (maschera OR bit-to-bit)

Maschera OR numero intero bit-to-bit, abilita i bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

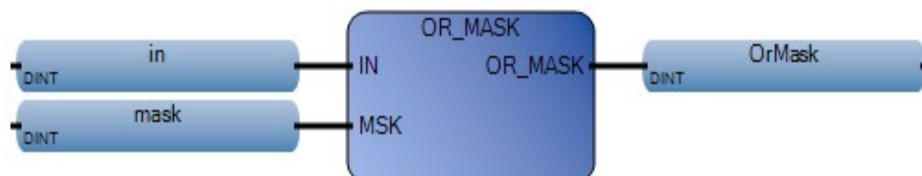
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



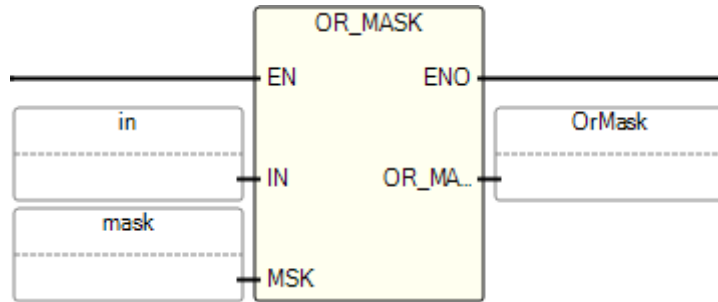
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della maschera per Integer OR bit-to-bit. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	DINT	Deve avere il formato Integer.
MSK	Ingresso	DINT	Deve avere il formato Integer.
OR_MASK	Uscita	DINT	Bit-to-bit logico OR tra IN e MSK.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali OR_MASK



Esempio di diagramma ladder OR_MASK



Esempi di testo strutturato OR_MASK

```
OR_MASK(
DINT OR_MASK(DINT IN, DINT MSK)
Mascheramento analogico OR bit a bit
```

```
1 in := 3;
2 mask := 6;
3 OrMask := OR_MASK(in, mask);
```

(* Equivalenza ST: *)

parity := OR_MASK(xvalue, 1); (* rende il valore sempre dispari *)

result := OR_MASK(16#abc, 16#f0f); (* uguale a 16#fbf *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
in	3	N/D		<input type="checkbox"/>	DINT	
mask	6	N/D		<input type="checkbox"/>	DINT	
OrMask	7	N/D		<input type="checkbox"/>	DINT	

Vedere anche

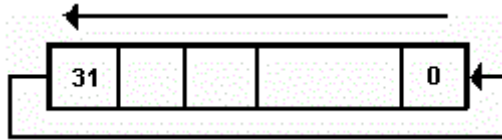
[Istruzioni binarie](#) a pagina 135

ROL (rotazione a sinistra)

Effettua la rotazione dell'input di tipo DINT di NbR bit a sinistra circolarmente e riempie i bit a destra con i bit fatti ruotare.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



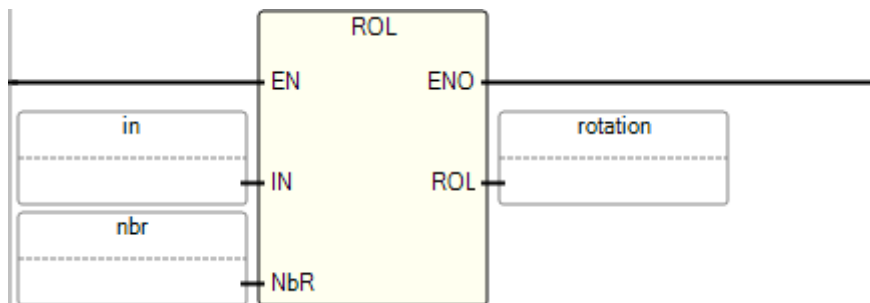
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della rotazione dei bit del valore Integer a sinistra. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	DINT	Valore integer.
NbR	Ingresso	DINT	Numero di rotazioni di 1 bit (in set [1..31]).
ROL	Uscita	DINT	Valore con rotazione a sinistra. Quando NbR <= 0, non avviene alcuna modifica.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali ROL



Esempio di Diagramma ladder ROL



Esempio di Testo strutturato ROL

```
ROL (
  DINT ROL(DINT IN, DINT Nbr)
  Ruota a sinistra
```

```
1 in := 123;
2 nbr := 2;
3 rotation := ROL(in, nbr);
```

(* Equivalenza ST: *)

result := ROL (register, 1);

(* registro = 2#0100_1101_0011_0101*)

(* risultato = 2#1001_1010_0110_1010*)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
rotation	492	N/D		<input checked="" type="checkbox"/>	DINT	
nbr	2	N/D		<input type="checkbox"/>	DINT	
in	123	N/D		<input type="checkbox"/>	DINT	

Vedere anche

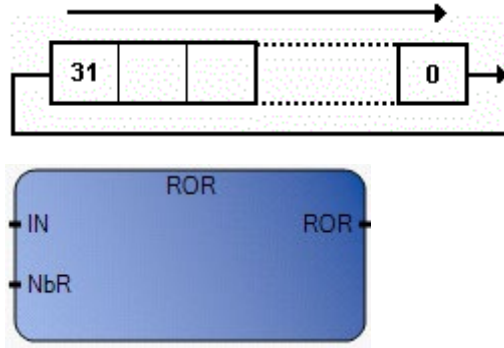
[Istruzioni binarie](#) a pagina 135

ROR (rotazione a destra)

Effettua la rotazione dell'input di tipo DINT di NbR bit a destra circolarmente e riempie i bit a sinistra con i bit fatti ruotare.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



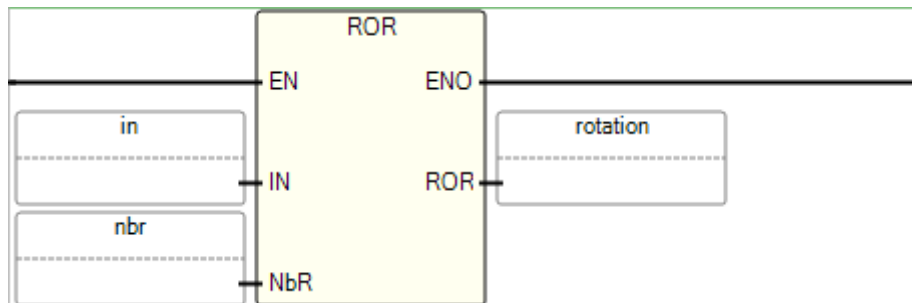
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della rotazione dei bit del valore Integer a destra. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	DINT	Qualsiasi valore intero.
NbR	Ingresso	DINT	Numero di rotazioni di 1 bit (in set [1..31]).
ROR	Uscita	DINT	Valore ruotato a destra. Non vi sono effetti se NbR <= 0.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali ROR



Esempio di Diagramma ladder ROR



Esempio di Testo strutturato ROR

```
ROR (
  DINT ROR(DINT IN, DINT Nbr)
  Ruota a destra
```

```
1 in := 123;
2 nbr := 2;
3 rotation := ROR(in, nbr);
```

(* Equivalenza ST: *)

risultato := ROR (registro, 1);

(* registro = 2#0100_1101_0011_0101 *)

(* risultato = 2#1010_0110_1001_1010 *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
rotation	-1073741794	N/D		<input checked="" type="checkbox"/>	DINT	
nbr	2	N/D		<input type="checkbox"/>	DINT	
in	123	N/D		<input type="checkbox"/>	DINT	

Vedere anche

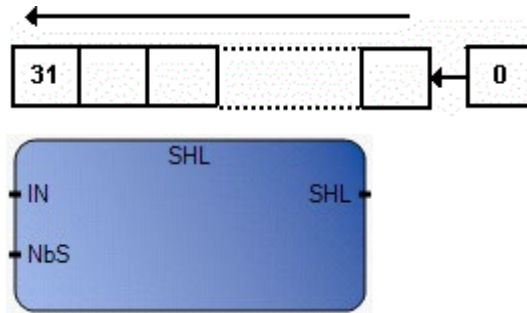
[Istruzioni binarie](#) a pagina 135

SHL (spostamento a sinistra)

Per gli integer a 32-bit, sposta gli integer a sinistra e inserisce 0 nel bit meno significativo.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



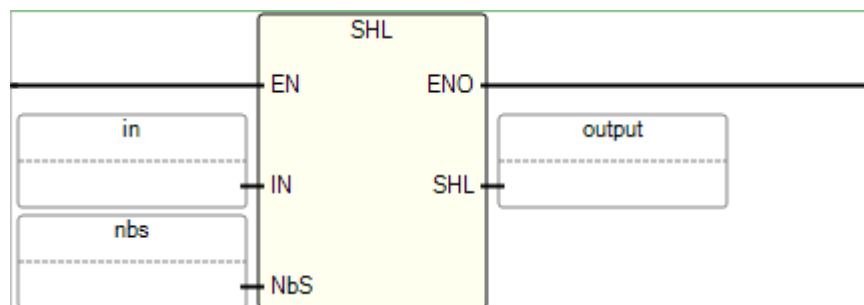
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: sposta gli integer a sinistra. FALSE: non vi è alcun movimento integer. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	DINT	Qualsiasi valore intero.
NbS	Ingresso	DINT	Numero di spostamenti (shift) di 1 bit (in set [1..31]).
SHL	Uscita	DINT	Valore spostato a sinistra. Non vi sono effetti se NbS <= 0. Se un valore di 0, sostituisce il bit meno significativo.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali SHL



Esempio di Diagramma ladder SHL



Esempio di Testo strutturato SHL

```
SHL (
  DINT SHL(DINT IN, DINT NbS)
  Scorri a sinistra
```

```
1 in := 123;
2 nbs := 2;
3 output := SHL(in, nbs);
```

(* Equivalenza ST: *)

risultato := SHL (registro,1);

(* registro = 2#0100_1101_0011_0101 *)

(* risultato = 2#1001_1010_0110_1010 *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
output	492	N/D		<input type="checkbox"/>	DINT
nbs	2	N/D		<input type="checkbox"/>	DINT
in	123	N/D		<input type="checkbox"/>	DINT

Vedere anche

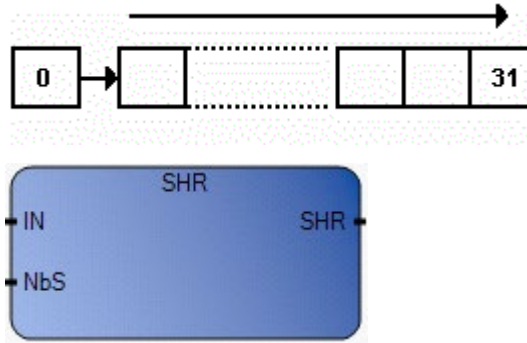
[Istruzioni binarie](#) a pagina 135

SHR (spostamento a destra)

Sposta a destra i 32 bit di un intero e replica il bit più a sinistra (bit significativo) per riempire i bit disponibili.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



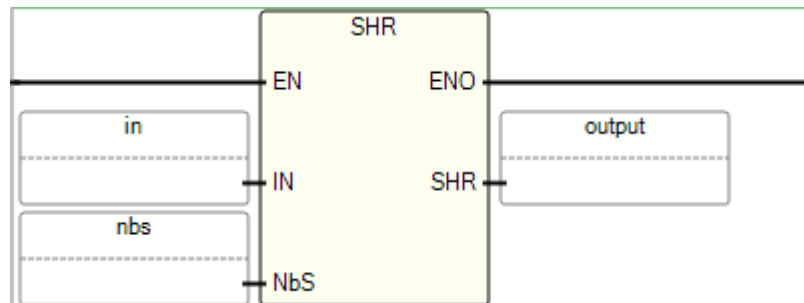
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: sposta gli integer a destra. FALSE: non vi è alcun movimento integer. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	DINT	Qualsiasi valore intero.
NbS	Ingresso	DINT	Numero di spostamenti (shift) di 1 bit (in set [1..31]).
SHR	Uscita	DINT	Valore spostato a destra. Non vi sono effetti se NbS <= 0. Se un valore di 0, sostituisce il bit più significativo.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali SHR



Esempio di Diagramma ladder SHR



Esempio di Testo strutturato SHR

```
SHR (
  DINT SHR(DINT IN, DINT NbS)
  Scorri a destra
```

```
1 in := 123;
2 nbs := 2;
3 output := SHR(in, nbs);
```

(* Equivalenza ST: *)

result := SHR (register,1);

(* registro = 2#1100_1101_0011_0101 *)

(* risultato = 2#0110_0110_1001_1010 *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
output	30	N/D		<input type="checkbox"/>	DINT
nbs	2	N/D		<input type="checkbox"/>	DINT
in	123	N/D		<input type="checkbox"/>	DINT

Vedere anche

[Istruzioni binarie](#) a pagina 135

XOR_MASK (maschera OR esclusivo)

Maschera OR intero esclusivo bit-to-bit, restituisce valori di bit invertiti.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

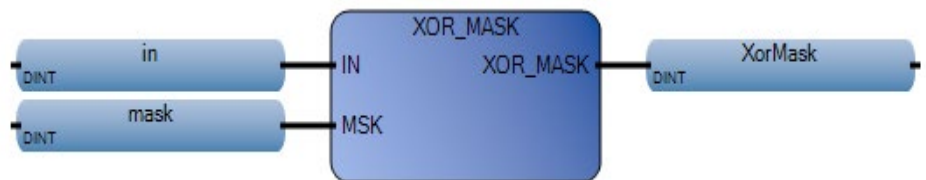
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



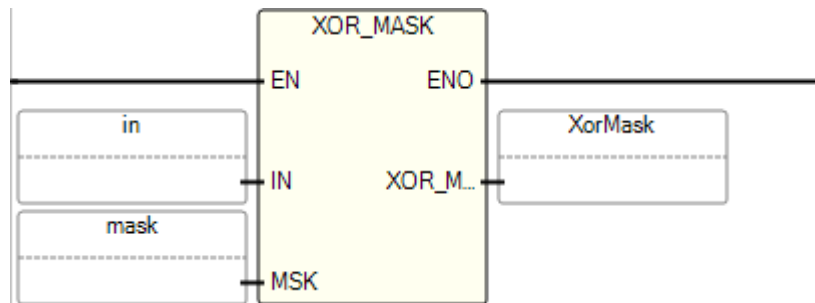
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della maschera per OR bit-to-bit esclusivo FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	DINT	Deve avere il formato Integer.
MSK	Ingresso	DINT	Deve avere il formato Integer.
XOR_MASK	Uscita	DINT	Bit-to-bit logico OR esclusivo tra IN e MSK.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali XOR_MASK



Esempio di diagramma ladder XOR_MASK



Esempio di testo strutturato XOR_MASK

```
XOR_MASK(
  DINT XOR_MASK(DINT IN, DINT MSK)
  Mascheramento analogico OR esclusivo bit a bit
```

```
1 in := 5;
2 mask := 6;
3 XorMask := XOR_MASK(in, mask);
```

(* Equivalenza ST: *)

crc32 := XOR_MASK (prevcrc, nextc);

result := XOR_MASK (16#012, 16#011); (* è uguale a 16#003 *)

Risultati

The screenshot shows a window titled "Monitoraggio della variabile" with a tab "Variabili locali - RA_XORMASK_LD". It contains a table with the following data:

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di da
XorMask	3	N/D		<input type="checkbox"/>	DINT
mask	6	N/D		<input type="checkbox"/>	DINT
in	5	N/D		<input type="checkbox"/>	DINT

At the bottom right of the window is a "Chiudi" button.

Vedere anche

[Istruzioni binarie](#) a pagina 135

Istruzioni booleane

Utilizzare le istruzioni booleane per determinare un valore di uscita in base a un calcolo logico partendo dagli ingressi. Le uscite del modulo possono essere controllate direttamente dal programma oppure in modo indipendente dal modulo, utilizzando le istruzioni booleane.

Funzione	Descrizione
MUX4B a pagina 175	Multiplexer tra quattro ingressi BOOL, restituisce un valore BOOL.
MUX8B a pagina 172	Multiplexer tra otto ingressi BOOL, restituisce un valore BOOL.
ITABLE a pagina 169	Fornisce il valore di uscita in base alla combinazione degli ingressi.
Blocco funzione	Descrizione
F_TRIG a pagina 159	Rileva un fronte di discesa di una variabile booleana.
RS a pagina 163	Reset bistabile dominante (priorità massima durante la definizione del comportamento delle istruzioni).
R_TRIG a pagina 161	Rileva un fronte di salita di una variabile booleana.
SR a pagina 167	Impostazione bistabile dominante.
Operatore	Descrizione
AND a pagina 165	Esegue un'operazione booleana AND tra due o più valori.
NOT a pagina 166	Converte i valori booleani in valori negati.
XOR a pagina 166	OR esclusivo booleano di due o più valori.
OR a pagina 164	OR booleano di due o più valori.

Vedere anche

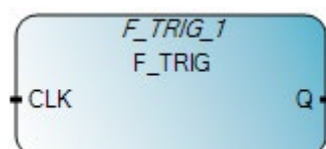
[Istruzione impostate in ordine alfabetico](#) a pagina 18

F_TRIG (rilevamento fronte di discesa)

Rileva un fronte di discesa di una variabile booleana. Il blocco F_TRIG imposta l'uscita Q per un ciclo quando l'ingresso CLK si attiva/disattiva da definito a risolto (per esempio, viene rilevato un fronte di discesa all'ingresso CLK).

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

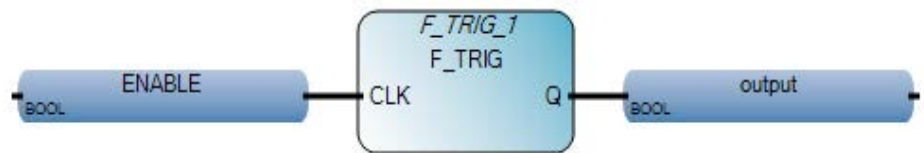
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



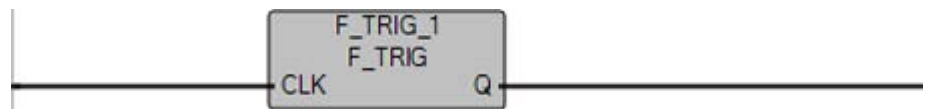
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
CLK	Ingresso	BOOL	Controlla l'ingresso per un fronte di discesa. Qualsiasi variabile booleana. TRUE = nessun fronte di discesa rilevato. FALSE = fronte di discesa rilevato sull'ingresso CLK; impostare l'uscita Q su TRUE.
Q	Uscita	BOOL	Indica lo stato dell'uscita Q. TRUE = fronte di discesa rilevato; impostare l'uscita Q per un ciclo in più. FALSE = nessuna modifica all'uscita Q.

Esempio di programmazione a blocchi funzionali F_TRIG



Esempio di diagramma ladder F_TRIG



Esempio di testo strutturato F_TRIG

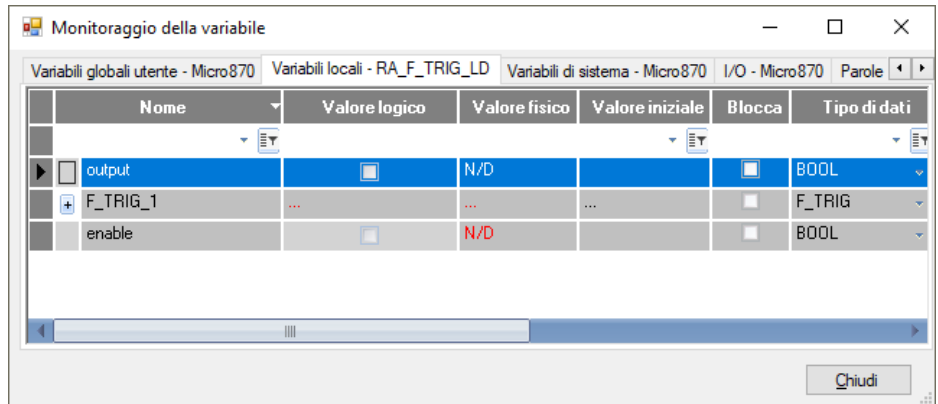
```
F_TRIG_1 (
void F_TRIG_1(BOOL CLK)
Tipo: F_TRIG, Individuazione fronte di discesa
```

```
1 F_TRIG_1(ENABLE);
2 IF F_TRIG_1.Q THEN
3   output := TRUE;
4 END_IF;
```

(* Equivalenza ST: F_TRIG1 è un'istanza del blocco F_TRIG *)

```
F_TRIG1(cmd);
nb_edge := ANY_TO_DINT(F_TRIG1.Q) + nb_edge;
```

Risultati



Vedere anche

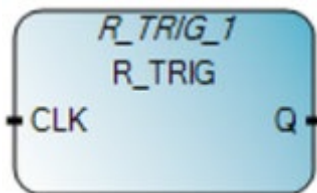
[Istruzioni booleane](#) a pagina 159

R_TRIG (rilevatore fronte di salita)

Rileva un fronte di salita di una variabile booleana. Il blocco R_TRIG imposta l'uscita Q per un ciclo quando l'ingresso CLK si attiva/disattiva da risolto a definito (per esempio, viene rilevato un fronte di salita all'ingresso CLK).

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

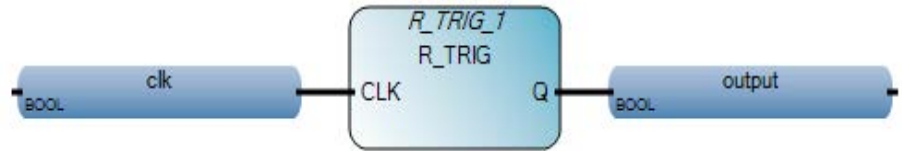
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



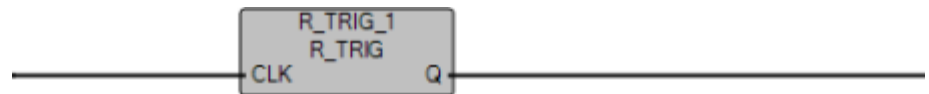
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
CLK	Ingresso	BOOL	Qualsiasi variabile booleana. TRUE: fronte di salita rilevato, imposta Q su TRUE. FALSE: nessun fronte di salita rilevato, imposta Q su FALSE.
Q	Uscita	BOOL	TRUE: quando CLK è TRUE. FALSE: in tutti gli altri casi.

Esempio di Programmazione a blocchi funzionali R_TRIG



Esempio di Diagramma ladder R_TRIG



Esempio di Testo strutturato R_TRIG

```
R_TRIG_1 (
void R_TRIG_1(BOOL CLK)
Tipo: R_TRIG, Individuazione fronte di salita
```

(* Equivalenza ST: R_TRIG1 è un'istanza del blocco R_TRIG *)

```
R_TRIG1 ( cmd ) ;
nb_edge := ANY_TO_DINT(R_TRIG1.Q) + nb_edge;
```

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
R_TRIG_1	<input type="checkbox"/>	R_TRIG	
output	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
clk	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	

Vedere anche

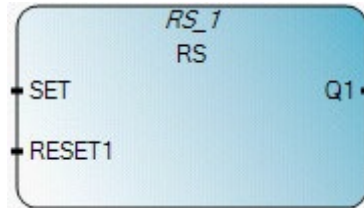
[Istruzioni booleane](#) a pagina 159

RS (reset/impostazione)

Ripristina o imposta bistabile dominante (priorità massima durante la definizione del comportamento delle istruzioni).

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



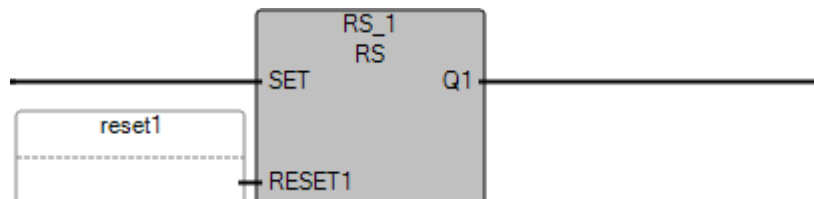
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
SET	Ingresso	BOOL	TRUE: imposta Q1 su TRUE.
RESET1	Ingresso	BOOL	TRUE: esegue il reset di Q1 su FALSE (dominante).
Q1	Uscita	BOOL	Stato memoria booleana.

Esempio di programmazione a blocchi funzionali RS



Esempio di Diagramma ladder RS



Esempio di Testo strutturato RS

```

RS_1 (
void RS_1(BOOL SET, BOOL RESET1)
Tipo: RS, Ripristina bistabile dominante

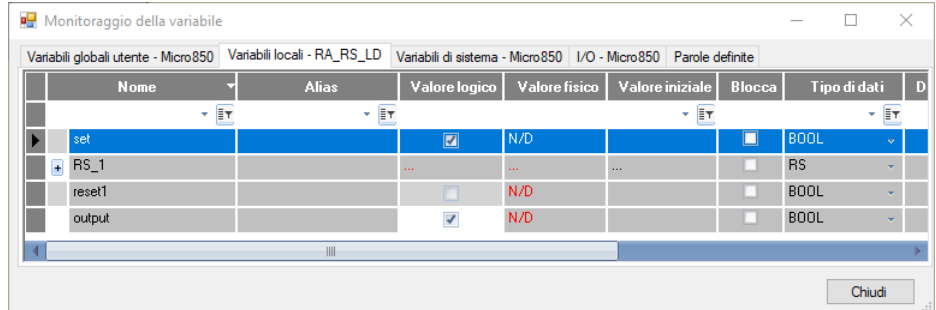
1: set := TRUE;
2: reset1 := FALSE;
3: RS_1(set, reset1);
4: output := RS_1.Q1;

```

(* Equivalenza ST: RS1 è un'istanza del blocco RS *)

```
RS1(start_cmd, (stop_cmd OR alarm));
command := RS1.Q1;
```

Risultati



Vedere anche

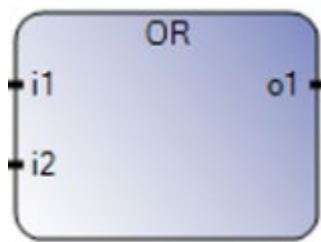
[Istruzioni booleane](#) a pagina 159

OR

Esegue un'operazione OR logica di due o più valori booleani e restituisce il valore booleano True se almeno un ingresso è True, altrimenti torna False.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
i1	Ingresso	BOOL	
i2	Ingresso	BOOL	
o1	Uscita	BOOL	OR booleano dei termini di ingresso. TRUE: quando uno o più ingressi sono TRUE. FALSE: quando gli ingressi sono FALSE.

Esempio di testo strutturato OR

(* Equivalenza ST: *)

```
bo10 := bi101 OR NOT (bi102);
bo5 := (bi51 OR bi52) OR bi53;
```

Vedere anche

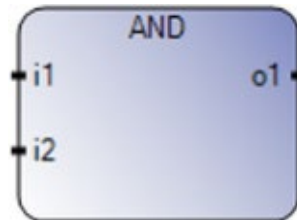
[Istruzioni booleane](#) a pagina 159

AND

Esegue un'operazione booleana AND tra due o più valori.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
i1	Ingresso	BOOL	Valore in tipo di dati Boolean.
i2	Ingresso	BOOL	Valore in tipo di dati Boolean.
o1	Uscita	BOOL	Risultato dell'operazione booleana AND dei valori di ingresso.

Esempio di testo strutturato AND

(* Equivalenza ST: *)

```
bo10 := bi101 AND NOT (bi102);
bo5 := (bi51 AND bi52) AND bi53;
```

Vedere anche

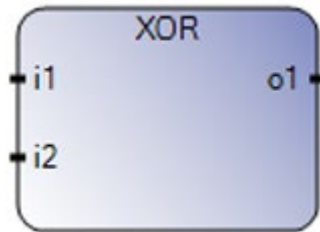
[Istruzioni booleane](#) a pagina 159

XOR (OR esclusivo)

Esegue un'operazione OR esclusiva tra due valori booleani.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
i1	Ingresso	BOOL	
i2	Ingresso	BOOL	
o1	Uscita	BOOL	OR esclusivo booleano dei due termini di ingresso. TRUE: quando uno o entrambi gli ingressi sono TRUE. FALSE: quando entrambi gli ingressi sono FALSE.

Esempio di Testo strutturato XOR

(* Equivalenza ST: *)

```
bo10 := bi101 XOR NOT (bi102);
bo5 := (bi51 XOR bi52) XOR bi53;
```

Vedere anche

[Istruzioni booleane](#) a pagina 159

NOT

Converte i valori booleani in valori negati.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
i1	Ingresso	BOOL	Qualsiasi valore Booleano o espressione complessa.
o1	Uscita	BOOL	TRUE quando IN è FALSE. FALSE quando IN è TRUE.

Esempio di Testo strutturato NOT

(* Equivalenza ST: *)

```
bo10 := NOT (bi101);
```

Vedere anche

[Istruzioni booleane](#) a pagina 159

SR (impostazione/reset)

Imposta un bistabile dominante.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
SET1	Ingresso	BOOL	TRUE: imposta Q1 su TRUE (dominante: priorità massima durante la definizione del comportamento delle istruzioni)
RESET	Ingresso	BOOL	TRUE: esegue il reset di Q1 su FALSE.
Q1	Uscita	BOOL	Stato memoria booleana. TRUE: quando SET1 è TRUE. FALSE: quando RESET è TRUE.

Esempio bistabile dominante

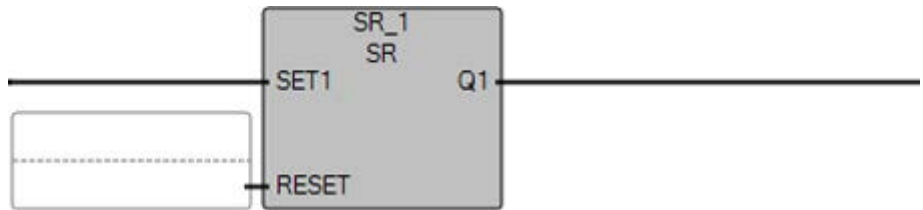
Set1	Reset	Q1	Risultato Q1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1

1	0	1	1
1	1	0	1
1	1	1	1

Esempio di programmazione a blocchi funzionali SR



Esempio di Diagramma ladder SR



Esempio di Testo strutturato SR

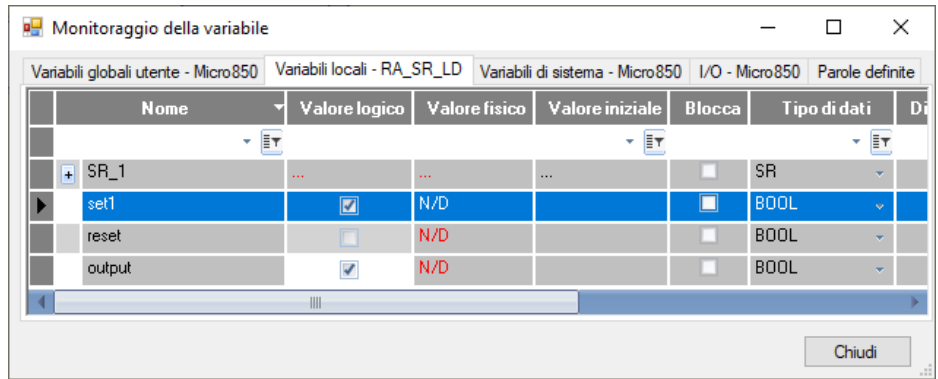
```
SR_1(
void SR_1(BOOL SET1, BOOL RESET)
Tipo: SR, Imposta bistabile dominante
```

```
1 set1 := TRUE;
2 reset := FALSE;
3 SR_1(set1, reset);
4 output := SR_1.Q1;
```

(* Equivalenza ST: SR1 è un'istanza del blocco SR *)

```
SR1((auto_mode & start_cmd), stop_cmd);
command := SR1.Q1;
```

Risultati



Vedere anche

[Istruzioni booleane](#) a pagina 159

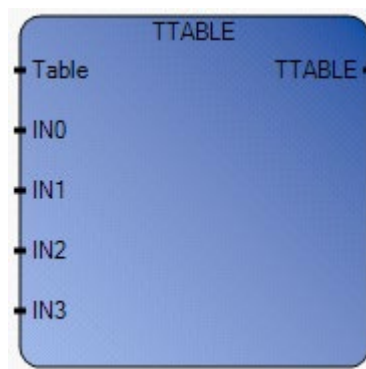
TTABLE (tavola della verità)

Fornisce il valore di uscita in base alla combinazione degli ingressi.

Se il valore è 0xABCD e gli ingressi da In3 a In0 corrispondono al numero 7, TTABLE è il valore del bit 7 nella tabella (che è 1). Il bit meno significativo nella tabella è il bit 0.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Tabella	Ingresso	UINT	Tabella valori veri per la funzione BOOLEAN.
IN0	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL.
IN1	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL.
IN2	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL.
IN3	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL.
TTABLE	Uscita	BOOL	Il valore di uscita in base alle combinazioni degli ingressi.

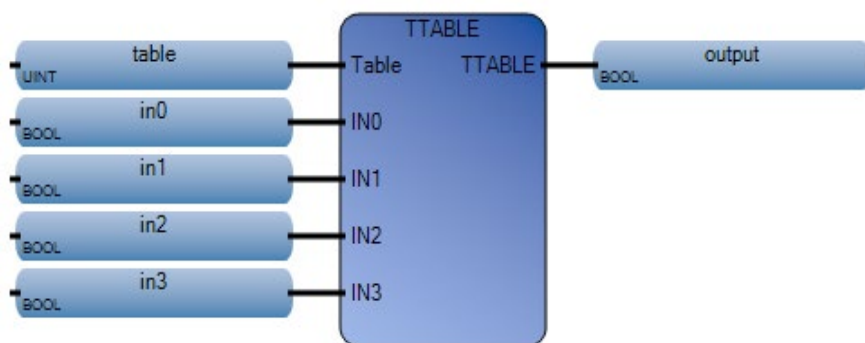
Combinazioni di ingressi TTABLE

L'istruzione TTABLE ha quattro ingressi, e pertanto 16 combinazioni. Tali combinazioni trovano posto in una tavola della verità: per ogni combinazione è possibile regolare il valore in uscita. Il numero di combinazioni configurabili dipende dal numero di ingressi collegati alla funzione.

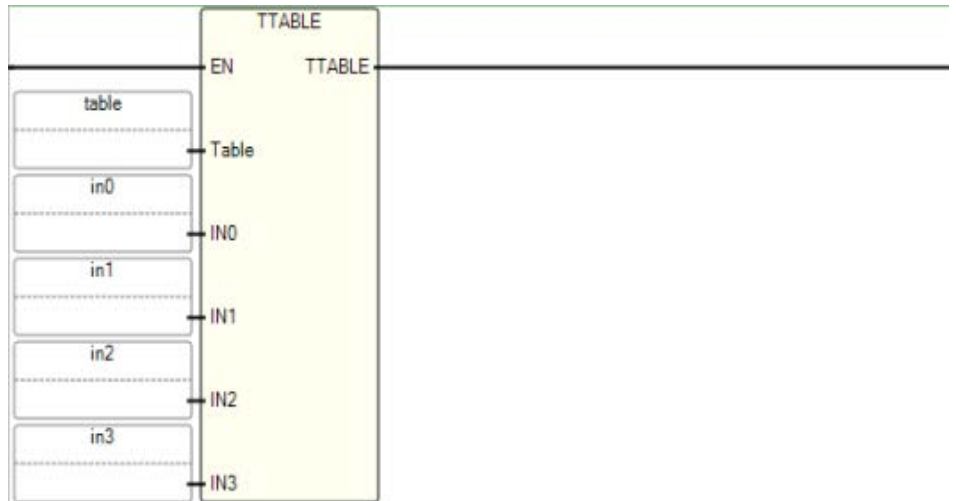
Esempio di combinazione tavola della verità.

Numero	In3	In2	In1	In0
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	1	1	1
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Esempio di programmazione a blocchi funzionali TTABLE



Esempio di Diagramma ladder TTABLE



Esempio di Testo strutturato TTABLE

```
TTABLE {
  BOOL TTABLE(UINT Table, BOOL IN0, BOOL IN1, BOOL IN2, BOOL IN3)
  Fornire l'uscita del valore in base alla combinazione di ingressi.
```

```
1 table := 217;
2 in0 := TRUE;
3 in1 := TRUE;
4 in2 := TRUE;
5 in3 := FALSE;
6 output := TTABLE(table, in0, in1, in2, in3);
```

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dato
in0	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
table	217	N/D		<input type="checkbox"/>	UINT
in1	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
in2	<input checked="" type="checkbox"/>	N/D		<input checked="" type="checkbox"/>	BOOL
in3	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
output	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL

Vedere anche

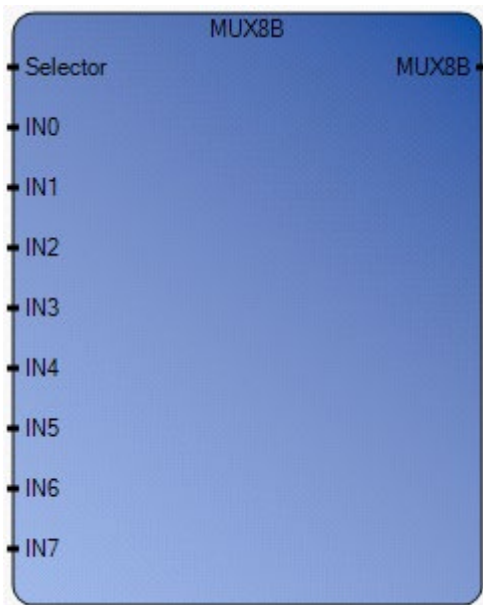
[Istruzioni booleane](#) a pagina 159

MUX8B (multiplexer di ingressi BOOL 8)

Multiplexer tra otto ingressi BOOL, restituisce un valore BOOL.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

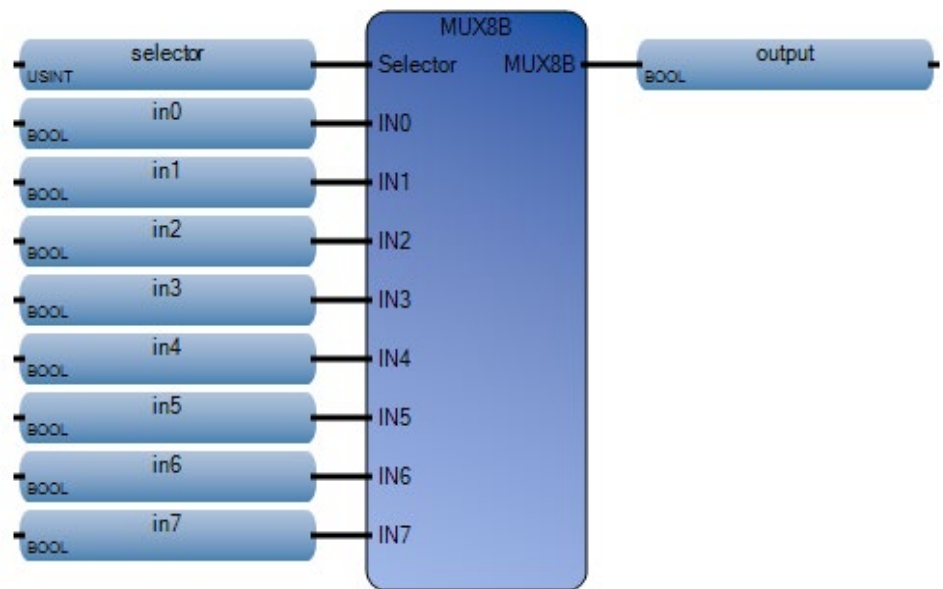


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

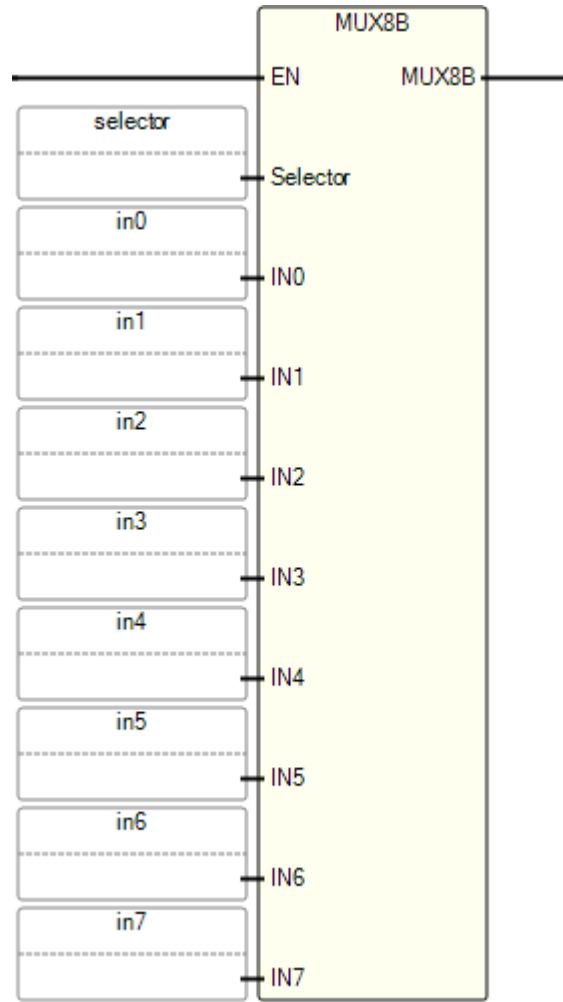
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Selettore	Ingresso	USINT	Il valore intero del selettore deve essere compreso nell'intervallo [0..7].
IN0	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 0. FALSE: quando il selettore non è 0.
IN1	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 1. FALSE: quando il selettore non è 1.
IN2	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 2. FALSE: quando il selettore non è 2.
IN3	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 3. FALSE: quando il selettore non è 3.
IN4	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 4. FALSE: quando il selettore non è 4.
IN5	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 5. FALSE: quando il selettore non è 5.

IN6	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 6. FALSE: quando il selettore non è 6.
IN7	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 7. FALSE: quando il selettore non è 7.
MUX8B	Uscita	BOOL	TRUE: quando: <ul style="list-style-type: none"> • In0 se selettore = 0 • In1 se selettore = 1 • In2 se selettore = 2 • In3 se selettore = 3 • In4 se selettore = 4 • In5 se selettore = 5 • In6 se selettore = 6 • In7 se selettore = 7 FALSE: per tutti gli altri valori del selettore.

Esempio di programmazione a blocchi funzionali MUX8B



Esempio di diagramma ladder MUX8B



Esempio di Testo strutturato MUX8B

MUX8B |
 BOOL **MUX8B**(USINT Selector, BOOL IN0, BOOL IN1, BOOL IN2, BOOL IN3, BOOL IN4, BOOL IN5, BOOL IN6, BOOL IN7)
 Multiplexer (8 ingressi) - accetta valori BOOL di ingresso e uscita.

```

1 selector := 7;
2 in0 := FALSE;
3 in1 := FALSE;
4 in2 := FALSE;
5 in3 := FALSE;
6 in4 := FALSE;
7 in5 := FALSE;
8 in6 := FALSE;
9 in7 := TRUE;
10 output := MUX8B(selector, in0, in1, in2, in3, in4, in5, in6, in7);
    
```

(* Equivalenza ST: *)

range := MUX8 (choice, 1, 5, 10, 50, 100, 500, 1000, 5000);

(* sceglie tra 8 intervalli predefiniti; ad esempio se choice è 3, range sarà 50 *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
selector	7	N/D		<input type="checkbox"/>	USINT	
output	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in7	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in6	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in5	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in4	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in3	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in2	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in1	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in0	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	

Vedere anche

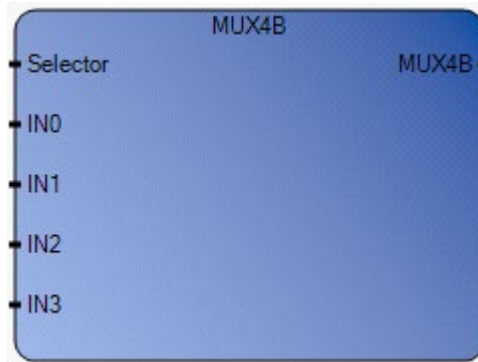
[Istruzioni booleane](#) a pagina 159

MUX4B (multiplexer di ingressi BOOL 4)

Multiplexer tra quattro ingressi BOOL, restituisce un valore BOOL.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

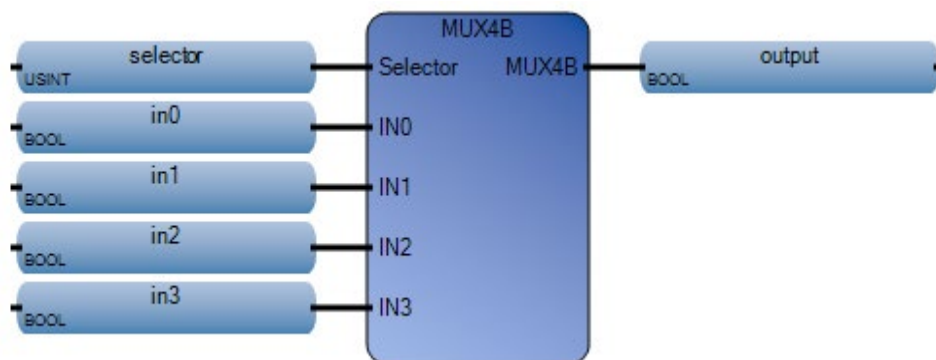


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

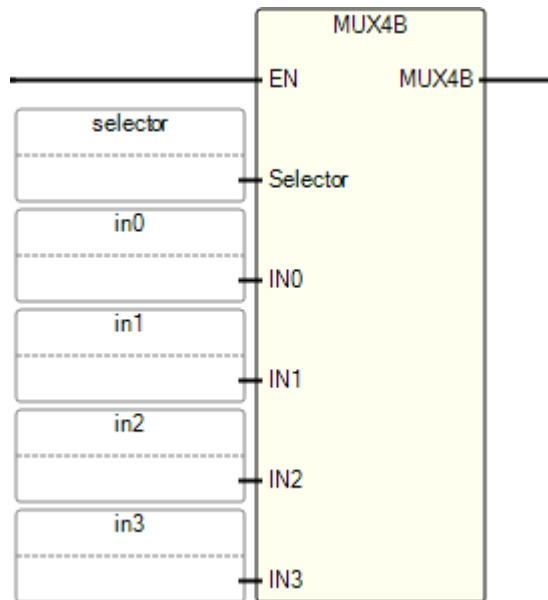
Parametro	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

Selettore	Ingresso	USINT	Il valore intero del selettore deve essere compreso nell'intervallo [0...3].
IN0	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 0. FALSE: quando il selettore non è 0.
IN1	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 1. FALSE: quando il selettore non è 1.
IN2	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 2. FALSE: quando il selettore non è 2.
IN3	Ingresso	BOOL	Qualsiasi valore di ingresso BOOL. TRUE: quando il selettore è 3. FALSE: quando il selettore non è 3.
MUX4B	Uscita	BOOL	TRUE: quando: <ul style="list-style-type: none"> • In0 se selettore = 0 • In1 se selettore = 1 • In2 se selettore = 2 • In3 se selettore = 3 FALSE: per tutti gli altri valori del selettore.

Esempio di programmazione a blocchi funzionali MUX4B



Esempio di diagramma ladder MUX4B



Esempio di Testo strutturato MUX4B

```
MUX4B (
  BOOL MUX4B(USINT Selector, BOOL IN0, BOOL IN1, BOOL IN2, BOOL IN3)
  Multiplexer (4 ingressi) - accetta valori BOOL di ingresso e uscita.
```

```

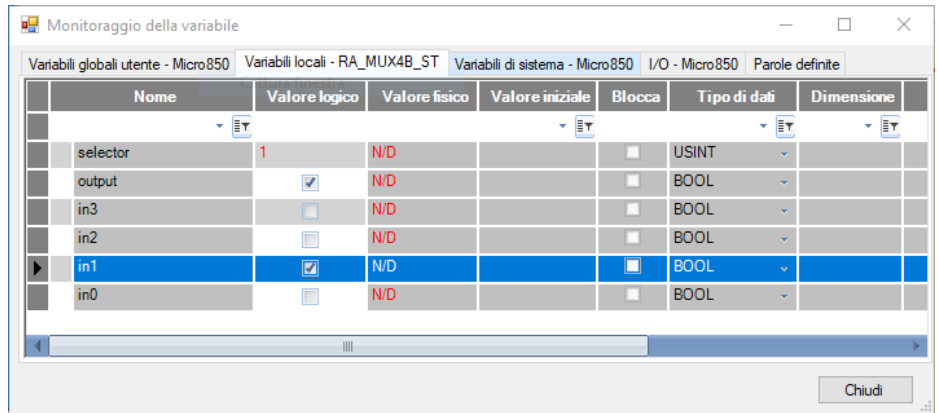
1 selector := 1;
2 in0 := FALSE;
3 in1 := TRUE;
4 in2 := FALSE;
5 in3 := FALSE;
6 output := MUX4B(selector, in0, in1, in2, in3);
```

(* Equivalenza ST: *)

range := MUX4 (scelta, 1, 10, 100, 1000);

(* sceglie tra 4 intervalli predefiniti; ad esempio se choice è 1, range sarà 10 *)

Risultati



The screenshot shows a window titled "Monitoraggio della variabile" with several tabs: "Variabili globali utente - Micro850", "Variabili locali - RA_MUX4B_ST", "Variabili di sistema - Micro850", "I/O - Micro850", and "Parole definite". The "Variabili locali - RA_MUX4B_ST" tab is active, displaying a table with the following columns: "Nome", "Valore logico", "Valore fisico", "Valore iniziale", "Blocca", "Tipo di dati", and "Dimensione". The table contains six rows of data:

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
selector	1	N/D		<input type="checkbox"/>	USINT	
output	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in3	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in2	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in1	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
in0	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	

A "Chiudi" button is located at the bottom right of the window.

Vedere anche

[Istruzioni booleane](#) a pagina 159

Istruzioni di comunicazione

Usare le istruzioni di comunicazione per leggere, scrivere, confrontare e convertire stringhe di comunicazione.

Blocco funzione	Descrizione
COM_IO_WDOG a pagina 179	Monitora le comunicazioni con il controllore.
MSG_CIPGENERIC a pagina 182	Invia un messaggio CIP generico esplicito.
MSG_CIPSYMBOLIC a pagina 191	Invia un messaggio CIP Symbolic esplicito.
MSG_MODBUS a pagina 196	Invia un messaggio Modbus.
MSG_MODBUS2 a pagina 202	Invia un messaggio MODBUS/TCP su un canale Ethernet.

Vedere anche

[Supporto per protocolli di comunicazione](#) a pagina 240

[Configurazione dei valori dei dati oggetto per i messaggi espliciti](#) a pagina 215

[Istruzione impostate in ordine alfabetico](#) a pagina 18

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

[Utilizzo dei blocchi funzione di comunicazione \(messaggistica\)](#) a pagina 215

COM_IO_WDOG

Monitora i messaggi esterni agli ingressi e alle uscite del controllore. Ad esempio, se il comando di scrittura CIP alla variabile `_IO_EM_DO_00` non viene ricevuto tramite EtherNet/IP nell'intervallo di timeout configurato, il timer watchdog scadrà e tutte le uscite delle controllore verranno ripristinate.

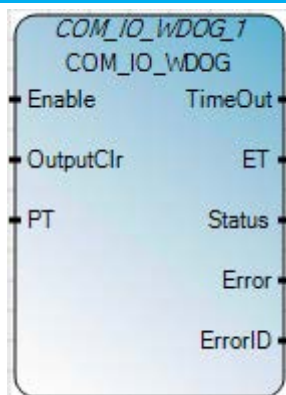
Sono supportati i protocolli EtherNet/IP, Modbus TCP e Modbus RTU.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator. È supportata solo con firmware versione 12.00 e successivi.

Nota:

- sebbene ci siano diverse istanze per questa istruzione, solo una può essere abilitata. In caso contrario, sarà generato un errore con ErrorID impostato su 2.
- Sono supportati solo ingressi/uscite digitale integrati nel controllore, di plug-in e di espansione.



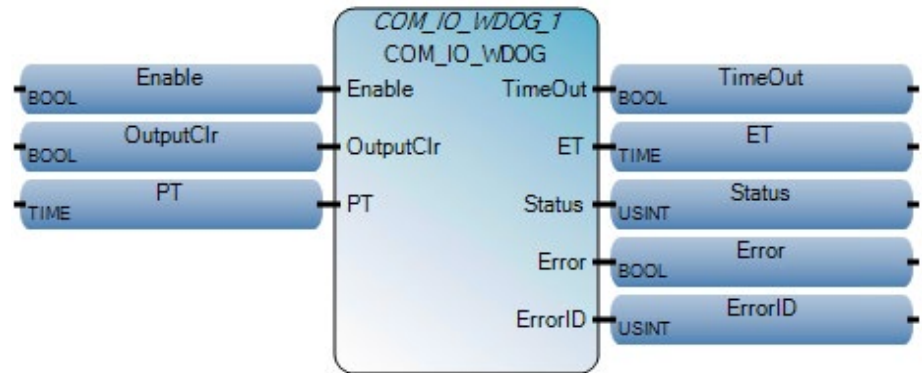
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	<p>TRUE: di seguito i tre casi della condizione TRUE.</p> <ol style="list-style-type: none"> 1. Sul fronte di salita di TRUE, viene avviata l'esecuzione del blocco funzione. Il bit Stato è impostato su 0 e gli altri bit vengono cancellati. Verificare le condizioni di errore seguenti. Se non si verificano errori, procedere con il passaggio successivo. <ul style="list-style-type: none"> • Se PresetValue è inferiore a un secondo, Error è impostato su TRUE ed ErrorID è impostato su 1, il bit Stato 4 è impostato, TimeOut ed ElapsedTime sono cancellati. • Se il timer è già stato acquisito dall'altra istanza, Error è impostato su TRUE ed ErrorID è impostato su 2, il bit Stato 4 è impostato, TimeOut ed ElapsedTime sono cancellati. 2. PresetTime > ElapsedTime. Se Enable è impostato su TRUE e viene avviata l'esecuzione del blocco funzione. Verifica se vengono ricevuti i comandi. Se vengono ricevuti i comandi, eseguire il reset del timer immediatamente. Impostare ElapsedTime su 0 e impostare il bit Stato su 2. Gli altri bit vengono cancellati. Se non vengono ricevuti comandi, impostare il bit di stato e gli altri bit vengono cancellati. 3. PresetTime = ElapsedTime. Se Enable è impostato su TRUE e viene avviata l'esecuzione del blocco funzione. Verificare se vengono ricevuti i comandi. Se vengono ricevuti, ripristinare il timer immediatamente. Impostare ElapsedTime su 0 e impostare il bit di stato 2. Gli altri bit vengono cancellati. In caso contrario, imposta il bit TimeOut, imposta il bit di stato e gli altri bit vengono cancellati. Impostare la variabile interna per cancellare le uscite digitali al termine della scansione se l'azione configurata è quella di eliminare tutte le uscite digitali al termine della scansione. <p>FALSE: l'istruzione non viene eseguita e le uscite del blocco funzione vengono cancellate.</p>
OutputClr	Ingresso	BOOL	<p>0: nessuna azione in caso di timeout.</p> <p>1: cancella tutte le uscite digitali al termine della scansione (integrata, EXIO e UPM) in caso di timeout.</p>
PT	Ingresso	TIME	<p>Tempo di attesa prima del timeout.</p> <p>Il valore di un timeout non può essere inferiore a un secondo; in caso contrario si verificherà un errore. Il valore massimo per PresetTime può essere il valore massimo entro il tipo di dati TIME.</p>
TimeOut	Uscita	BOOL	<p>TRUE: ElapsedTime è uguale a PresetTime.</p> <p>FALSE: Enable è impostato su FALSE, il timer non è trascorso o si è verificato un errore.</p>
ET	Uscita	TIME	<p>Il tempo trascorso.</p> <p>I valori possibili variano da 0 ms a 1193h2m47s294ms.</p>
Stato	Uscita	USINT	<p>Stato del blocco funzione.</p> <p>Bit 0: Enable</p> <p>Bit 1: timer in esecuzione, nessun ingresso o uscita ricevuti.</p> <p>Bit 2: comando ingresso o uscita ricevuto.</p> <p>Bit 3: timeout. Nessun comando di ingresso o uscita ricevuto.</p> <p>Bit 4: si è verificato un errore.</p> <p>Gli altri bit sono riservati.</p>
Errore	Uscita	BOOL	Indica che si è verificato un errore.
ErrorID	Uscita	USINT	Se si verifica un errore, ErrorID contiene il codice dell'errore.

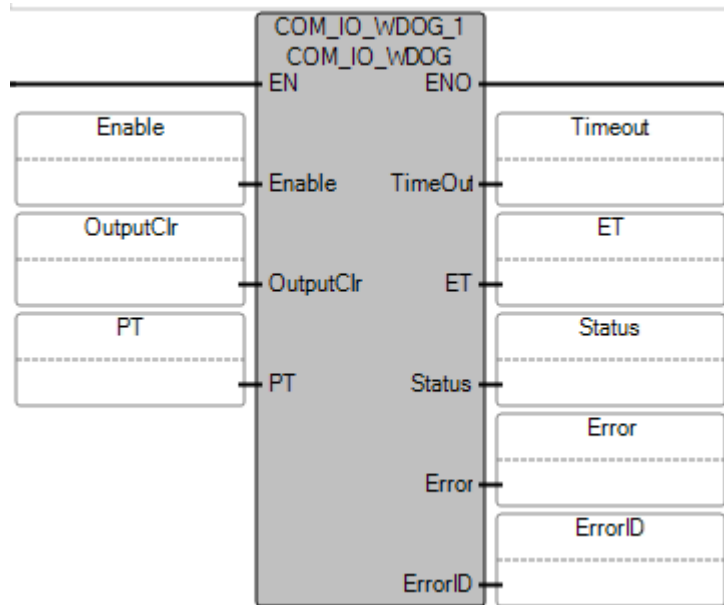
Codice errore COM_IO_WDOG

Codice ErrorID	Descrizione errore
1	PresetTime è inferiore a un secondo.
2	Un'altra istanza del blocco funzione COM_IO_WDOG è già in esecuzione.

Esempio di Programmazione a blocchi funzionali COM_IO_WDOG



Esempio di Diagramma ladder COM_IO_WDOG



Esempio di Testo strutturato COM_IO_WDOG

```
COM_IO_WDOG_1(
void COM_IO_WDOG_1(BOOL Enable, BOOL OutputClr, TIME PT)
Tipo: COM_IO_WDOG, COM_IO_WDOG permette all'utente di monitorare le comunicazioni con il controllore
```

```
1 COM_IO_WDOG_1(Enable, OutputClr, PT);
2 Timeout :=COM_IO_WDOG_1.Timeout;
3 ET :=COM_IO_WDOG_1.ET;
4 Status :=COM_IO_WDOG_1.Status;
5 Error :=COM_IO_WDOG_1.Error;
6 ErrorID :=COM_IO_WDOG_1.ErrorID;
```

Vedere anche

[Istruzioni di comunicazione](#) a pagina 179

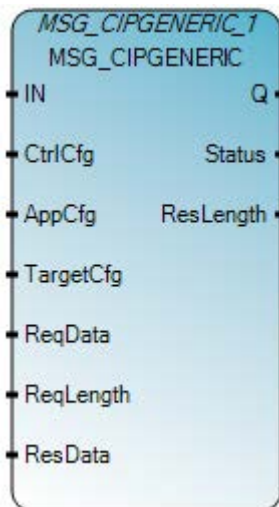
MSG_CIPGENERIC (messaggio generico per Common Industrial Protocol)

Invia un messaggio CIP (Common Industrial Protocol) esplicito tramite un canale Ethernet o una porta seriale.

In una scansione è possibile elaborare un massimo di quattro richieste di messaggio per canale. Per le programmazioni in ladder diagram, le richieste di messaggio sono eseguite al termine di una scansione ladder.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

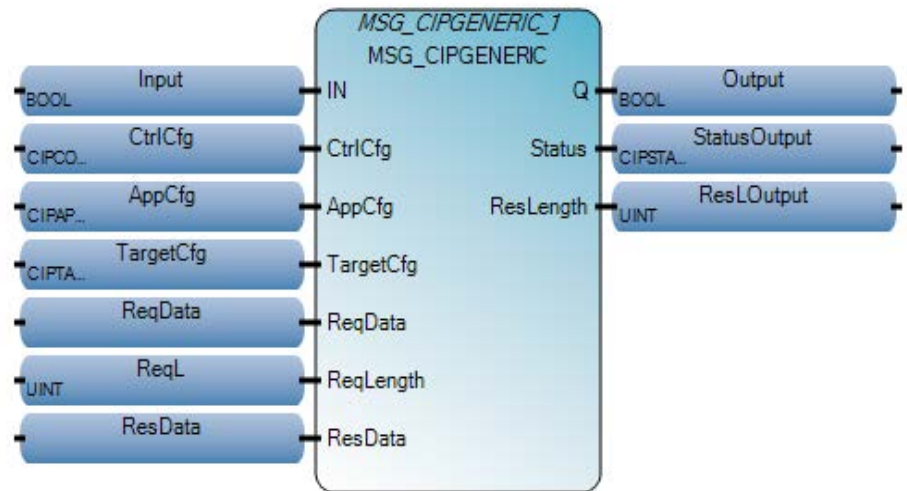


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

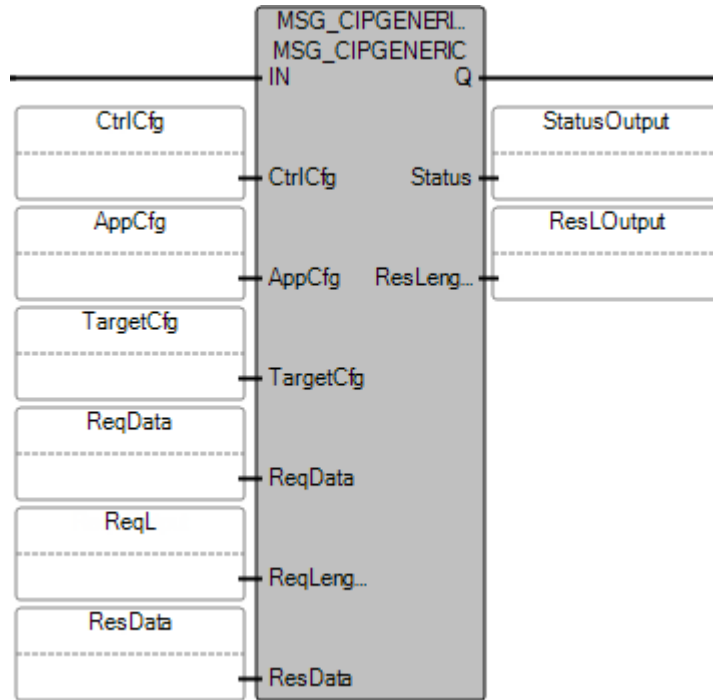
Parametro	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: fronte di salita non rilevato, non attivo.
CtrlCfg	Ingresso	CIPCONTROLCFG	Configurazione per il controllo dell'esecuzione del blocco funzione. Utilizzare i parametri del tipo di dati CIPCONTROLCFG per definire CtrlCfg.
AppCfg	Ingresso	CIPAPPCFG	Configurazione del servizio CIP e del percorso dell'applicazione (EPATH). Utilizzare i parametri del tipo di dati CIPAPPCFG per definire AppCfg.
TargetCfg	Ingresso	CIPTARGETCFG	Configurazione del dispositivo di destinazione. Utilizzare i parametri del tipo di dati CIPTARGETCFG per definire TargetCfg.
ReqData	Ingresso	USINT[1..1]	Dati della richiesta messaggio CIP. La dimensione del vettore non deve essere maggiore della dimensione ReqLength.
ReqLength	Ingresso	UINT	Lunghezza dei dati della richiesta messaggio CIP: • 0 - 490
ResData	Ingresso	USINT[1..1]	Dati della risposta messaggio CIP. La dimensione del vettore non deve essere maggiore della dimensione ReqLength. Quando un MSG è attivato o riattivato, i dati dell'array ResData vengono cancellati.
Q	Uscita	BOOL	Gli output di questa istruzione sono aggiornati in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: istruzione MSG terminata con successo. FALSE: istruzione MSG non terminata.
Stato	Uscita	CIPSTATUS	Lo stato del blocco istruzione. Quando è attivato o riattivato un MSG, tutti gli elementi in Stato vengono ripristinati. L'uscita dello stato è definita nel tipo di dati CIPSTATUS.
ResLength	Uscita	UINT	Lunghezza dei dati della risposta messaggio CIP: • 0 - 490 Quando un MSG è attivato o riattivato, ResLength viene ripristinato su 0.

Esempio di Programmazione a blocchi funzionali MSG_CIPGENERIC



Esempio di Diagramma ladder MSG_CIPGENERIC



Esempio di Testo strutturato MSG_CIPGENERIC

```
MSG_CIPGENERIC_1(in1,Ctrl1,appl,target1,request1,length1,data1);
out1 :- MSG_CIPGENERIC_1.Q;
status1 :- MSG_CIPGENERIC_1.Status;
resleng1 :- MSG_CIPGENERIC_1.ResLength;
```

```
MSG_CIPGENERIC 2(|
void MSG_CIPGENERIC 2(BOOL IN, CIPCONTROLCFG CtrlCfg, CIPAPPCFG AppCfg, CIPTARGETCFG TargetCfg, USINT[1..1] ReqData, USINT ReqLength, USINT[1..1] ResData)
Tipo: MSG_CIPGENERIC, Inviare un messaggio CIP esplicito.
```

Vedere anche

- [Tipo di dati CIPCONTROLCFG a pagina 185](#)
- [Tipo di dati CIPAPPCFG a pagina 184](#)
- [Tipo di dati CIPTARGETCFG a pagina 188](#)
- [Tipo di dati CIPSTATUS a pagina 186](#)
- [Esempio: Come creare un programma di generazione messaggi MSG_CIPGENERIC per leggere i dati da un controllore a pagina 217](#)

Tipo di dati CIPAPPCFG

Utilizzare questa tabella per definire i parametri per il tipo di dati CIPAPPCFG.

Parametri	Tipo di dati	Descrizione
Servizio	USINT	Codice servizio: 1 - 127

Parametri	Tipo di dati	Descrizione
Classe	UINT	Valore ID della classe del segmento logico: 1 - 65535
Istanza	UDINT	Valore ID dell'istanza del segmento logico: 0 - 4294967295
Attributo	UINT	Valore ID dell'attributo del segmento logico: 1 - 65535, 0 - ID attributo non utilizzato
MemberCnt	USINT	Conteggio ID membri. Valori ID membro massimi utilizzati: 1 - 3, 0 - ID membro non utilizzato
MemberId	UINT[3]	Valori ID membro: 0 - 65535

Vedere anche

[MSG_CIPSYMBOLIC](#) a pagina 191

[MSG_CIPGENERIC](#) a pagina 182

Tipo di dati CIPCONTROLCFG Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati CIPCONTROLCFG.

Parametri	Tipo di dati	Descrizione
Annulla	BOOL	TRUE: annulla l'esecuzione del blocco funzione. Il bit è cancellato quando viene abilitato il messaggio. Se è impostato il parametro Cancel e il messaggio è abilitato (bit EN impostato) e non eseguito (bit DN non impostato), l'esecuzione del messaggio viene interrotta e viene impostato il bit ER.
TriggerType	USINT	Rappresenta uno dei seguenti: <ul style="list-style-type: none"> • 0: Msg attivato una volta (quando IN passa da False a True) • Da 1 a 65535: valore attivazione ciclica in millisecondi. Msg è attivato periodicamente quando IN è True. Impostare il valore su 1, per attivare MSG il più rapidamente possibile.
StrMode	USINT	Riservato per uso futuro.

Attivazione messaggio CIP

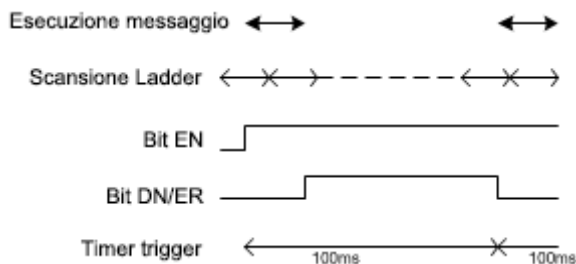
Un messaggio CIP può essere attivato periodicamente impostando il parametro TriggerType su un valore diverso da zero.

Utilizzare questa tabella per definire le azioni per il parametro Tipo di trigger.

Azione	Risultati
Il messaggio è abilitato.	Il timer attivazione si attiva.
Il tempo impostato nel timer attivazione è esaurito prima del completamento del messaggio.	Il messaggio è attivato immediatamente nel ciclo di scansione ladder successivo.
Il messaggio viene completato prima che si esaurisca il tempo impostato nel timer attivazione.	Il messaggio è attivato quando il tempo impostato nel timer attivazione è esaurito.

Esempio: attivazione messaggio

Nell'esempio seguente il valore TriggerType è impostato su 100.



Vedere anche

[MSG_CIPSYMBOLIC](#) a pagina 191

[MSG_CIPGENERIC](#) a pagina 182

Tipo di dati CIPSTATUS

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati CIPSTATUS.

Parametri	Tipo di dati	Descrizione
Errore	BOOL	Questo bit è impostato su TRUE quando viene rilevata una condizione di errore durante l'esecuzione del blocco funzione.
ErrorID	UINT	Valore codice di errore. Gli ErrorID sono definiti nei codici di errore CIPSTATUS.
SubErrorID	UINT	Valore codice di sottoerrore. I SubErrorID sono definiti nei codici di errore CIPSTATUS.
ExtErrorID	UINT	Valore codice di errore per stato CIP esteso.
StatusBits	UINT	È possibile utilizzare questo parametro per verificare i bit di controllo: <ul style="list-style-type: none"> • Bit 0: EN: abilitazione • Bit 1: EW - Attesa abilitazione • Bit 2: ST - Avvio • Bit 3: ER - Errore • Bit 4: DN - Eseguito • Bit 5: CIPCONN - chiusura connessione CIP • Bit 6: EIPSESS - chiusura sessione EIP • Gli altri bit sono riservati I StatusBits sono definiti per i bit di status CIPSTATUS.

Vedere anche

[Codici di errore CIPSTATUS](#) a pagina 187

[Bit di stato CIPSTATUS](#) a pagina 187

[MSG_CIPSYMBOLIC](#) a pagina 191

[MSG_CIPGENERIC](#) a pagina 182

Bit di stato CIPSTATUS

I bit di stato CIPSTATUS si impostano in base allo stato dell'esecuzione del messaggio, dei buffer di comunicazione e delle condizioni del piolo.

-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bit	Nome	Descrizione	Comportamento
0	EN	Enable	Impostato quando il piolo passa a True. Rimane impostato finché non viene impostato il bit DN o ER e il piolo passa a False.
1	EW	Enable Waiting	Impostato quando il buffer di comunicazione è allocato per la richiesta di messaggio. Cancellato quando è impostato il bit ST.
2	ST	Start	Impostato quando il messaggio è stato trasmesso ed è in attesa di una risposta. Cancellato quando è impostato il bit DN.
3	ER	Errore	Impostato quando la trasmissione del messaggio non riesce. Viene scritto un codice di errore in ErrorID. Il bit ER e i valori del codice di errore sono cancellati al successivo passaggio del piolo da False a True.
4	DN	Done	Impostato quando il messaggio viene trasmesso correttamente. Il bit DN è cancellato al successivo passaggio del piolo da False a True. Quando è impostato il bit Done, tutti gli altri bit vengono cancellati, a indicazione del corretto completamento del MSG. Quando viene rilevato un errore ed è impostato il bit Error, gli altri bit di stato (EN/EW/ST) non vengono cancellati.
5	CIPCONN	Done	Impostare quando la connessione CIP per la comunicazione viene terminata. Il bit CIPCONN è applicabile quando ConnClose è True, per gli altri casi il bit CIPCONN è False. Il bit CIPCONN viene usato anche per Seriale, Ethernet e USB.
6	EIPSESS	Done	Impostare quando la sessione incapsulamento CIP per la comunicazione viene terminata. Il bit EIPSESS è applicabile quando ConnClose è True, per gli altri casi il bit EIPSESS è False. Questo bit è utilizzato solo per Ethernet.

Vedere anche

[Tipo di dati CIPSTATUS](#) a pagina 186

Codici di errore CIPSTATUS

Utilizzare questa tabella per determinare i valori dei parametri per i campi ErrorID e SubErrorID del parametro CIPSTATUS quando è impostato il bit ER.

Codice ErrorID	SubErrorID	Descrizione codice di errore
33		Errori relativi alla configurazione del parametro
	32	Numero canale errato.
	36	Tipo di connessione CIP non supportato.
	40	Tipo di dati CIP simbolico non supportato.
	41	Nome CIP simbolico non valido.
	43	Valore classe CIP o conteggio MemberID non valido.
	48	La dimensione dell'array per i dati di ingresso del blocco di istruzioni non è sufficiente.
	49	Percorso di destinazione non valido.
	50	Codice servizio errato.
	51	La dimensione dell'array per i dati di trasmissione del blocco di istruzioni è troppo grande per la comunicazione CIP. la lunghezza massima dei dati utente da trasmettere varia in base alle diverse configurazioni del messaggio. Se il payload totale del messaggio CIP (compresi i dati utente e il sovraccarico del messaggio CIP) supera i 504 byte, viene generato un errore 0x21 (subError 0x33).
	52	Valore del tipo di segmento non valido.
	53	Valore di timeout UCCM errato. Se il valore di timeout incapsulamento è inferiore al timeout UCCM o la differenza tra timeout incapsulamento e timeout UCCM è minore o uguale a un secondo, viene generato un errore 0x21 (subError 0x35).

Codice ErrorID	SubErrorID	Descrizione codice di errore
	54	Valore di timeout connesso non valido. Se il valore di timeout incapsulamento è inferiore a quello del timeout del messaggio CONNESSO o la differenza tra il timeout di incapsulamento e il timeout del messaggio CONNESSO è minore o uguale a un secondo, viene segnalato un errore 0x21 (subError 0x36).
55	Errori relativi al timeout	
	112	Timeout del messaggio durante l'attesa nella rispettiva coda.
	113	Timeout del messaggio durante l'attesa della connessione al livello del collegamento.
	114	Timeout del messaggio durante l'attesa della trasmissione al livello del collegamento.
	115	Timeout del messaggio durante l'attesa della risposta dal livello del collegamento.
69	Codici degli errori relativi al formato di risposta del server	
	65	La risposta al messaggio non corrisponde alla richiesta.
	68	Tipo di dati della risposta al messaggio non valido/supportato. (MSG_CIPSYMBOLIC).
208	Nessun indirizzo IP configurato per la rete.	
209	Numero massimo di connessioni utilizzate. Nessuna connessione disponibile.	
210	Indirizzo Internet o del nodo non valido.	
217	Esecuzione del messaggio annullata dall'utente. (Il parametro Cancel era impostato su TRUE.)	
218	Spazio buffer di rete non disponibile.	
222	Riservato.	
223	L'indirizzo del collegamento non è disponibile. È in corso una modifica alla configurazione TCP/IP o Ethernet.	
224	Codice di errore della risposta CIP. SubErrorID specifica lo stato del CIP, mentre ExtErrorID specifica il valore dello stato esteso del CIP. Fare riferimento alla specifica del CIP per i valori dei codici dei possibili errori.	
255	<p>Il canale è in arresto oppure è in corso una riconfigurazione. Un codice di errore viene generato immediatamente dopo l'accensione, finché non si stabilisce una connessione. Si tratta di un comportamento normale.</p> <p>Potrebbe verificarsi anche nelle seguenti situazioni:</p> <ul style="list-style-type: none"> • Un cavo Ethernet è scollegato • Un indirizzo IP non può essere individuato • È presente un plug-in della porta seriale, ma non configurato 	

Vedere anche

[Tipo di dati CIPSTATUS](#) a pagina 186

Tipo di dati CIPTARGETCFG

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati CIPTARGETCFG.

Parametri	Tipo di dati	Descrizione
Percorso	STRING[80]	Percorso della destinazione. È possibile specificare massimo due hop. La sintassi del percorso è: • {"<porta>,<indirizzo nodo/slot>"}2
CipConnMode	USINT	Tipo di connessione CIP. • 0 - Non connesso (impostazione predefinita) • 1 - Connessione Class3

Parametri	Tipo di dati	Descrizione
UcmmTimeout	UDINT	Timeout messaggio non connesso (in millisecondi). Il tempo di attesa di una risposta per i messaggi non connessi, inclusa l'esecuzione di una connessione per i messaggi non connessi. <ul style="list-style-type: none"> • Valori validi: 250-10.000 ms. • Impostare su 0 per utilizzare il valore predefinito 3.000 (3 secondi). • Per i valori minori di 250 ms, verrà impostato il valore 250 ms (minimo). • Per i valori maggiori di 10.000 ms, verrà impostato il valore 10.000 (massimo).
ConnMsgTimeout	UDINT	Timeout connessione Class3 (in millisecondi). Il tempo di attesa di una risposta per i messaggi connessi. La connessione si interrompe una volta trascorso il tempo di timeout. <ul style="list-style-type: none"> • Valori validi: 800-10.000 ms. • Impostare su 0 per utilizzare il valore predefinito 10.000 (10 secondi). • Per i valori minori di 800, verrà impostato il valore 800 ms (minimo). • Per i valori maggiori di 10.000 ms, verrà impostato il valore 10.000 (massimo).
ConnClose	BOOL	Comportamento alla chiusura della connessione: <ul style="list-style-type: none"> • TRUE - Chiusura della connessione al completamento del messaggio. • FALSE - Nessuna chiusura della connessione al completamento del messaggio (impostazione predefinita).

Vedere anche

[Connessioni per messaggio CIP/EIP](#) a pagina 190

[Timer per timeout messaggio CIP](#) a pagina 191

[MSG_CIPSYMBOLIC](#) a pagina 191

[MSG_CIPGENERIC](#) a pagina 182

[Percorso di destinazione per la messaggistica CIP](#) a pagina 189

Percorso di destinazione per la messaggistica CIP

Il percorso di destinazione per la messaggistica CIP contiene i parametri che determinano il percorso e la destinazione del messaggio CIP.

Il parametro della stringa relativa al percorso di destinazione usa la seguente sintassi:

- "<porta locale>, <indirizzo prima destinazione>, [<porta locale prima destinazione>, <indirizzo seconda destinazione>]"

Il primo hop deve essere presente, il secondo è facoltativo.

Elemento della stringa	Descrizione
Porta locale	Porta locale utilizzata per inviare il messaggio. La porta deve essere una porta seriale CIP o EtherNet/IP attiva. Le porte USB non sono supportate.
Indirizzo prima destinazione	Indirizzo di destinazione del primo hop. <ul style="list-style-type: none"> • Come EIP, specificare l'indirizzo IP di destinazione. L'indirizzo IP deve essere un indirizzo unicast e non un indirizzo con valore 0, multicast, broadcast, locale o loopback (127.x.x.x). • Come CIP seriale, specificare l'indirizzo del nodo di destinazione. Il valore supportato è 1.

Porta locale della prima destinazione	Porta locale utilizzata per inviare il messaggio.
Indirizzo seconda destinazione	Indirizzo di destinazione del secondo hop.

Esempio di percorso di destinazione

Nella seguente tabella sono elencati valori di esempio utilizzati in una stringa di percorso di destinazione, accanto a una descrizione dei risultati relativi a ciascuna stringa.

Elemento della stringa	Descrizione
"0,0"	Il dispositivo di destinazione è il dispositivo locale.
"6,1"	Tramite la porta 6 (porta seriale UPM del Micro830) raggiunto il nodo su 1.
"4,192.168.1.100"	Tramite la porta 4 (porta Ethernet integrata del Micro850) raggiunto il nodo all'indirizzo 192.168.1.100.
"4,192.168.1.100,1,0"	Tramite la porta 4 (porta Ethernet integrata del Micro850) raggiunto il nodo all'indirizzo 192.168.0.100 (modulo Logix ENET). Dal modulo ENET tramite la porta backplane (porta 1) raggiunto il controllore Logix allo slot 0.

Vedere anche

[Tipo di dati CIPTARGETCFG](#) a pagina 188

Connessioni per messaggio CIP/EIP

Per l'esecuzione del messaggio client sono supportate massimo 16 connessioni CIP (classe 3) e 16 connessioni EIP. Nella seguente tabella è illustrato il comportamento della connessione CIP/EIP.

Scenario	Risultati
La richiesta messaggio è abilitata e CipConnMode=1.	Se non esiste una connessione alla destinazione, viene stabilita una connessione CIP. Se esiste già una connessione alla destinazione, viene utilizzata la connessione CIP esistente.
La richiesta messaggio è abilitata, CipConnMode=1 e la porta locale del messaggio è Ethernet.	Se non esiste una connessione EIP alla destinazione, prima di una connessione CIP viene stabilita una connessione EIP.
La richiesta messaggio è abilitata, CipConnMode=0 e la porta locale del messaggio è Ethernet.	Se non esiste una connessione EIP alla destinazione, viene stabilita una connessione EIP.
L'esecuzione del messaggio è completata e ConnClose è impostato su True.	Se è presente solo una connessione alla destinazione, la connessione viene terminata. Se sono presenti più connessioni alla destinazione, la connessione viene terminata al completamento dell'esecuzione dell'ultimo messaggio. Al termine di una connessione CIP, viene terminata anche qualsiasi connessione EIP associata. Se più connessioni CIP usano la stessa connessione EIP, la connessione EIP viene terminata successivamente a tutte le connessioni CIP associate.
Quando ConnClose è true, la connessione di messaggio e sessione EIP viene terminata al completamento dell'esecuzione del messaggio.	Se messaggi condividono la stessa connessione, la connessione viene terminata al completamento dell'ultimo messaggio.
Una connessione CIP o EIP che non è associata a qualsiasi messaggio attivo, viene terminata se è inattiva per x secondi. Dove x è un valore di Timeout dell'inattività di incapsulamento configurabile che può essere impostato mediante il servizio CIP impostato.	Vedere Oggetti CIP specifica Volume II TCP/IP per ulteriori informazioni relative al servizio CIP impostato.
L'esecuzione del messaggio è completata e ConnClose è impostato su False.	La connessione non viene terminata.

Scenario	Risultati
La connessione non è associata a un messaggio attivo e rimane inattiva per l'intervallo di tempo specificato nel parametro ConnTimeOut.	La connessione viene terminata.
Il controllore passa da una modalità di esecuzione (Esecuzione, Esecuzione remota, Scansione singola remota di test e piolo singolo remoto) a una di non esecuzione.	Per tutte le connessioni attive viene forzata la chiusura.

Vedere anche

[Tipo di dati CIPTARGETCFG](#) a pagina 188

Timer per timeout messaggio CIP

Nella seguente tabella è descritto il comportamento dei timer per parametri CIPTARGETCFG timeout (UcmmTimeout e ConnMsgTimeout) in base alle richieste e allo stato del messaggio.

Azione	Risultati
Il messaggio è abilitato.	Il timer UcmmTimeout è attivato
È richiesta la connessione	Il timer ConnMsgTimeout è attivato
Il timer ConnMsgTimeout è attivo	Il timer UcmmTimeout è disabilitato
La richiesta di connessione è completata	Il timer UcmmTimeout è riattivato

Vedere anche

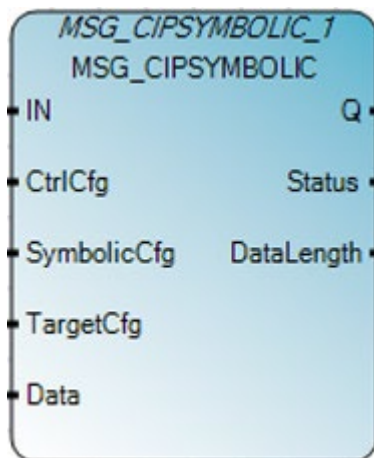
[Tipo di dati CIPTARGETCFG](#) a pagina 188

**MSG_CIPSYMBOLIC
(messaggio simbolico per
Common Industrial Protocol)**

Invia un messaggio CIP (Common Industrial Protocol) simbolico tramite un canale Ethernet o una porta seriale.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



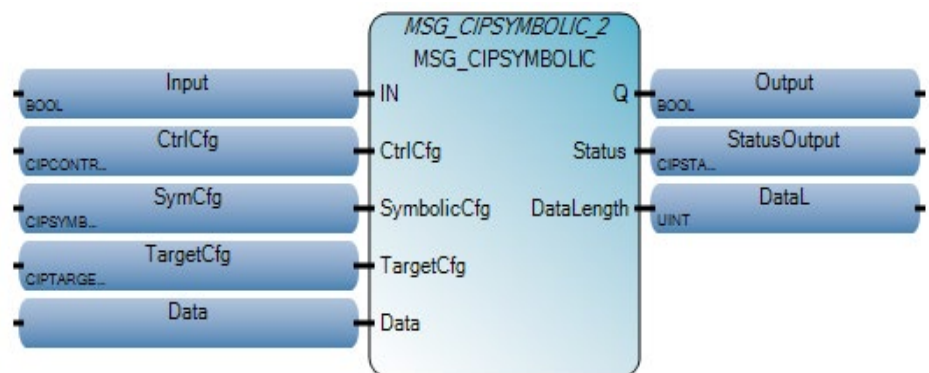
Operazione MSG_CIPSYMBOLIC

Quando è abilitato il blocco funzione, i buffer di ricezione per le operazioni di lettura vengono cancellati sul fronte di salita di Enable.

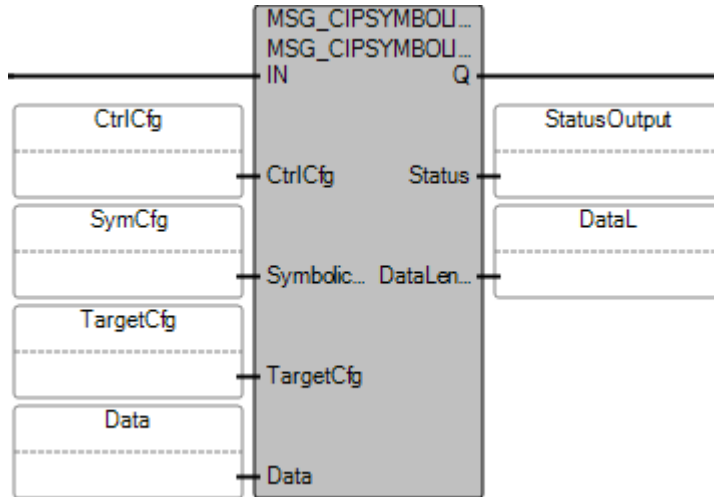
Argomenti

Parametri	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: fronte di salita non rilevato, non attivo.
CtrlCfg	Ingresso	CIPCONTROLCFG	Configurazione per il controllo dell'esecuzione del blocco funzione. Utilizzare i parametri del tipo di dati CIPCONTROLCFG per definire CtrlCfg.
SymbolicCfg	Ingresso	CIPSYMBOLICCFG	Informazioni di lettura e scrittura per il simbolo.
TargetCfg	Ingresso	CIPTARGETCFG	Configurazione del dispositivo di destinazione. Utilizzare i parametri del tipo di dati CIPTARGETCFG per definire TargetCfg.
Dati	Ingresso	USINT[490]	Il comando di lettura memorizza i dati restituiti dal server. Il comando di scrittura inserisce nel buffer i dati da inviare al server. Quando un MSG è attivato o riattivato, Data viene cancellato per il comando Lettura MSG.
Q	Uscita	BOOL	Gli output di questa istruzione sono aggiornati in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: istruzione MSG terminata con successo. FALSE: istruzione MSG non terminata.
Stato	Uscita	CIPSTATUS	Stato esecuzione blocco funzione Quando è attivato o riattivato un MSG, tutti gli elementi in Stato vengono ripristinati. L'uscita dello stato è definita nel tipo di dati CIPSTATUS.
DataLength	Uscita	UDINT	Numero di byte dati per servizio Lettura. Per servizio Scrittura il numero è 0. Quando un MSG è attivato o riattivato, DataLength viene ripristinato su 0 per il comando Lettura MSG.

Esempio di Programmazione a blocchi funzionali MSG_CIPSYMBOLIC



Esempio di Diagramma ladder MSG_CIPSYMBOLIC



Esempio di Testo strutturato MSG_CIPSYMBOLIC

```
MSG_CIPSYMBOLIC_1(int1, Ctrl1, symbol1, target1, data1);
out1 := MSG_CIPGENERIC_1.Q;
status1 := MSG_CIPGENERIC_1.St;
resleng1 := MSG_CIPGENERIC_1.ResLength;

MSG_CIPGENERIC_2(
void MSG_CIPGENERIC_2(BOOL IN, CIPCONTROLCFG CtrlCfg, CIPAPPCFG AppCfg, CIPTARGETCFG TargetCfg, USINT[1..1] ReqData, UINT ReqLength, USINT[1..1] ResData)
Tipo: MSG_CIPGENERIC, Inviare un messaggio CIP esplicito.
```

Vedere anche

- [Tipo di dati CIPCONTROLCFG](#) a pagina 185
- [Tipo di dati CIPTARGETCFG](#) a pagina 188
- [Tipo di dati CIPSYMBOLICCFG](#) a pagina 193
- [Istruzioni di comunicazione](#) a pagina 179
- [Esempio: Come creare un programma di generazione messaggi MSG_CIPSYMBOLIC per scrivere i dati in un controllore](#) a pagina 226

Tipo di dati CIPSYMBOLICCFG Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati CIPSYMBOLICCFG.

Parametri	Tipo di dati	Descrizione
Servizio	USINT	Codice servizio: <ul style="list-style-type: none"> • 0 - Lettura (impostazione predefinita) • 1 - Scrittura
Simbolo	STRING	Nome della variabile per lettura/scrittura. <ul style="list-style-type: none"> • Massimo 80 caratteri. • Il campo non può essere vuoto. Sintassi dei simboli definita nella sintassi di scrittura/lettura dei messaggi simbolici.

Count	UINT	Numero degli elementi della variabile per lettura/scrittura: <ul style="list-style-type: none"> • Valori validi: 1 - 490 • 1 è utilizzato se il valore è impostato su 0.
Tipo	Definito dall'utente	Tipo di dati definito dall'utente per la variabile di destinazione. Tipo definito nel supporto del tipo di dati simbolico.
Offset	USINT	Riservato per uso futuro. Un offset dei byte della variabile Lettura/Scrittura usata per leggere/scrivere una variabile di grandi dimensioni che non è possibile elaborare in un messaggio. <ul style="list-style-type: none"> • 0 - 0xFF
Riservato per uso futuro.		

Supporto per il tipo di dati simbolico

Utilizzare questa tabella per determinare i tipi di dati supportati da MSG_CIPSYMBOLIC.

Tipo di dati	Valore (esadecimale) del tipo di dati	Descrizione
BOOL	193 (0xC1)	Booleano logico con valori TRUE (1) e FALSE (0).
SINT	194 (0xC2)	Valore intero con segno a 8 bit
INT	195 (0xC3)	Valore intero con segno a 16 bit
DINT	196 (0xC4)	Valore intero con segno a 32 bit
LINT	197 (0xC5)	Valore intero con segno a 64 bit
USINT	198 (0xC6)	Valore intero senza segno a 8 bit
UINT	199 (0xC7)	Valore intero senza segno a 16 bit
UDINT	200 (0xC8)	Valore intero senza segno a 32 bit
ULINT	201 (0xC9)	Valore intero senza segno a 64 bit
REAL	202 (0xCA)	Valore virgola mobile a 32 bit
LREAL	203 (0xCB)	Valore virgola mobile a 64 bit
STRING	218 (0xDA)	Stringa di caratteri

Vedere anche

[MSG_CIPSYMBOLIC](#) a pagina 191

[Sintassi scrittura/lettura dei messaggi simbolici](#) a pagina 194

Sintassi scrittura/lettura dei messaggi simbolici

La sintassi definisce le combinazioni di simboli di un blocco istruzione di lettura/scrittura valido.

Nomi validi simbolo

Per essere valido, il nome di ogni simbolo deve soddisfare i requisiti descritti di seguito.

- Deve iniziare con una lettera o un carattere di sottolineatura, seguito da lettere, cifre o caratteri di sottolineatura singoli.
- Deve essere composto da un massimo di 40 caratteri.
- Non può contenere due caratteri di sottolineatura consecutivi.

- Deve contenere i caratteri speciali [] . , , come separatori.

Sintassi dei simboli

Utilizzare questa tabella per definire la sintassi valida per i simboli. Sono supportate solo le variabili globali.

Simbolo	Sintassi	Esempio
Variabile	PROGRAM:<nome programma>,<nome simbolo>	PROGRAM:POU1.MyTag
Array	<nome simbolo>[dim3, dim2, dim1] (la dimensione massima supportata è 3).	MyTag1[0] MyTag2[3,6] MyTag3[1,0,4]
Struttura	<nome simbolo>.<nome simbolo del campo struttura>	MyTag4.time.year MyTag5.local.time[1].year

Vedere anche

[MSG_CIPSYMBOLIC](#) a pagina 191

[Tipo di dati CIPSYMBOLICCFG](#) a pagina 193

Dimensione pacchetto dati supportata per la funzione CIP seriale

Per i controllori Micro830, Micro850 e Micro870, la porta seriale integrata e le porte seriali plug-in supportano la comunicazione CIP seriale. Il pacchetto dati di comunicazione CIP seriale include dati utente e intestazione del pacchetto CIP.

Durante l'uso come client CIP seriale, le porte seriali Micro830/Micro850 possono supportare un massimo di 490 byte di dati utente in lettura/scrittura. Questa specifica massima riguarda i pacchetti dati CIP seriale con una dimensione minima dell'intestazione del pacchetto. Se la dimensione dell'intestazione del pacchetto è maggiore della dimensione minima per l'intestazione del pacchetto, la dimensione massima dei dati utente supportabile dal client CIP è inferiore a 490 byte. Se la dimensione del pacchetto dati è maggiore della dimensione massima dei dati supportata dal client CIP, il blocco funzione segnala un errore (0x21) e un errore secondario (0x33).

Durante l'uso come server CIP seriale, le porte seriali Micro830, Micro850 e Micro870 possono supportare un minimo di 255 byte di dati utente in lettura/scrittura. Questa specifica per la dimensione dei dati utente minima riguarda i pacchetti dati CIP seriale con una dimensione massima dell'intestazione del pacchetto. Se la dimensione dell'intestazione del pacchetto CIP è minore della dimensione massima per l'intestazione del pacchetto, il client CIP può supportare dimensioni del pacchetto dati maggiori della specifica minima (ovvero, maggiori di 255 byte). Tuttavia, se la dimensione dei dati utente è maggiore della dimensione massima dei dati supportata dalla funzione del server CIP, il pacchetto dati CIP può essere scartato e il client va in timeout.

IMPORTANTE Per la funzione server CIP seriale, si consiglia di non effettuare operazioni in lettura/scrittura per più di 255 byte per i dati utente in un singolo messaggio CIP.

Vedere anche

[MSG_CIPSYMBOLIC](#) a pagina 191

MSG_MODBUS (messaggio modbus)

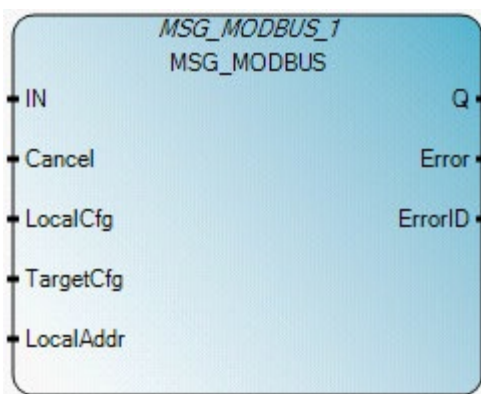
Invia un messaggio Modbus tramite una porta seriale.

Dettagli operazione:

- In una scansione è possibile elaborare un massimo di quattro richieste di messaggio per canale. Per le programmazioni in ladder diagram, le richieste di messaggio sono eseguite al termine di una scansione ladder.
- quando un trigger è impostato su continuo, anche i codici di errore saranno cancellati in modo continuo. Per visualizzare i codici di errore, aggiungere un piolo prima dell'istruzione MSG_MODBUS.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: fronte di salita non rilevato, non avviato.
Annulla	Ingresso	BOOL	TRUE: annulla l'esecuzione del blocco funzione. FALSE: quando IN è TRUE. Input Annulla dominante.
LocalCfg	Ingresso	MODBUSLOCPARA	Definisce l'ingresso struttura (dispositivo locale). Definire la struttura d'ingresso per il dispositivo locale utilizzando il tipo di dati MODBUSLOCPARA.
TargetCfg	Ingresso	MODBUSTARPARA	Definisce l'ingresso struttura (dispositivo locale). Definire la struttura d'ingresso per il dispositivo di destinazione utilizzando il tipo di dati MODBUSTARPARA.
LocalAddr	Ingresso	MODBUSLOCADDR	MODBUSLOCADDR è un array di 125 parole usato dai comandi di lettura per archiviare i dati (da 1 a 125 parole) restituiti dal Modbus slave e dai comandi di scrittura per inserire i dati nel buffer (da 1 a 125 parole) da inviare al Modbus slave.

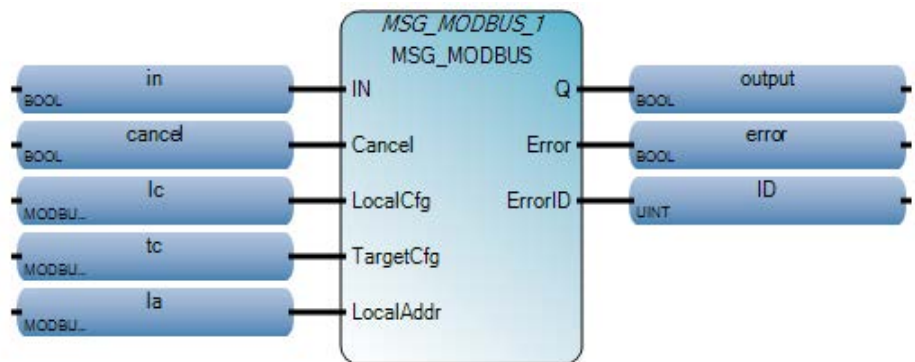
Q	Uscita	BOOL	Gli output di questa istruzione sono aggiornati in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: istruzione MSG terminata con successo. FALSE: istruzione MSG non terminata.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore MSG_MODBUS.

Codici di errore MSG_MODBUS

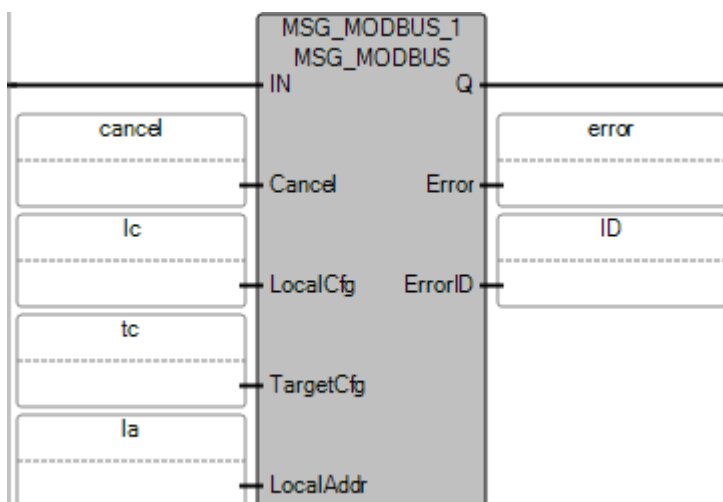
La tabella seguente descrive i codici di errore per MSG_MODBUS.

Codice errore	Descrizione errore
3	Il valore di TriggerType è stato modificato da 2 - 255.
20	Il driver di comunicazione locale non è compatibile con l'istruzione MSG.
21	Si è verificato un errore del parametro di configurazione canale locale.
22	L'indirizzo del ponte locale o di destinazione è maggiore dell'indirizzo massimo di nodo.
33	Parametro del file MSG non valido.
54	Modem non trovato.
55	Timeout del messaggio nel processore locale. Timeout del livello di collegamento.
217	L'utente ha cancellato il messaggio.
129	Funzione non valida.
130	Indirizzo dati non valido.
131	Valore dati non valido.
132	Guasto nel dispositivo slave.
133	Conferma.
134	Il dispositivo slave è occupato.
135	Conferma negativa.
136	Errore di parità in memoria.
137	Risposta non standard.
255	Il canale è stato spento.

Esempio di Programmazione a blocchi funzionali MSG_MODBUS



Esempio di Diagramma ladder MSG_MODBUS



Esempio di Testo strutturato MSG_MODBUS

```
MSG_MODBUS2_1 (
void MSG_MODBUS2_1(BOOL IN, BOOL Cancel, MODBUS2LOCPARA LocalCfg, MODBUS2TARPARA TargetCfg, MODBUSLOCADDR LocalAddr)
Tipo: MSG_MODBUS2, Inviare un messaggio Modbus.

1 MSG_MODBUS2_1(in, cancel, lc, tc, la);
2 output := MSG_MODBUS2_1.Q;
3 error := MSG_MODBUS2_1.Error;
4 ID := MSG_MODBUS2_1.ErrorID;
```

Vedere anche

- [Istruzioni di comunicazione](#) a pagina 179
- [Esempio: Come configurare una comunicazione Modbus per la lettura da e la scrittura in una unità](#) a pagina 236
- [Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208
- [Tipo di dati MODBUSTARPARA](#) a pagina 202
- [Tipo di dati MODBUSLOCPARA](#) a pagina 198

Tipo di dati MODBUSLOCPARA Utilizzare questa tabella per determinare i valori dei parametri per il tipo di dati MODBUSLOCPARA.

Parametri	Tipo di dati	Descrizione
Canale	UINT	Numero porta seriale PLC Micro800: <ul style="list-style-type: none"> • 2 per la porta seriale integrata o • 5-9 per i plug-in della porta seriale installati negli slot da 1 a • 5 per lo slot 1 • 6 per lo slot 2 • 7 per lo slot 3 • 8 per lo slot 4 • 9 per lo slot 5
TriggerType	USINT	Rappresenta uno dei seguenti: <ul style="list-style-type: none"> • 0: Msg attivato una volta (quando IN passa da False a True) • 1: Msg attivato di continuo quando IN è True • Altro valore: Riservato
Cmd	USINT	Rappresenta uno dei seguenti: <ul style="list-style-type: none"> • 01: lettura stato della bobina (0xxxx) • 02: lettura stato ingresso (1xxxx) • 03: lettura registri di mantenimento (4xxxx) • 04: lettura registri ingresso (3xxxx) • 05: scrittura bobina singola (0xxxx) • 06: scrittura registro singolo (4xxxx) • 15: scrittura bobine multiple (0xxxx) • 16: scrittura registri multipli (4xxxx) • Altri: Supporto del comando personalizzato. <p>Supporto del comando personalizzato MODBUSLOCPARA: Sono supportati anche i comandi personalizzati nell'intervallo 0-255 e non ancora assegnati al comando Modbus. Se viene usato un comando personalizzato, LocalCfg:ElementCnt conterrà il numero dei byte ricevuti. La risposta sarà ricevuta in Local Address Data e sovrascriverà i dati richiesti.</p> <ul style="list-style-type: none"> • Esempio per CMD=0x2B • Local Address Data 1:0x0E, READ_DEVICE_ID_MEI • Local Address Data 2:0x01, READ_DEV_ID_BASIC • Local Address Data 3:0x00, Read Vendor Object
ElementCnt	UINT	Limiti <ul style="list-style-type: none"> • Per ingressi Lettura bobina/Discreto: 2000 bit • Per Lettura registro: 125 parole • Per Scrittura bobina: 1968 bit • Per Scrittura registro: 123 parole

Attivazione messaggio MSG_MODBUS

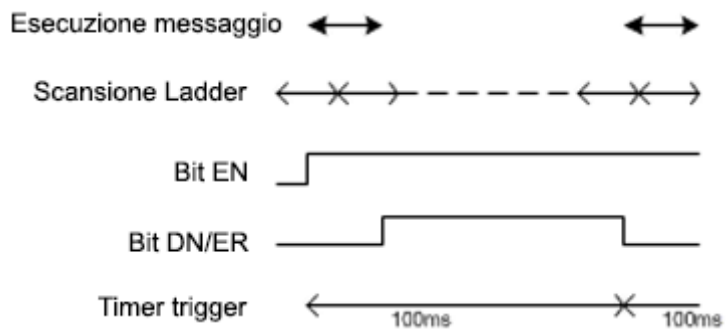
Un messaggio Modbus può essere attivato periodicamente impostando il parametro TriggerType su un valore diverso da zero.

La tabella seguente descrive il funzionamento del parametro tipo di trigger quando utilizzato con il blocco funzione MSG_MODBUS.

Azione	Risultati
Il messaggio è abilitato.	Il timer attivazione si attiva.
Il tempo impostato nel timer attivazione è esaurito prima del completamento del messaggio.	Il messaggio è attivato immediatamente nel ciclo di scansione ladder successivo.
Il messaggio viene completato prima che si esaurisca il tempo impostato nel timer attivazione.	Il messaggio è attivato quando il tempo impostato nel timer attivazione è esaurito.

Esempio: attivazione messaggio

Nell'esempio seguente il valore TriggerType è impostato su 100.

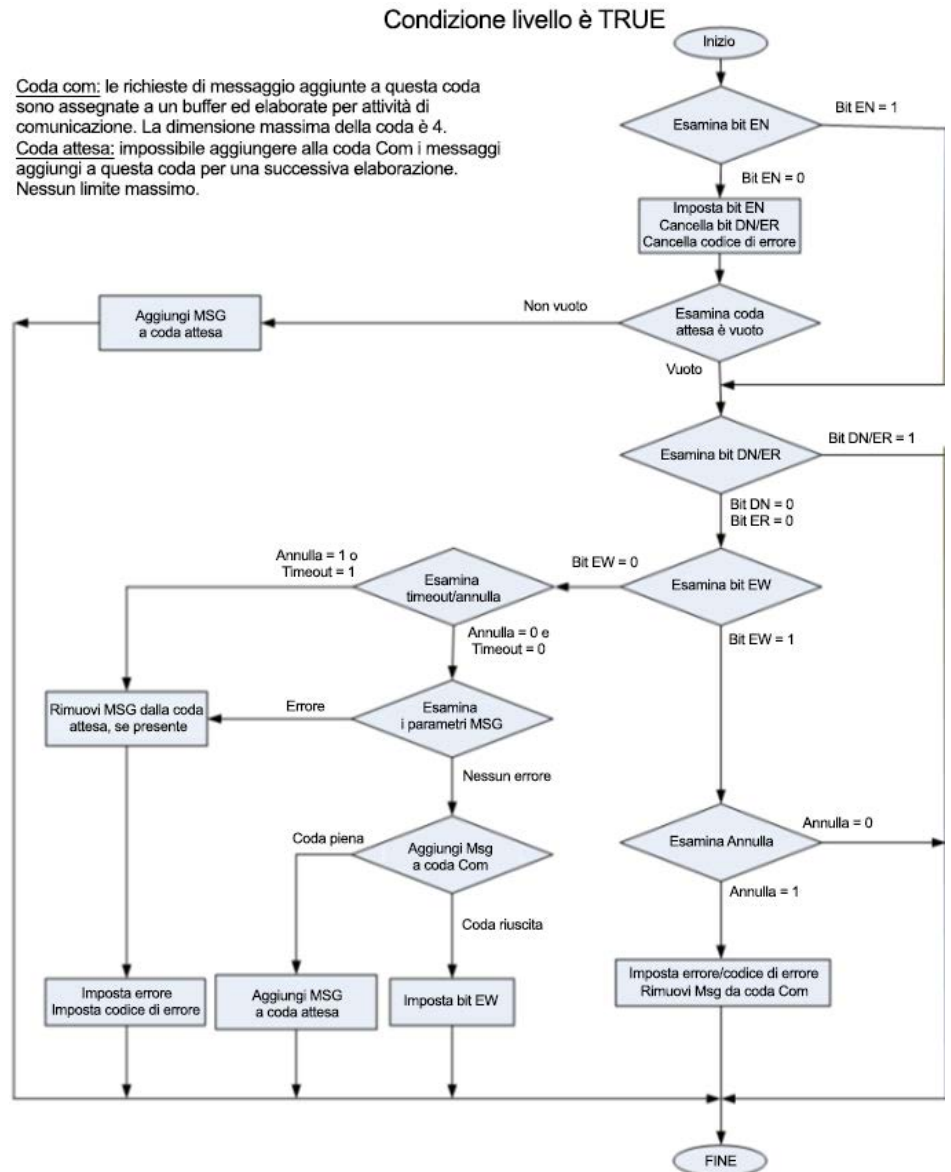


Vedere anche

[MSG MODBUS](#) a pagina 196

Processo di esecuzione del messaggio (Rung = TRUE)

Nel seguente diagramma dei processi sono descritti gli eventi relativi alle istruzioni dei messaggi che si verificano quando la condizione Rung è True.



Coda Com: le richieste di messaggio aggiunte alla coda Com hanno un buffer allocato ed elaborato dall'attività di comunicazione. Il limite massimo per la coda è 4.

Coda Attesa: i messaggi che non è possibile aggiungere alla coda Com sono aggiunti alla coda Attesa per la successiva elaborazione. Per la coda Attesa non è previsto un limite di dimensione massimo.

Vedere anche

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

[Istruzioni di comunicazione](#) a pagina 179

[MSG MODBUS](#) a pagina 196

Tipo di dati MODBUSTARPARA

Nella tabella seguente è descritto il tipo di dati MODBUSTARPARA.

Parametri	Tipo di dati	Descrizione
Addr	UDINT	Indirizzo dati destinazione (1 - 65536). Diminuisce di uno durante l'invio.
Node	USINT	Il valore predefinito per l'indirizzo del nodo slave è 1. L'intervallo è 1 - 247. Zero è l'indirizzo di trasmissione Modbus ed è valido solo per comandi di scrittura Modbus (ad esempio, 5, 6, 15 e 16).

Vedere anche

[MSG_MODBUS](#) a pagina 196

MSG_MODBUS2 (messaggio MODBUS/TCP)

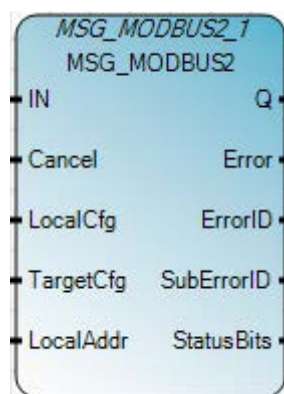
Invia un messaggio MODBUS/TCP su un canale Ethernet.

Dettagli operazione:

- In una scansione è possibile elaborare un massimo di quattro richieste di messaggio per canale. Per le programmazioni in ladder diagram, le richieste di messaggio sono eseguite al termine di una scansione ladder.
- Quando MSG_MODBUS2 è abilitato, i buffer di ricezione per le operazioni di lettura vengono cancellati sul fronte di salita di Enable.
- Annullando l'esecuzione dell'istruzione MSG_MODBUS2 non è garantito che la richiesta di messaggio in uscita venga cancellata, ma la risposta non sarà elaborata.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Stato ingresso ramo. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: fronte di salita non rilevato, non attivo.

Annulla	Ingresso	BOOL	TRUE: annulla l'esecuzione del blocco funzione. Annullando l'esecuzione dell'istruzione MSG_MODBUS2 non è garantito che la richiesta di messaggio in uscita venga cancellata, ma la risposta non sarà elaborata. FALSE: quando IN è TRUE. Input Annulla dominante.
LocalCfg	Ingresso	MODBUS2LOCPARA	Definisce l'ingresso della struttura (dispositivo locale). Definire la struttura d'ingresso per il dispositivo locale utilizzando il tipo di dati MODBUS2LOCPARA.
TargetCfg	Ingresso	MODBUS2TARPARA	Definisce l'ingresso della struttura (dispositivo locale). Definire la struttura d'ingresso per il dispositivo di destinazione utilizzando il tipo di dati MODBUS2TARPARA.
LocalAddr	Ingresso	MODBUSLOCADDR	Tipo di dati MODBUSLOCADDR è un vettore a 125 parole. Utilizzo di LocalAddr: <ul style="list-style-type: none"> • Per i comandi di lettura, archiviare i dati (1-125 parole) restituiti dallo slave Modbus. • Per i comandi di scrittura, inserire nel buffer i dati (1-125 parole) da inviare allo slave Modbus.
Q	Uscita	BOOL	Gli output di questa istruzione sono aggiornati in modo asincrono rispetto alla scansione del programma. L'output Q non è utilizzabile per riattivare l'istruzione, in quanto IN viene attivato sul fronte. TRUE: istruzione MSG terminata con successo. FALSE: istruzione MSG non terminata.
Errore	Uscita	BOOL	Indica un errore rilevato. TRUE: è stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti nei codici di errore Modbus2.
SuberrorID	Uscita	UINT	Si utilizza per verificare i bit di stato: <ul style="list-style-type: none"> • Bit 0: EN: abilitazione • Bit 1: EW - Attesa abilitazione • Bit 2: ST - Avvio • Bit 3: ER - Errore • Bit 4: DN - Eseguito Gli altri bit sono riservati.
StatusBits	Uscita	UINT	Valore del codice di errore SubError quando Error è TRUE. Quando un MSG è attivato o riattivato, un SubErrorID impostato in precedenza viene cancellato.

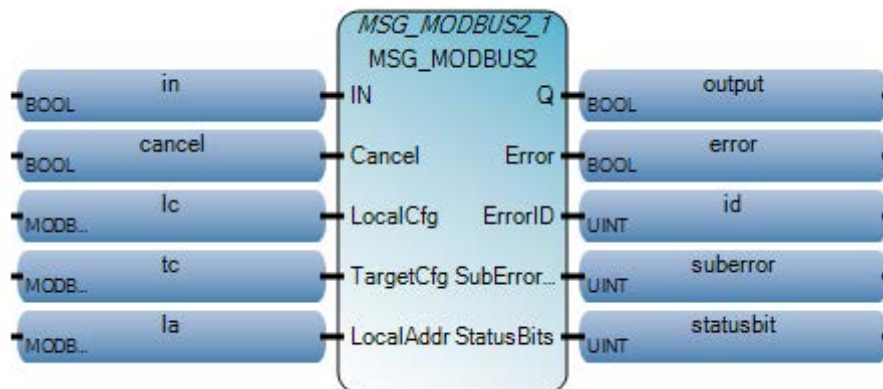
Codici di errore ed errore secondario MSG_MODBUS2

Quando è impostato il bit ER, i campi ErrorID e SubErrorID visualizzano i seguenti codici di errore.

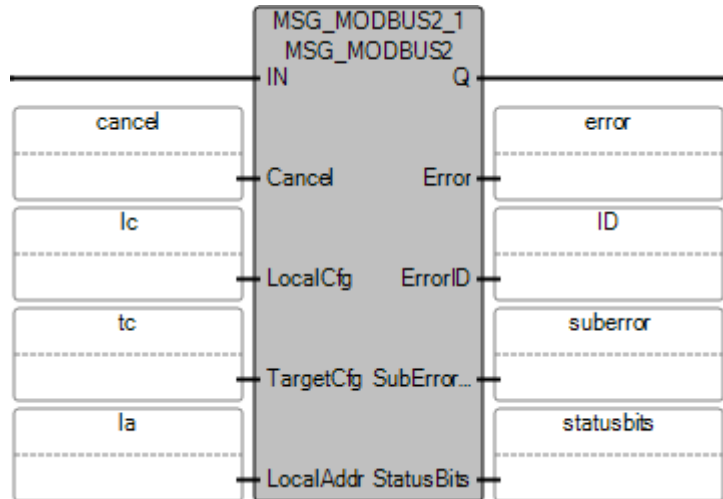
ID errore	SubErrorID	Descrizione
33	Errori relativi alla configurazione del parametro	
	32	Numero canale errato.
	37	Numero di elementi errato.
	38	Indirizzo dati errato.
55	Errori relativi al timeout	
	112	Timeout del messaggio durante l'attesa nella rispettiva coda.
	113	Timeout del messaggio durante l'attesa di una connessione al livello del collegamento.
	114	Timeout del messaggio durante l'attesa della trasmissione al livello del collegamento.

	115	Timeout del messaggio durante l'attesa della risposta dal livello del collegamento.
69	Codici degli errori relativi al formato di risposta del server	
208	Nessun indirizzo IP configurato per la rete.	
209	Numero massimo di connessioni utilizzate. Nessuna connessione disponibile.	
210	Indirizzo Internet o del nodo non valido.	
217	Esecuzione del messaggio annullata dall'utente. (Il parametro Cancel era impostato su TRUE.)	
222	Impossibile stabilire una connessione di rete prima del timeout.	
255	Il canale è in arresto oppure è in corso una riconfigurazione. Un codice di errore viene generato immediatamente dopo l'accensione, finché non si stabilisce una connessione. Si tratta di un comportamento normale. Può essere generato anche se un cavo Ethernet viene scollegato o se non è possibile rilevare un indirizzo IP.	
	Codici di errore della risposta slave	
129	Codice di funzione non valido	
130	Indirizzo dati non valido	
131	Valore dati non valido	
132	Errore del server	
133	Conferma	
134	Conferma negativa	
136	Errore di parità in memoria	
137	Codice di errore della risposta non standard. Il codice di errore effettivo può trovarsi nel SubErrorID.	

Esempio di programmazione a blocchi funzionali MSG_MODBUS2



Esempio di Diagramma ladder MSG_MODBUS2



Esempio di Testo strutturato MSG_MODBUS2

```
MSG_MODBUS2_1(
void MSG_MODBUS2_1(BOOL IN, BOOL Cancel, MODBUS2LOCPARA LocalCfg, MODBUS2TARPARA TargetCfg, MODBUSLOCADDR LocalAddr)
Tipo: MSG_MODBUS2, Inviare un messaggio Modbus.
```

```
1 MSG_MODBUS2_1(in, cancel, lc, tc, la);
2 output := MSG_MODBUS2_1.Q;
3 error := MSG_MODBUS2_1.Error;
4 ID := MSG_MODBUS2_1.ErrorID;
```

Vedere anche

[Istruzioni di comunicazione](#) a pagina 179

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

[Tipo di dati MODBUS2LOCPARA](#) a pagina 205

[Tipo di dati MODBUS2TARPARA](#) a pagina 206

Tipo di dati MODBUS2LOCPARA

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati MODBUS2LOCPARA.

Parametri	Tipo di dati	Descrizione
Canale	UINT	Numero della porta Ethernet locale: <ul style="list-style-type: none"> • 4 per la porta Ethernet integrata di Micro850 e Micro820.
TriggerType	UDINT	Tipo di attivazione messaggio: <ul style="list-style-type: none"> • 0: Msg attivato una volta (quando IN passa da False a True) • 1 - 65535: valore di attivazione ciclico in millisecondi. Messaggio attivato periodicamente, quando IN è True e l'esecuzione del messaggio precedente è stata completata. • Impostare il valore su 1 per attivare i messaggi il più rapidamente possibile. Vedere Attivazione messaggio MSG_MODBUS2.

Cmd	USINT	<p>Comando Modbus:</p> <ul style="list-style-type: none"> • 01: lettura stato della bobina (0xxxx) • 02: lettura stato ingresso (1xxxx) • 03: lettura registri di mantenimento (4xxxx) • 04: lettura registri ingresso (3xxxx) • 05: scrittura bobina singola (0xxxx) • 06: scrittura registro singolo (4xxxx) • 15: scrittura bobine multiple (0xxxx) • 16: scrittura registri multipli (4xxxx) • Altri: Supporto del comando personalizzato <p>Supporto del comando personalizzato MODBUS2LOCPARA: Sono supportati anche i comandi personalizzati nell'intervallo 0-255 e non ancora assegnati a un comando Modbus. Se viene usato un comando personalizzato, LocalCfg:ElementCnt conterrà il numero di byte ricevuti. La risposta sarà ricevuta in Local Address Data e sovrascriverà i dati richiesti. Esempio per CMD=0x2B:</p> <ul style="list-style-type: none"> • Local Address Data 1:0x0E, READ_DEVICE_ID_MEI • Local Address Data 2:0x01, READ_DEV_ID_BASIC • Local Address Data 3:0x00, Read Vendor Object
ElementCnt	UINT	<p>Limiti</p> <ul style="list-style-type: none"> • Per ingressi Lettura bobina/Discreto: 2000 bit • Per Lettura registro: 125 parole • Per Scrittura bobina: 1968 bit • Per Scrittura registro: 123 parole

Attivazione messaggio MSG_MODBUS2

Un messaggio Modbus può essere attivato periodicamente impostando il parametro TriggerType su un valore diverso da zero.

Nella tabella seguente viene descritto cosa accade se il parametro TriggerType viene utilizzato con il blocco funzione MSG_MODBUS2.

Azione	Risultati
Il messaggio è abilitato.	Il timer attivazione si attiva.
Il tempo impostato nel timer di attivazione scade prima del completamento del messaggio.	Il messaggio è attivato immediatamente nel ciclo di scansione ladder successivo.
Il messaggio viene completato prima che il tempo impostato nel timer di attivazione scada.	Il messaggio è attivato quando il tempo impostato nel timer attivazione è esaurito.

Vedere anche

[MSG_MODBUS2](#) a pagina 202

Tipo di dati MODBUS2TARPARA

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati MODBUS2TARPARA.

Parametri	Tipo di dati	Descrizione
Addr	UDINT	<p>Indirizzo dati Modbus del dispositivo di destinazione:</p> <ul style="list-style-type: none"> • 1 - 65536. • Diminuisce di uno durante l'invio. • Il firmware usa low-word dell'indirizzo, se il valore dell'indirizzo è superiore a 65536.

Parametri	Tipo di dati	Descrizione
NodeAddress[4]	USINT	Indirizzo IP del dispositivo di destinazione. L'indirizzo IP deve essere un indirizzo unicast valido e non 0, multicast, broadcast, locale o loopback (127.x.x.x). Ad esempio, per specificare 192.168.2.100: <ul style="list-style-type: none"> • NodeAddress[0]=192 • NodeAddress[1]=168 • NodeAddress[2]=2 • NodeAddress[3]=100
Porta	UINT	Numero della porta TCP di destinazione. La porta Modbus/TCP standard è 502. 1 - 65535 Impostare a 0 per utilizzare il valore predefinito 502
UnitId	USINT	Identificatore unità. Utilizzato per comunicare con i dispositivi slave tramite bridge Modbus. Fare riferimento alle specifiche Modbus per maggiori dettagli. Tenere presente che Micro800 non deve tentare di convalidare questo valore. 0 - 255 Impostare su 255 se il dispositivo di destinazione non è un bridge.
MsgTimeOut	UDINT	Timeout messaggio (in millisecondi). Il tempo di attesa di una risposta a un comando avviato. <ul style="list-style-type: none"> • 250-10.000 • Impostare su 0 per utilizzare il valore predefinito 3.000. • Un valore inferiore a 250 (valore minimo) viene impostato su 250. • Un valore superiore a 10.000 (valore massimo) viene impostato su 10.000. Vedere Timer per timeout messaggio Modbus/TCP.
ConnTimeOut	UDINT	Timeout connessione TCP (in millisecondi). Tempo di attesa per la corretta esecuzione della connessione TCP al dispositivo di destinazione. <ul style="list-style-type: none"> • 250-10.000 • Impostare su 0 per utilizzare il valore predefinito 5.000. • Un valore inferiore a 250 (valore minimo) viene impostato su 250. • Un valore superiore a 10.000 (valore massimo) viene impostato su 10.000. Vedere Timer per timeout messaggio Modbus/TCP.
ConnClose	BOOL	Comportamento alla chiusura della connessione TCP. <ul style="list-style-type: none"> • True: chiude la connessione TCP al completamento del messaggio. • False: non chiude la connessione TCP al completamento del messaggio [impostazione predefinita]. Vedere Connessioni per messaggio Modbus/TCP.

Timer per timeout messaggio Modbus/TCP

La tabella seguente descrive il comportamento per MsgTimeOut e ConnTimeOut in base alle richieste e allo stato del messaggio.

Azione	Risultati
Il messaggio è abilitato.	Attiva il timer MsgTimeOut.
È richiesta la connessione TCP.	Attiva il timer ConnTimeOut.
Il timer ConnMsgTimeout è attivo.	Disabilita il timer MsgTimeOut.
La richiesta di connessione è completata.	Riattiva il timer MsgTimeOut.

Connessioni per messaggio Modbus/TCP

Il client Modbus/TCP supporta un massimo di 16 connessioni. Nella seguente tabella è descritto il comportamento della connessione Modbus/TCP.

Scenario	Risultati
La richiesta di messaggio è abilitata e non esiste alcuna connessione con la destinazione.	In mancanza di una connessione con la destinazione, viene stabilita una nuova connessione. Se esiste già una connessione con la destinazione, viene utilizzata la connessione esistente.
L'esecuzione del messaggio è completata e ConnClose è impostato su True.	Se è presente solo una connessione alla destinazione, la connessione viene terminata. Se sono presenti più connessioni alla destinazione, la connessione viene terminata al completamento dell'esecuzione dell'ultimo messaggio.
L'esecuzione del messaggio è completata e ConnClose è impostato su False.	La connessione non viene terminata.
La connessione non è associata a un messaggio attivo e rimane inattiva per l'intervallo di tempo specificato nel parametro ConnTimeOut.	La connessione viene terminata.
Il controllore passa da una modalità di esecuzione (Esecuzione, Esecuzione remota, Scansione singola remota di test e piolo singolo remoto) a una di non esecuzione.	Per tutte le connessioni attive viene forzata la chiusura.

Vedere anche

[MSG_MODBUS2](#) a pagina 202

Processi di esecuzione di messaggi e diagrammi di temporizzazione

Nei seguenti argomenti sono descritte la modalità e la tempistica di esecuzione delle istruzioni dei messaggi MSG_CIPGENERIC, MSG_CIPSYMBOLIC e MSG_MODBUS2 in base alle relative condizioni di rung e bit.

- Processo di esecuzione del messaggio (generale)
- Processo di esecuzione del messaggio (Rung = TRUE)
- Diagramma di temporizzazione dell'esecuzione del messaggio (Rung = True)
- Processo di esecuzione del messaggio (Rung = FALSE)
- Diagramma di temporizzazione dell'esecuzione del messaggio (Rung = FALSE)
- Processo di esecuzione di messaggi (errore)
- Diagramma di temporizzazione dell'esecuzione del messaggio (errore)

Vedere anche

[Istruzioni di comunicazione](#) a pagina 179

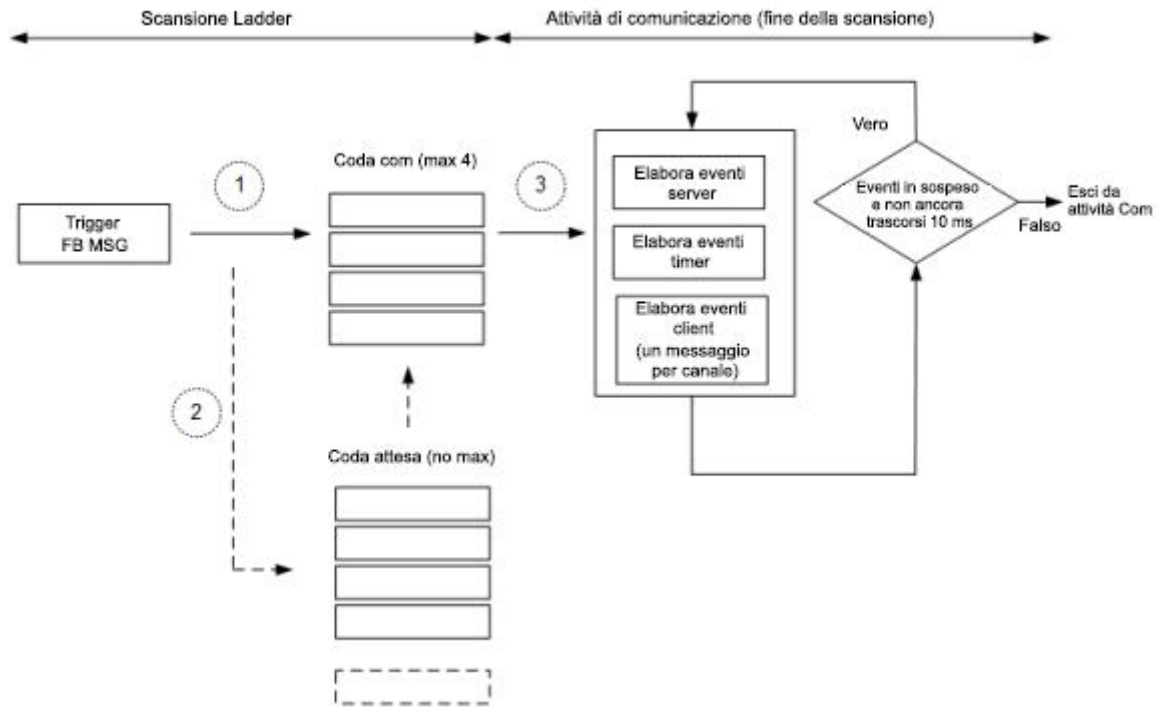
[MSG_MODBUS2](#) a pagina 202

[MSG_CIPSYMBOLIC](#) a pagina 191

[MSG_CIPGENERIC](#) a pagina 182

Processo di esecuzione del messaggio (generale)

Nel seguente diagramma sono illustrate la modalità e la tempistica dell'esecuzione dei messaggi in base allo stato della coda Com.



Nella seguente tabella è descritta la sequenza di eventi identificata nel diagramma precedente.

N.	Descrizione degli eventi
1	Il messaggio è abilitato. Se la coda Com è vuota, il buffer è allocato per il messaggio e il messaggio viene aggiunto alla coda Com per la trasmissione. La dimensione della coda Com è 4. Ogni canale ha una coda distinta.
2	Se la coda Com è piena, il messaggio viene aggiunto alla coda Attesa. Non appena la coda Com si svuota, il messaggio viene aggiunto alla coda Com dalla coda Attesa. Non esiste un limite di dimensione per la coda Attesa. Ogni canale ha una coda distinta.
3	L'attività di comunicazione esegue i messaggi presenti nella coda Com a ogni Fine scansione per la trasmissione. Le code di ogni canale vengono elaborate una per volta con metodo round robin. Viene eseguito un messaggio per ciascun canale. Il processo continua fino all'esecuzione di tutti i messaggi o fino alla scadenza della comunicazione pianificata (10 ms). Come da pianificazione, nella successiva Fine scansione viene visualizzato per primo il canale successivo all'ultimo elaborato.

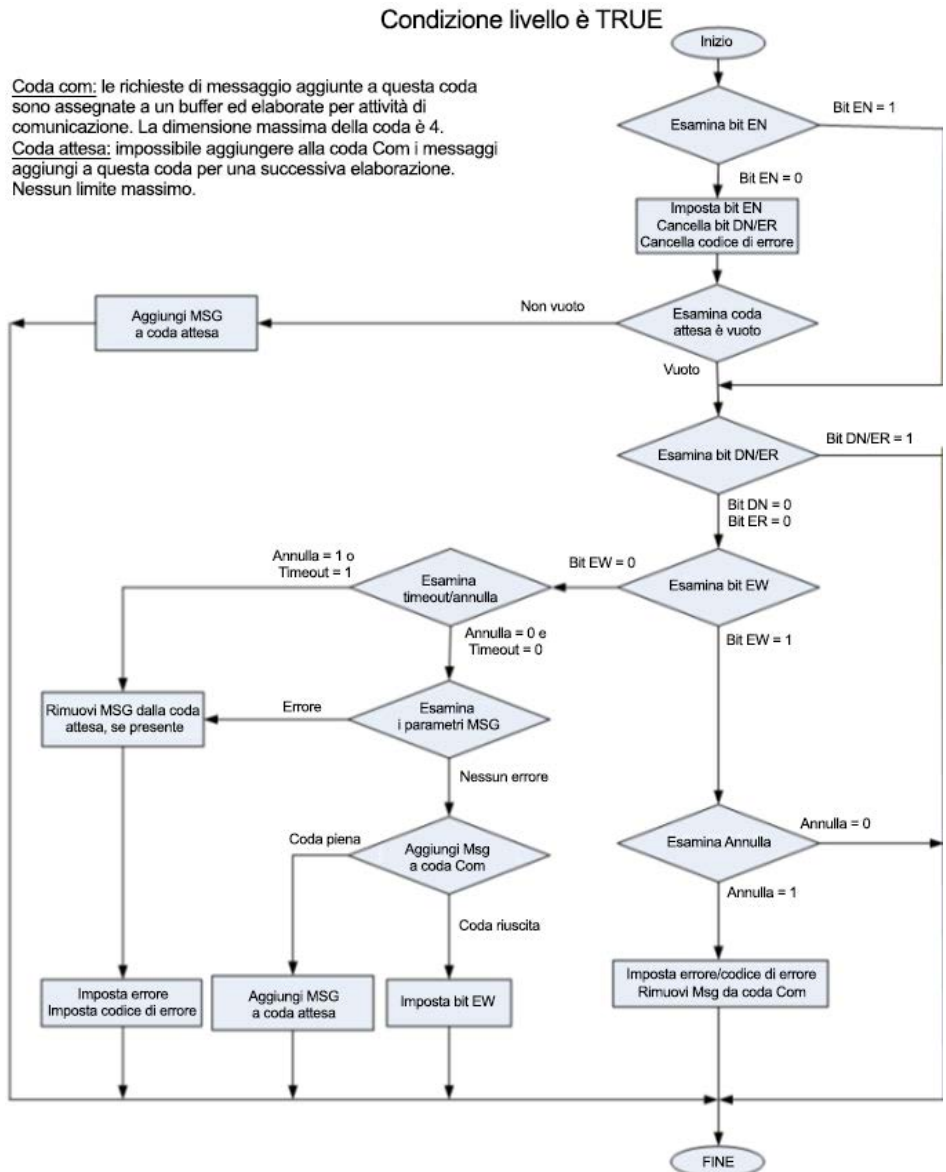
Vedere anche

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

[Istruzioni di comunicazione](#) a pagina 179

Processo di esecuzione del messaggio (Rung = TRUE)

Nel seguente diagramma dei processi sono descritti gli eventi relativi alle istruzioni dei messaggi che si verificano quando la condizione Rung è True.



Coda Com: le richieste di messaggio aggiunte alla coda Com hanno un buffer allocato ed elaborato dall'attività di comunicazione. Il limite massimo per la coda è 4.

Coda Attesa: i messaggi che non è possibile aggiungere alla coda Com sono aggiunti alla coda Attesa per la successiva elaborazione. Per la coda Attesa non è previsto un limite di dimensione massimo.

Vedere anche

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

[Istruzioni di comunicazione](#) a pagina 179

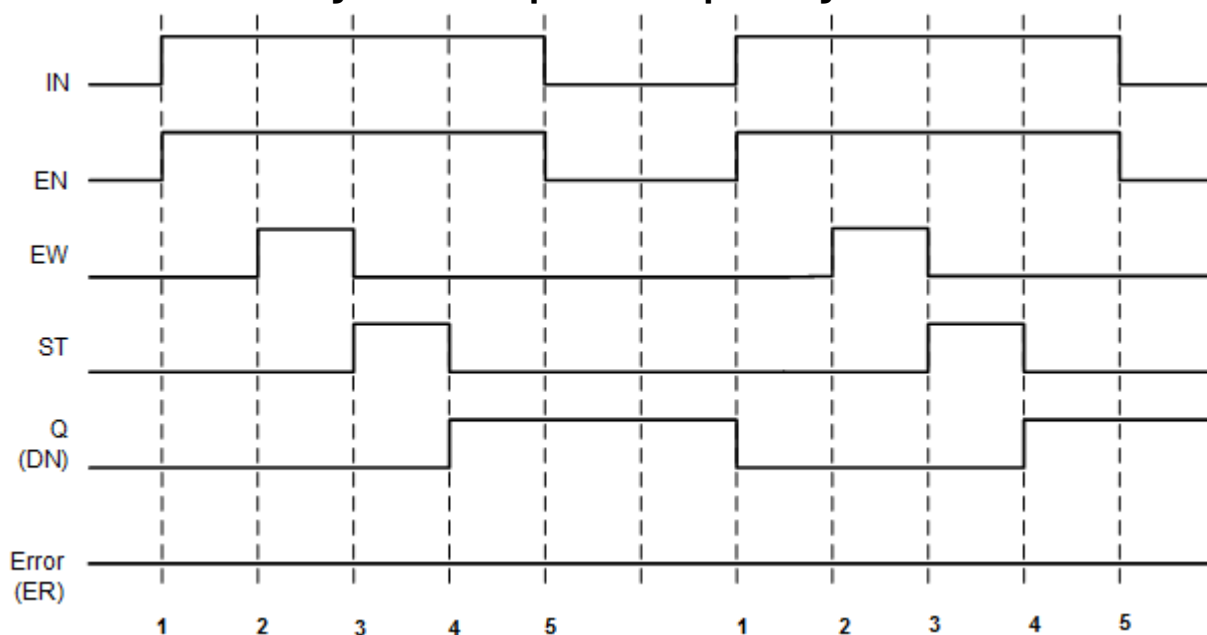
[MSG MODBUS](#) a pagina 196

Diagramma di temporizzazione dell'esecuzione del messaggio (Rung = TRUE)

Nella seguente tabella sono descritte le condizioni dei messaggi e lo stato dei bit per ogni passo dell'esecuzione identificato nel diagramma di temporizzazione, finché la condizione Rung rimane True.

Passo	Descrizione messaggio	Stato dei bit
1	La condizione Rung passa a TRUE. L'esecuzione del messaggio è abilitata.	Il bit EN è impostato. Tutti gli altri bit sono cancellati.
2	Il buffer di controllo dei messaggi viene acquisito. A questo punto i dati di ingresso (vale a dire, il parametro "data" per i messaggi di scrittura) vengono copiati per la trasmissione. Le modifiche successive ai dati di ingresso non verranno applicate al messaggio trasmesso.	Il bit EW è impostato.
3	Inizia la trasmissione del messaggio.	Il bit EW è cancellato. Il bit ST è impostato.
4	Viene ricevuta la risposta al messaggio.	Il bit ST è cancellato. Il bit DN è impostato.
5	La condizione Rung passa a FALSE.	Il bit EN è cancellato.

Diagramma di temporizzazione per (Rung = TRUE)



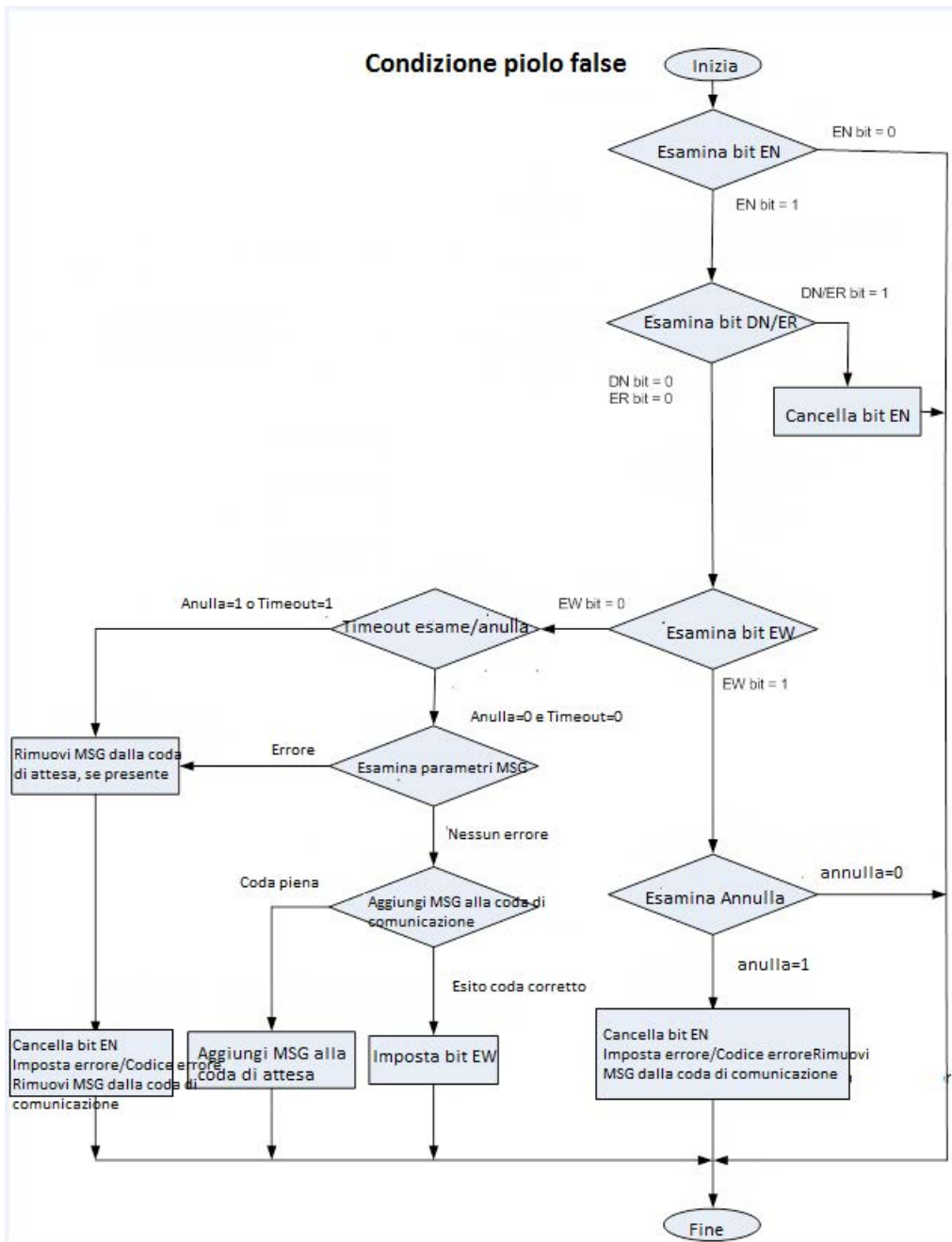
Vedere anche

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

[Istruzioni di comunicazione](#) a pagina 179

Processo di esecuzione del messaggio (Rung = FALSE)

Nel seguente diagramma dei processi sono descritti gli eventi relativi alle istruzioni dei messaggi che si verificano quando la condizione Rung è False.



Vedere anche

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

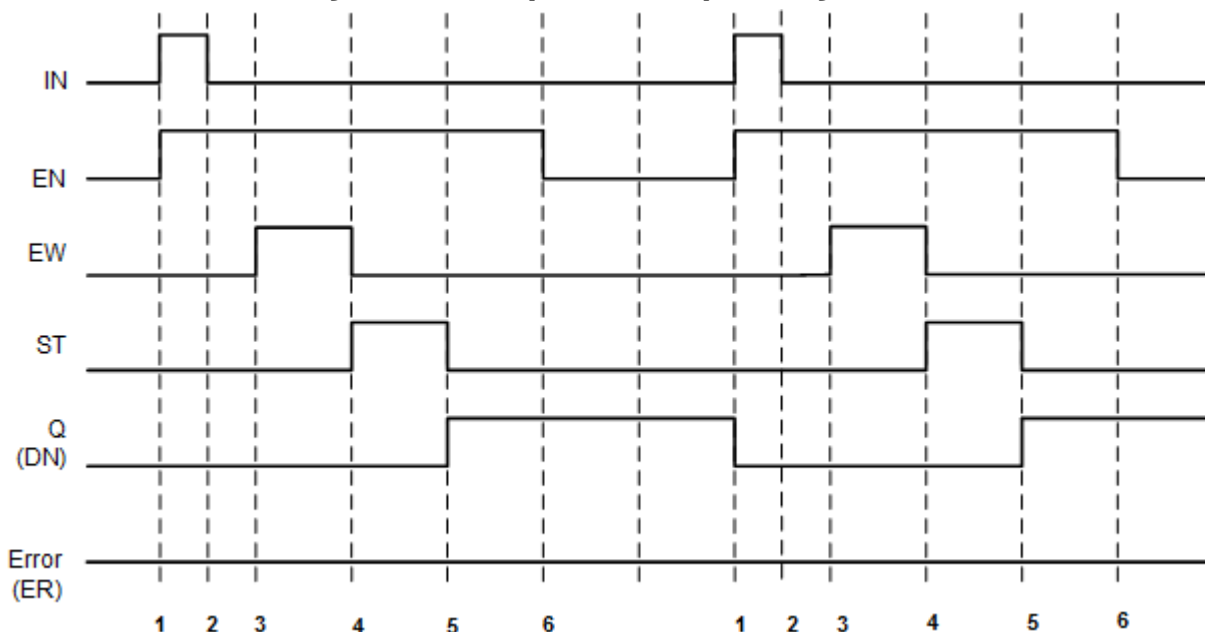
[Istruzioni di comunicazione](#) a pagina 179

Diagramma di temporizzazione dell'esecuzione del messaggio (Rung = FALSE)

Nella seguente tabella sono descritte le condizioni dei messaggi e lo stato dei bit per ogni passo dell'esecuzione identificato nel diagramma di temporizzazione, finché la condizione Rung rimane FALSE durante l'esecuzione.

Passo	Descrizione messaggio	Stato dei bit
1	La condizione Rung passa a TRUE. L'esecuzione del messaggio è abilitata.	Il bit EN è impostato. Tutti gli altri bit sono cancellati.
2	La condizione Rung passa a FALSE. L'esecuzione del messaggio continua.	
3	Viene acquisito il buffer del messaggio. A questo punto i dati di ingresso (vale a dire, il parametro "data" per i messaggi di scrittura) vengono copiati per la trasmissione. Le modifiche successive ai dati di ingresso non verranno applicate al messaggio trasmesso.	Il bit EW è impostato.
4	Inizia la trasmissione del messaggio.	Il bit EW è cancellato. Il bit ST è impostato.
5	Viene ricevuta la risposta al messaggio.	Il bit ST è cancellato. Il bit DN è impostato.
6	Il messaggio viene scansionato di nuovo dopo il passo 5.	Il bit EN è cancellato.

Diagramma di temporizzazione per (Rung = FALSE)



Vedere anche

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

[Istruzioni di comunicazione](#) a pagina 179

Processo di esecuzione di messaggi (errore)

Nella seguente tabella sono descritte le condizioni dei messaggi e lo stato dei bit per ogni passo dell'esecuzione identificato nel diagramma di temporizzazione, quando si verifica un errore durante l'esecuzione.

Passo	Descrizione messaggio	Stato dei bit
1	La condizione Rung passa a TRUE. L'esecuzione del messaggio è abilitata.	Il bit EN è impostato. Tutti gli altri bit sono cancellati.
2	Il buffer del messaggio viene acquisito.	Il bit EW è impostato.

Passo	Descrizione messaggio	Stato dei bit
3	Inizia la trasmissione del messaggio.	Il bit EW è cancellato. Il bit ST è impostato.
4	La trasmissione del messaggio entra in timeout.	I bit EW e ST non cambiano.
4-6	Con la condizione Rung passata a FALSE.	Il bit EN è cancellato. Il bit ER è impostato.

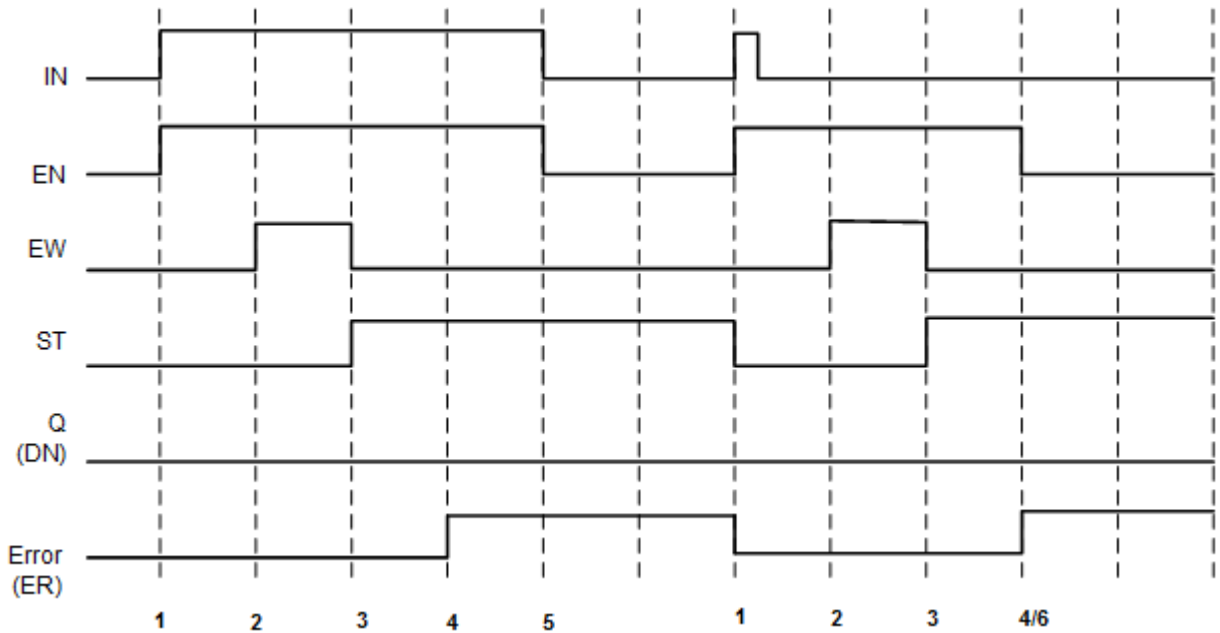
Vedere anche

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

[Istruzioni di comunicazione](#) a pagina 179

Diagramma di temporizzazione dell'esecuzione del messaggio (errore)

Il seguente diagramma di temporizzazione mostra un modello tipico di errore che si verifica durante l'esecuzione.



Vedere anche

[Processi di esecuzione di messaggi e diagrammi di temporizzazione](#) a pagina 208

[Istruzioni di comunicazione](#) a pagina 179

Utilizzo dei blocchi funzione di comunicazione (messaggio)

Questa sezione fornisce dettagli ed esempi specifici per l'utilizzo delle istruzioni di comunicazione nei programmi logici. Per esempi e dettagli sull'utilizzo dei blocchi funzione MSG_CIPGENERIC e MSG_CIPSYMBOLIC al fine di creare programmi, vedere i seguenti argomenti.

Vedere anche

[Istruzioni di comunicazione](#) a pagina 179

[Configurazione dei valori dei dati oggetto per i messaggi espliciti](#) a pagina 215

[Esempio: Come creare un programma di generazione messaggi MSG_CIPGENERIC per leggere i dati da un controllore](#) a pagina 217

[Esempio: Come creare un programma di generazione messaggi MSG_CIPSYMBOLIC per scrivere i dati in un controllore](#) a pagina 226

[Esempio: Come configurare una comunicazione Modbus per la lettura da e la scrittura in una unità](#) a pagina 236

Configurazione dei valori dei dati oggetto per i messaggi espliciti (MSG_CIPGENERIC)

Per utilizzare il blocco funzione MSG_CIPGENERIC per la messaggistica esplicita, sarà necessario configurare il parametro AppCfg con i valori corretti.

Per maggiori informazioni sulla comunicazione tramite messaggi

Esistono diverse risorse informative sull'uso e sull'implementazione delle comunicazioni via messaggistica, inclusa la guida di Connected Components Workbench, i manuali utente e la Rockwell Automation Literature Library.

Nella seguente tabella sono elencate ulteriori fonti di informazioni in merito alla comunicazione tramite messaggi.

Fonte di informazioni	Descrizione	Come trovare le informazioni
Manuale dell'utente per uno specifico dispositivo di comunicazione	Contiene informazioni importanti sulla messaggistica e informazioni specifiche per la configurazione dei blocchi funzione per i messaggi.	Menu Guida Connected Components Workbench
Adattatore EtherNet/IP 22-COMM-E FRN 1.xxx, appendice C	Fornisce informazioni sugli oggetti EtherNet/IP accessibili utilizzando i messaggi espliciti.	Menu Guida Connected Components Workbench
Specifica EtherNet/IP	Definisce gli oggetti da includere in ogni dispositivo CIP: Oggetto Identity, oggetto Message Router e oggetto Network.	Sito Web ODVA (http://www.odva.org)
Controllori programmabili Micro800: Guida introduttiva alla messaggistica client CIP	Fornisce istruzioni rapide per l'utilizzo della messaggistica CIP generica e simbolica nei controllori a logica programmabile (PLC) Micro830 e Micro850.	Rockwell Automation Literature Library

Per accedere ai manuali dell'utente e alle guide rapide:

1. Per accedere alle rapide guide, nel menu **Guida**, fare clic su **Visualizza Guida**.
2. Fare doppio clic su **Connected Components Workbench**.
3. Fare doppio clic su **Guida introduttiva a Connected Components Workbench**.
4. Per accedere ai manuali dell'unità, nel menu **Guida**, fare clic su **Manuali dell'utente** per visualizzare la finestra di dialogo dei manuali.
5. Fare clic sul segno più (+) accanto a Unità per espandere la categoria, quindi espandere la classe fino a individuare il manuale desiderato.
6. Fare doppio clic sul nome del manuale per aprire il file PDF.
7. Per accedere al manuale di EtherNet/IP, nel menu **Guida**, fare clic su **manuali dell'utente** per visualizzare la finestra di dialogo dei manuali.
8. Fare clic sul segno più (+) accanto a **Unità** per espandere la categoria, quindi espandere la classe Periferiche PowerFlex classe 4.
9. Fare doppio clic sul manuale dell'utente dell'adattatore EtherNet/IP 22-COMM-E per aprire il file PDF.

Per accedere ai manuali da Rockwell Automation Literature Library:

1. Visitare <http://literature.rockwellautomation.com>.
2. Per accedere alle versioni in lingua diversa dall'inglese dei manuali dell'utente, selezionare la lingua dalla casella a discesa Lingua della pubblicazione (nell'angolo destro).
3. Nella casella **Cerca**, inserire il numero completo o parziale del catalogo del dispositivo. Ad esempio, inserire 2080-LC30 per visualizzare i manuali utente Micro830.
4. Nella casella **Ricerca**, digitare il numero di catalogo completo o parziale del dispositivo. Ad esempio, inserire 2080-LC30 per visualizzare i manuali utente Micro830.

Dati dell'oggetto Registro CIP

I blocchi funzione MSG_CIPGENERIC utilizzano i dati dell'oggetto Registro CIP nel parametro AppCfg. I dati dell'oggetto includono quanto segue:

- Codice classe
- Istanza
- Attributo istanza
- Servizio

Valori per il parametro MSG_CIPGENERIC AppCfg

Per configurare i parametri del blocco funzione MSG_CIPGENERIC, utilizzare i valori dell'oggetto registro CIP per le variabili di ingresso.

L'immagine seguente mostra i valori dei dati dell'oggetto registro CIP utilizzati nei parametri del blocco funzione MSG_CIPGENERIC.

Nome	Alias	Tipo di dati	Dimensione	Valore predefinito	Valore iniziale
+ MSG_ReadDrive		MSG_CIPGEN	
- MyAppCfg		CIPAPPCFG	
MyAppCfg.Service		USINT			14
MyAppCfg.Class		UINT			7
MyAppCfg.Instance		UDINT			4
MyAppCfg.Attribute		UINT			4
MyAppCfg.MemberCnt		USINT			

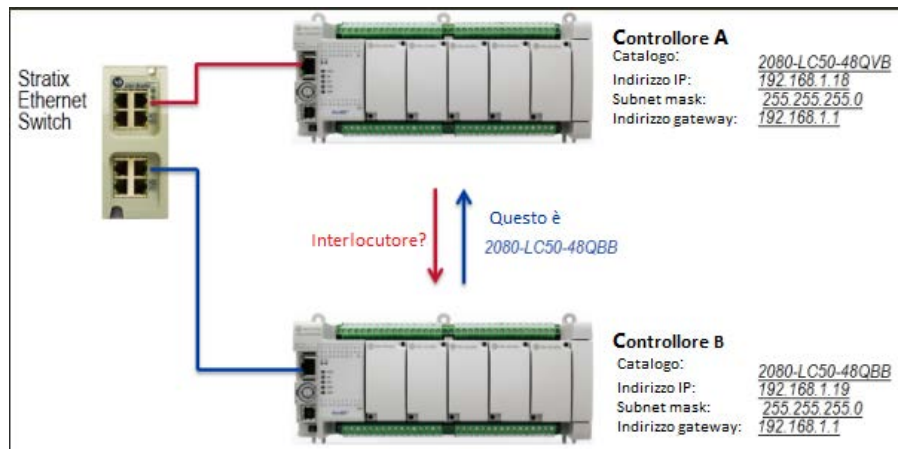
Vedere anche

[MSG_CIPGENERIC](#) a pagina 182

[Utilizzo dei blocchi funzione di comunicazione \(messaggistica\)](#) a pagina 215

Esempio: Come creare un programma di generazione messaggi MSG_CIPGENERIC per leggere i dati da un controllore

Nell'esempio è illustrata la modalità di creazione di un programma di generazione messaggi per recuperare le informazioni di catalogo del controllore B dal controllore A, utilizzando un blocco funzione MSG_CIPGENERIC e un blocco funzione COP.



Per creare un programma di messaggistica MSG_CIPGENERIC utilizzabile per leggere le informazioni da un controllore diverso, eseguire le attività seguenti.

No	Attività
1	Identificazione dei valori iniziali per le variabili di ingresso (MSG_CIPGENERIC) a pagina 218
2	Aggiunta di variabili e di un blocco funzione MSG_CIPGENERIC a pagina 219
3	Configurazione dei valori iniziali per le variabili a pagina 220
4	Aggiunta di un contatto e di una bobina a pagina 222
5	Aggiunta di un blocco funzione COP, di variabili e di un contatto (MSG_CIPGENERIC) a pagina 223
6	Verifica della corretta configurazione IP sul controllore B a pagina 225

Vedere anche

[Configurazione dei valori dei dati oggetto per i messaggi espliciti](#) a pagina 215

[Utilizzo dei blocchi funzione di comunicazione \(messaggistica\)](#) a pagina 215

Identificazione dei valori iniziali per le variabili di input, MSG_CIPGENERIC

Attenersi alla seguente procedura per aggiungere variabili di ingresso e valori iniziali e ottenere i valori di oggetto identità per configurare i valori iniziali del parametro AppCfg.

Per aggiungere variabili di ingresso e valori iniziali:

1. Nel menu **Guida**, fare clic su **Manuali utente**.
2. Espandere Unità e individuare il manuale dell'utente relativo al tipo di adattatore di comunicazione in uso (Manuale dell'utente dell'adattatore EtherNet/IP).
3. Fare doppio clic sul manuale per aprirlo.
4. Consultare le intestazioni dell'Appendice per individuare la sezione contenente le informazioni sugli oggetti EtherNet/IP accessibili mediante i messaggi espliciti (Appendice C).
5. Consultare la sezione Appendice e individuare il tipo di oggetto relativo al messaggio esplicito (oggetto identità).
6. Identificare i valori iniziali dei parametri AppCfg in base al tipo di informazioni da recuperare.

Esempio di dati dell'oggetto Ethernet/IP e dei parametri AppCfg

Nella seguente tabella sono identificati i dati specifici dell'oggetto Ethernet/IP utilizzati per leggere le informazioni di catalogo da un controllore.

Esempio di variabile di ingresso	Parametro AppCfg	Opzione dati dell'oggetto Ethernet/IP	Descrizione	Valore iniziale
MyAppCfg.Service	Servizio	Codice servizio	Implementazione per classe = sì Implementazione per istanza = sì Ottenimento singolo attributo	14 (0x0E in esadecimale)
MyAppCfg.Class	Classe	Codice classe	Classe oggetto EtherNet/IP = oggetto identità	01
MyAppCfg.Instance	Istanza	Istanze	22-COMM-E	01
MyAppCfg.Attribute	Attributo	Attributo istanza	Ottenimento nome prodotto e classificazione come SHORT STRING	07

Vedere anche

[Aggiunta di variabili e di un blocco funzione per messaggi generici MSG_CIPGENERIC](#) a pagina 219

[Esempio: Come configurare i messaggi espliciti \(MSG_CIPGENERIC\)](#) a pagina 217

Aggiunta di variabili e di un blocco funzione MSG_CIPGENERIC

Per aggiungere un blocco funzione MSG_CIPGENERIC per una Programmazione in Ladder Diagram, quindi aggiungere variabili di ingresso al blocco funzione, attenersi alla seguente procedura.

Per aggiungere un blocco funzione MSG_CIPGENERIC:

1. Aggiungere un controllore:
 - Espandere la cartella **Controllori** e la cartella Micro850 per visualizzare tutti i controllori Micro850.
 - Fare doppio clic su un controllore (2080-LC50-24QVB) per aggiungerlo all'**Organizzatore progetto**.
2. Aggiungere un programma con ladder diagram:
 - Nell'**Organizzatore progetto**, fare clic con il pulsante destro del mouse su **Programmi**, scegliere **Aggiungi**, quindi fare clic su **Nuovo LD: Ladder diagram**.
 - Fare clic con il pulsante destro del mouse sull'icona del diagramma ladder nell'**Organizzatore progetto**, selezionare **Rinomina** e digitare CIPexplicitMessage.
 - Fare doppio clic su Programmazione in Ladder Diagram nell'**Organizzatore progetto** per visualizzare la POU di LD nell'editor di lingue.
3. Aggiungere il blocco di funzione MSG_CIPGENERIC:
 - Nella **Casella degli strumenti**, selezionare il **blocco istruzione**, quindi trascinarlo e rilasciarlo nel ramo del ladder per visualizzare il **Selettore blocco istruzione**.
 - Nel campo di ricerca digitare **MSG** per visualizzare i blocchi di funzione per i messaggi.
 - Digitare **MSG_ReadDrive** nel campo **Istanza**.
 - Fare doppio clic su **MSG_CIPGENERIC** per aggiungere un'istanza del blocco di funzione al ladder diagram.
4. Aggiungere variabili di ingressi locali MSG_CIPGENERIC:
 - Nell'**Organizzatore progetto**, fare doppio clic su **Variabili locali** per visualizzare la pagina **Variabili locali**.
 - Nella pagina **Variabili** aggiungere le variabili e i tipi di dati elencati nella tabella.

Parametri	Nome variabile	Tipo di dati
CtrlCfg	MyCtrlCfg	CIPCONTROLCFG
AppCfg	MyAppCfg	CIPAPPCFG
TargetCfg	MyTargetCfg	CIPTARGETCFG
ReqData	MyReqData	USINT
ReqLength	MyReqLength	UINT
ResData	MyResData	USINT (array)

5. Per la variabile MyResData fare doppio clic su **Dimensione** e modificare la dimensione dell'array su [1..81].

L'aspetto della pagina **Variabili** deve essere simile all'immagine raffigurata di seguito.

	Nome	Alias	Tipo di dati	Dimensione	Valore progetto	Valore iniziale
	MSG_ReadDrive		MSG_CIPGEN	
	MyCtrlCfg		CIPCONTROL	
	MyAppCfg		CIPAPPCFG	
	MyTargetCfg		CIPTARGETCF	
	MyReqData		UDINT			
	MyReqLength		UDINT			
	MyResData		UDINT	[1..81]

Vedere anche

[Configurazione dei valori iniziali \(MSG_CIPGENERIC\)](#) a pagina 220

[Esempio: Come configurare i messaggi espliciti \(MSG_CIPGENERIC\)](#) a pagina 217

[Identificazione dei valori iniziali per le variabili di ingresso \(MSG_CIPGENERIC\)](#) a pagina 218

Configurazione dei valori iniziali per le variabili

Per aggiungere valori iniziali alle variabili di ingresso già create in precedenza e assegnarle al parametro di ingresso corretto del blocco funzione MSG_CIPGENERIC, attenersi alla seguente procedura.

Per assegnare variabili a MSG_CIPGENERIC:

- Per configurare i valori iniziali della variabile di ingresso MyCtrlCfg:
 - Nella pagina **Variabili locali**, espandere MyCtrlCfg per visualizzare i relativi parametri.
 - Immettere i seguenti valori nella colonna **Valore iniziale** di ciascun parametro.

Parametri	Valore iniziale	Commenti
MyCtrlCfg.Cancel	Lasciare vuoto	Non necessario.
MyCtrlCfg.TriggerType	0	È necessario recuperare il numero di catalogo una sola volta.
MyCtrlCfg.StrMode	Lasciare vuoto	Non necessario.

- Per configurare i valori iniziali della variabile di ingresso MyAppCfg:
 - Nella pagina **Variabili locali**, espandere MyAppCfg per visualizzare i relativi parametri.
 - Immettere i seguenti valori nella colonna **Valore iniziale** di ciascun parametro.

Parametri	Valore iniziale
MyAppCfg.Service	14
MyAppCfg.Class	01
MyAppCfg.Instance	01

MyAppCfg.Attribute 07

3. Per configurare i valori iniziali della variabile di ingresso MyTargetCfg

- Nella pagina **Variabili locali**, espandere MyTargetCfg per visualizzare i relativi parametri.
- Immettere i seguenti valori nella colonna **Valore iniziale** di ciascun parametro.

Parametri	Valore iniziale	Commenti
MyTargetCfg.Path	"4,192.168.100.4"	Il primo "4" indica che il messaggio è inviato tramite la porta Ethernet integrata. 192.168.100.4 è l'indirizzo IP dell'interfaccia Ethernet del dispositivo.
MyTargetCfg.CipConnMode	0	Per i messaggi CIP è preferibile impostarlo su Non connesso.
MyTargetCfg.UcmmTimeout	vuoto	I messaggi non connessi prevedono un timeout con impostazione predefinita di 3000 millisecondi, se i relativi Valori iniziali sono vuoti.
MyTargetCfg.ConnMsgTimeout	vuoto	I messaggi connessi prevedono un timeout con impostazione predefinita di 3000 millisecondi, se i relativi Valori iniziali sono vuoti.
MyTargetCfg.ConnClose	FALSE	Per la messaggistica connessa la connessione CIP può essere chiusa immediatamente dopo il completamento dell'istruzione di messaggio, impostando il Valore iniziale su TRUE.

I parametri nella pagina **Variabili** devono avere l'aspetto raffigurato nella seguente immagine.

Nome	Alias	Tipo di dati	Dimensione	Valore progett	Valore iniziale
MSG_ReadDrive		MODBUSLOC	
MyCtrlCfg		CIPCONTROL	
MyCtrlCfg.Cancel		BOOL			
MyCtrlCfg.TriggerType		UDINT			
MyCtrlCfg.StrMode		USINT			
MyAppCfg		CIPAPPCFG	
MyAppCfg.Service		USINT			14
MyAppCfg.Class		UINT			01
MyAppCfg.Instance		UDINT			01
MyAppCfg.Attribute		UINT			07
MyAppCfg.MemberCnt		USINT			
MyAppCfg.MemberId		CIPMEMBERID	
MyTargetCfg		CIPTARGETCF	
MyTargetCfg.Path		STRING			'4,192.168.100.4'
MyTargetCfg.CipConnMode		USINT			0
MyTargetCfg.UcmmTimeout		UDINT			
MyTargetCfg.ConnMsgTimeout		UDINT			
MyTargetCfg.ConnClose		BOOL			FALSE
MyReqData		USINT			
MyReqLength		UDINT			
MyResData		UDINT	[1..81]
MyResData[1]		UDINT			

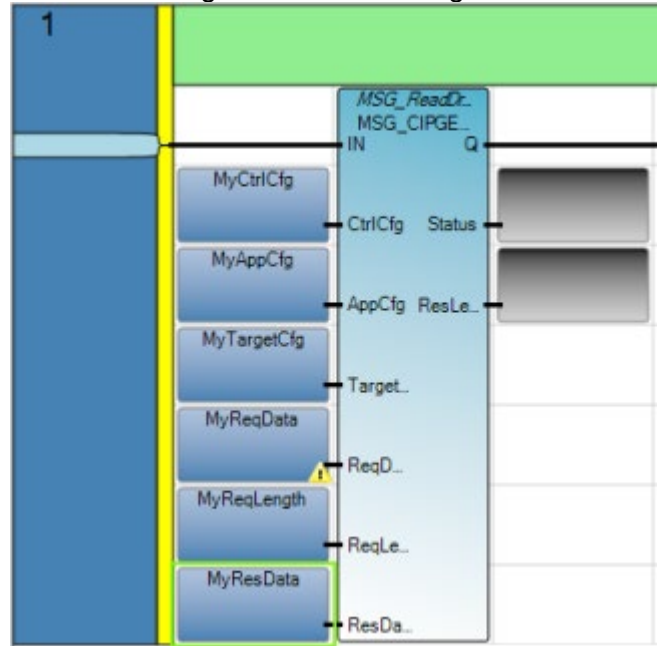
4. Per assegnare variabili ai parametri

- Nella POU del ladder diagram fare clic in cima al blocco di ingresso della variabile per aprire l'elenco a discesa con le variabili.
- Dall'elenco, assegnare ogni parametro di ingresso alla relativa variabile di ingresso corretta, come indicato nella seguente tabella.

Parametri	Variabile di ingresso	Commenti
CtrlCfg	MyCtrlCfg	Il numero di catalogo deve essere recuperato solo una volta, quindi il valore iniziale di MyCtrlCfg.TriggerType è impostato su 0.
AppCfg	MyAppCfg	I valori iniziali sono stati determinati cercando i valori dei dati oggetto di Servizio, Classe, Istanza e Attributo.
Destinazione	MyTargetCfg	I valori iniziali servono per la configurazione del dispositivo di destinazione.
ReqData	MyReqData	Essendo un messaggio di lettura, non vi è richiesta di dati, pertanto i parametri ReqData non sono utilizzati.
ReqLength	MyReqLength	Essendo un messaggio di lettura, non vi è richiesta di dati, pertanto i parametri ReqLength non sono utilizzati.

ResData	MyResData	<p>La stringa con il numero di catalogo è memorizzata nell'array con formato stringa breve ODVA.</p> <p>Il primo elemento dell'array definisce la lunghezza della stringa, gli altri rappresentano il valore esadecimale del carattere della stringa.</p> <p>Il numero massimo di caratteri è 80 più la lunghezza dell'elemento, quindi MyResData è definito come un array a 1 dimensione con 81 elementi.</p>
---------	-----------	--

L'istanza del blocco funzione MSG_CIPGENERIC deve essere simile all'immagine mostrata di seguito.



Vedere anche

[Aggiunta di variabili e di un blocco funzione per messaggi generici MSG_CIPGENERIC](#) a pagina 219

[Aggiunta di un contatto e di una bobina](#) a pagina 222

[Esempio: Come configurare i messaggi espliciti \(MSG_CIPGENERIC\)](#) a pagina 217

Aggiunta di un contatto e di una bobina

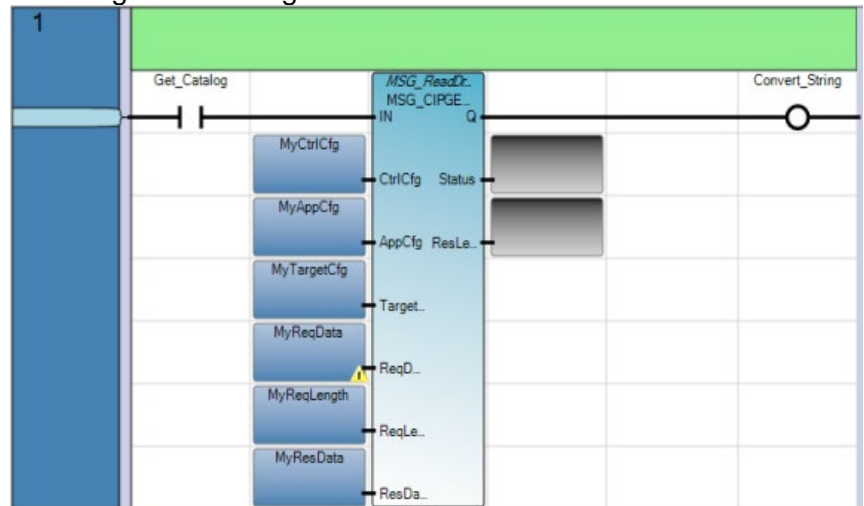
Attenersi ai passaggi seguenti per aggiungere una bobina e un contatto all'istruzione MSG_CIPGENERIC che converte le informazioni del catalogo in una stringa leggibile dall'utente.

Per aggiungere una bobina a MSG_CIPGENERIC:

1. Nella **casella degli strumenti**, selezionare **Contatto diretto**, quindi trascinarlo e rilasciarlo a sinistra dell'ingresso con il blocco funzione MSG_CIPGENERIC sul primo piolo del ladder.
2. Nel **selettore di variabili**, digitare **Get_Catalog** nel campo Nome del contatto.

3. Nella **casella degli strumenti**, selezionare **Bobina diretta**, quindi trascinarla e rilasciarla a destra dell'uscita con il blocco di funzione MSG_CIPGENERIC sul primo piolo del ladder.
4. Nel **selettore di variabili**, digitare **Convert_String** nel campo Nome della bobina.

Il primo piolo del programma con ladder diagram per la messaggistica MSG_CIPGENERIC dovrebbe apparire come nella seguente immagine.



Vedere anche

[Aggiunta di un blocco funzione COP, di variabili e contatti](#) a pagina 223

[Configurazione dei valori iniziali \(MSG_CIPGENERIC\)](#) a pagina 220

[Esempio: Come configurare i messaggi espliciti \(MSG_CIPGENERIC\)](#) a pagina 217

Aggiunta di un blocco funzione COP, di variabili e di un contatto (MSG_CIPGENERIC)

Attenersi ai passaggi seguenti per aggiungere un blocco funzione COP, le variabili e un contatto. L'istruzione COP si utilizza per convertire i dati dal tipo di dati di origine (ad esempio DINT o REAL) al tipo di dati di destinazione. In questo esempio le informazioni sul catalogo vengono convertite in una stringa leggibile dall'utente.

Per aggiungere un blocco funzione COP:

1. Nella **Casella degli strumenti**, selezionare il **Ramo**, quindi trascinarlo e rilasciarlo direttamente sotto il primo ramo del ladder per aggiungere un secondo ramo.
2. Aggiungere il blocco di funzione COP:
 - Nella **Casella degli strumenti**, selezionare **Blocco**, quindi trascinarlo e rilasciarlo nel secondo ramo del ladder per aprire il **Selettore blocco istruzione**.

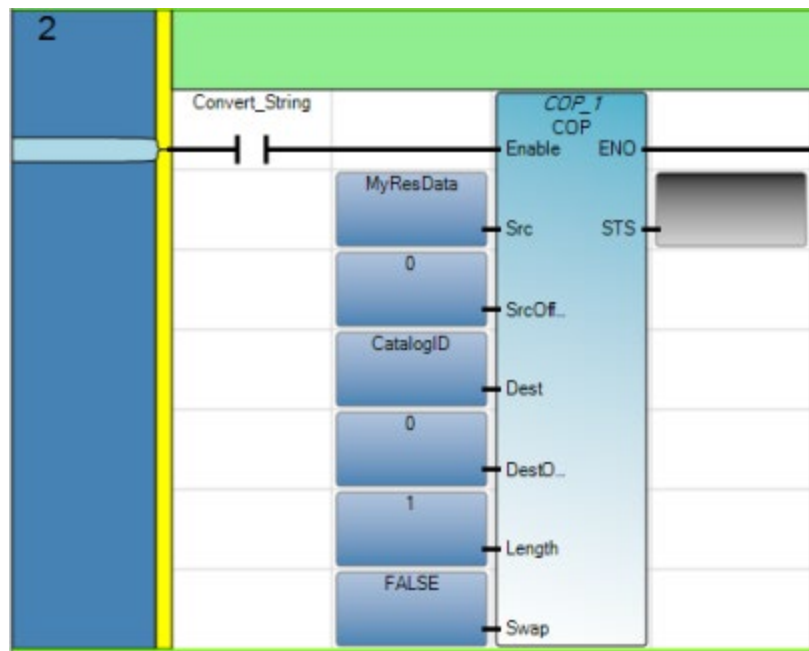
- Fare doppio clic su **COP** per aggiungere un'istanza del blocco di funzione al ladder diagram.
3. Aggiungere le variabili locali di ingresso per COP:
- Nell'**Organizzatore progetto**, fare doppio clic su **Variabili locali** per visualizzare la pagina **Variabili locali**.
 - Nella pagina **Variabili locali**, aggiungere le variabili e i tipi di dati elencati nella seguente tabella.

Parametri	Nome variabile	Tipo di dati
Src	MyResData	Array USINT
SrcOffset	0	UINT
Dest	CatalogID	Array STRING
DestOffset	0	UINT
Lunghezza	1	UINT
Swap	FALSE	BOOLEAN

4. Per la variabile CatalogID, fare doppio clic su **Dimensione** e modificare la dimensione del vettore su [1..1]
5. Aggiungere un contatto:
- Nella **casella degli strumenti**, selezionare **Contatto diretto**, quindi trascinarlo e rilasciarlo a sinistra dell'ingresso con il blocco di funzione COP sul secondo piolo del ladder.
 - Nel **selettore di variabili**, selezionare la variabile **Convert_String** per il contatto.

Risultato

Il secondo piolo del programma con ladder diagram per la messaggistica MSG_CIPGENERIC deve avere l'aspetto raffigurato nella seguente immagine.



Vedere anche

[Aggiunta di un contatto e di una bobina](#) a pagina 222

[Configurazione dei valori iniziali \(MSG_CIPGENERIC\)](#) a pagina 220

[Verifica della corretta configurazione IP sul controllore B](#) a pagina 225

Verifica della corretta configurazione IP sul controllore B

Per verificare se le impostazioni relative all'indirizzo IP del controllore B sono corrette, attenersi alla seguente procedura.

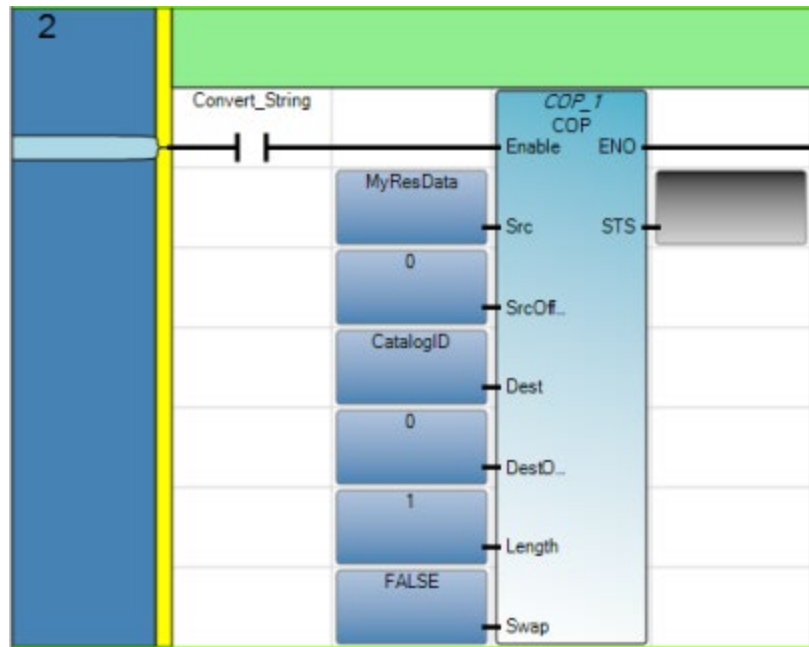
Per verificare l'indirizzo IP:

1. Aprire lo Spazio di Lavoro dell'applicazione di un controllore:
2. In **Organizzatore progetto** fare doppio clic sul controllore per aprire lo spazio di lavoro del controllore.
3. Nello Spazio di Lavoro del controllore, espandere **Ethernet** nella struttura ad albero del **controllore**, quindi fare clic su Protocollo IP per visualizzare la pagina di configurazione del controllore.
4. Verificare che le impostazioni relative all'indirizzo IP siano corrette, come indicato nella seguente tabella.

Opzione di configurazione IP	Valore
Indirizzo IP	192.168.1.19
Subnet mask	255.255.255.0
Indirizzo gateway	192.168.1.1

Risultati

Le opzioni relative al Protocollo IP nella pagina di configurazione del controllore devono avere l'aspetto raffigurato nella seguente immagine.



Vedere anche

[Aggiunta di un blocco funzione COP, di variabili e contatti](#) a pagina 223

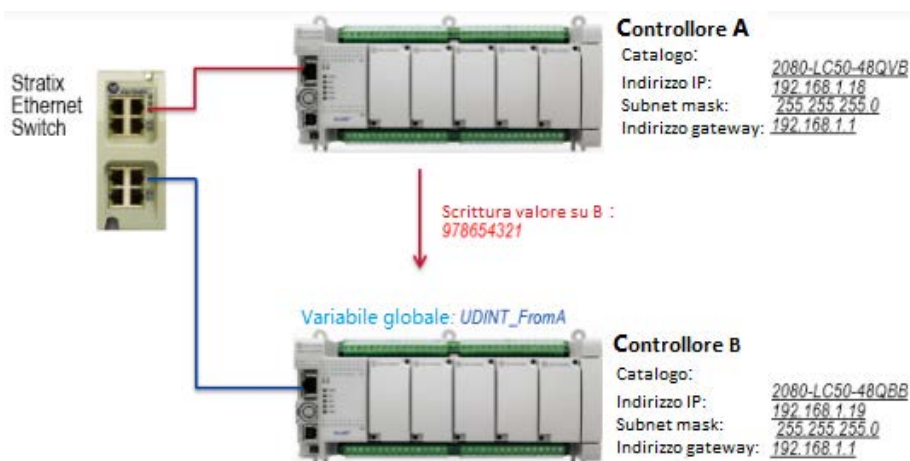
[Aggiunta di un blocco funzione MSG_CIPSYMBOLIC, di variabili e di un contatto](#) a pagina 230

[Creazione di una variabile globale per il controllore B](#) a pagina 233

[Esempio: Come creare un programma di generazione messaggi MSG_CIPSYMBOLIC per scrivere i dati in un controllore](#) a pagina 226

Esempio: Come creare un programma di generazione messaggi MSG_CIPSYMBOLIC per scrivere i dati in un controllore

In questo esempio viene illustrato come creare un programma di generazione messaggi per scrivere un valore dal controllore A alla variabile globale del controllore B.



Per creare un programma di messaggistica MSG_CIPSYMBOLIC utilizzabile per scrivere un valore in una variabile globale su un controllore diverso, eseguire le operazioni seguenti.

No	Attività
1	Aggiunta di un blocco funzione COP, di variabili e di un contatto (MSG_CIPSYMBOLIC) a pagina 227
2	Aggiunta di un operatore Equal e di una bobina a pagina 228
3	Aggiunta di un blocco funzione MSG_CIPSYMBOLIC, di variabili e di un contatto a pagina 230
4	Verifica della corretta configurazione IP sul controllore B a pagina 225
5	Creazione di una variabile globale per il controllore B a pagina 233
6	Revisione dei risultati del programma completo a pagina 234

Vedere anche

[Utilizzo dei blocchi funzione di comunicazione \(messaggistica\)](#) a pagina 215

Aggiunta di un blocco funzione COP, di variabili e di un contatto (MSG_CIPSYMBOLIC)

L'istruzione COP si utilizza per convertire i dati immessi nel tipo di dati di destinazione, in modo che sia compatibile con la variabile del controllore.

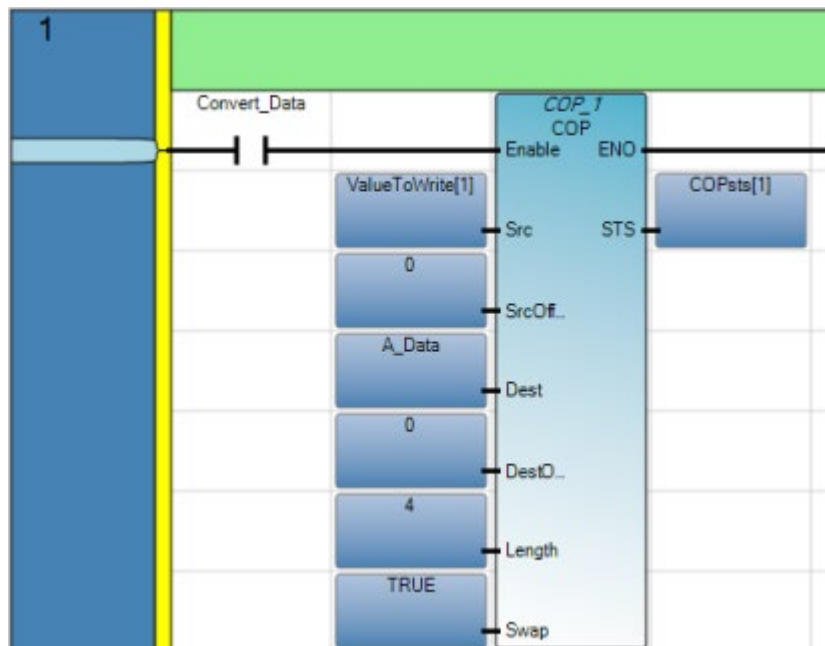
Per aggiungere un blocco funzione COP:

1. Aggiungere un controllore:
 - Espandere la cartella **Controllori** e la cartella Micro850 per visualizzare tutti i controllori Micro850.
 - Fare doppio clic su un controllore (2080-LC50-24QVB) per aggiungerlo all'**Organizzatore progetto**.
2. Aggiungere un programma con ladder diagram:
 - Nell'**Organizzatore progetto**, fare clic con il pulsante destro del mouse su **Programmi**, scegliere **Aggiungi**, quindi fare clic su **Nuovo LD: Ladder diagram**.
 - Fare clic con il pulsante destro del mouse sull'icona del diagramma ladder nell'**Organizzatore progetto**, selezionare **Rinominae** digitare **CIPSymbolicMessage**.
 - Fare doppio clic su Programmazione in Ladder Diagram nell'**Organizzatore progetto** per visualizzare la POU di LD nell'editor di lingue.
3. Aggiungere un blocco di funzione COP:
 - Nella **Casella degli strumenti**, selezionare **Blocco istruzione**, quindi trascinarlo e rilasciarlo nel primo ramo del ladder per aprire il **Selettore blocco istruzione**.
 - Nel campo di ricerca digitare **COP** e fare doppio clic su **COP** per aggiungere un'istanza del blocco di funzione al ladder diagram.
4. Aggiungere variabili COP e valori iniziali:
 - Nella POU del diagramma ladder, fare doppio clic su **Variabili locali** per aprire la pagina **Variabili locali**.
 - Nella pagina **Variabili** aggiungere le variabili e i tipi di dati elencati nella tabella riportata di seguito.
5. Creare array:
 - Per ValueToWrite, fare doppio clic su **Dimensione** e modificare la dimensione del vettore su [1..1].
 - Per A_Data, fare doppio clic su **Dimensione** e modificare la dimensione del vettore su [1..4].
6. Immettere i dati dalla colonna **Valore** della tabella riportata di seguito nel campo **Valore iniziale** di ciascuna variabile.
7. Aggiungere un contatto:
 - Nella **Casella degli strumenti**, selezionare **Contatto diretto**, quindi trascinarlo e rilasciarlo a sinistra dell'ingresso del blocco funzione COP sul primo ramo del ladder.
 - Nel **Selettore di variabili**, assegnare al contatto una variabile denominata **Convert_Data**.

Utilizzare le variabili definite nella tabella per il blocco funzione COP.

Parametri	Nome variabile	Tipo di dati
Src	ValueToWrite	Array UDINT Valore iniziale: 987654321
SrcOffset	0	UINT
Dest	A_Data	Array USINT
DestOffset	0	UINT
Lunghezza	4	UINT
Swap	TRUE	BOOLEAN
STS	COPsts	Array UINT

Il primo piolo del programma ladder diagram per la messaggistica MSG_CIPSYMBOLIC deve avere l'aspetto raffigurato nella seguente immagine.



Vedere anche

[Aggiunta di un operatore Equal e di una bobina](#) a pagina 228

[Esempio: Come creare un programma di generazione messaggi MSG_CIPSYMBOLIC per scrivere i dati in un controllore](#) a pagina 226

Aggiunta di un operatore Equal e di una bobina

L'istruzione Equal si utilizza per attivare un valore di scrittura, se la conversione del tipo di dati è eseguita correttamente. Per aggiungere un operatore Equal (=), variabili e una bobina, attenersi alla seguente procedura.

Per aggiungere un operatore Equal:

1. Nella **Casella degli strumenti**, selezionare il **Ramo**, quindi trascinarlo e rilasciarlo direttamente sotto il primo ramo del ladder per aggiungere un secondo ramo.
2. Aggiungere un operatore Equal:
 - Nella **Casella degli strumenti**, selezionare **Blocco istruzione**, quindi trascinarlo e rilasciarlo nel secondo ramo del ladder per aprire il **Selettore blocco istruzione**.
 - Nel campo di ricerca digitare il segno "=" e fare doppio clic su "=" per aggiungere un'istanza dell'operatore al ladder diagram.
3. Per aggiungere variabili Equal:
 - Nella POU del Diagramma ladder, fare doppio clic su una variabile per visualizzare il **Selettore di variabili**.
 - Nel **Selettore di variabili**, assegnare i nomi delle variabili, come elencato nella seguente tabella.

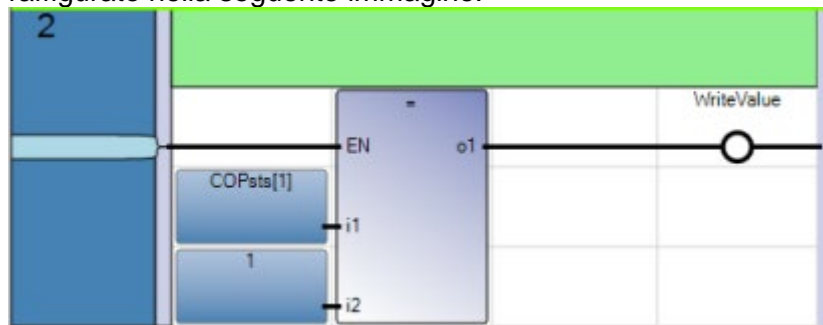
Parametri	Nome variabile
i1	COPsts
i2	1

4. Per aggiungere una bobina a un operatore Equal:

Nella **Casella degli strumenti**, selezionare **Bobina diretta**, quindi trascinarla e rilasciarla a destra dell'uscita dell'operatore Equal sul secondo ramo del ladder.

Nel **selettore di variabili**, digitare **WriteValue** nel campo Nome della bobina.

Il secondo piolo del programma con ladder diagram per la messaggistica MSG_CIPGENERIC deve avere l'aspetto raffigurato nella seguente immagine.



Vedere anche

[Aggiunta di un blocco di funzione COP, di variabili e di un contatto](#) a pagina 227

[Aggiunta di un blocco funzione MSG_CIPSYMBOLIC, di variabili e di un contatto](#) a pagina 230

[Esempio: Come creare un programma di generazione messaggi MSG_CIPSYMBOLIC per scrivere i dati in un controllore](#) a pagina 226

Aggiunta di un blocco funzione MSG_CIPSYMBOLIC, di variabili e di un contatto

Per aggiungere un blocco di funzione MSG_CIPSYMBOLIC, le variabili di ingresso e un contatto a un diagramma ladder, attenersi alla seguente procedura.

Per aggiungere blocco funzione e variabili:

1. Nella **casella degli strumenti** selezionare il **Ramo**, quindi trascinarlo e rilasciarlo direttamente sotto il secondo ramo del ladder per aggiungere un terzo ramo.
2. Aggiungere il blocco di funzione MSG_CIPSYMBOLIC:
 - Nella **casella degli strumenti** selezionare il **blocco istruzione**, quindi trascinarlo e rilasciarlo nel ramo del ladder per aprire il **Selettore di blocco istruzione**.
 - Nel campo di ricerca digitare **MSG** per visualizzare i blocchi di funzione per i messaggi.
 - Digitare **MSG_WriteVariable** nel campo **Istanza**.
 - Fare doppio clic su **MSG_CIPSYMBOLIC** per aggiungere un'istanza del blocco di funzione denominato MSG_WriteVariable al ladder diagram.
3. Assegnare nomi di variabili:
 - Nella POU del Diagramma ladder, fare doppio clic su una variabile per visualizzare il **Selettore di variabili**.
 - Nel **Selettore di variabili**, assegnare i nomi delle variabili, come elencato nella seguente tabella.
4. Per la variabile Data, fare doppio clic su **Dimensione** e modificare la dimensione del vettore su [1...4].
5. Configurare i valori iniziali per le variabili locali, aggiungere i valori iniziali CtrlCfg:
 - Nella pagina **Variabili locali** espandere il parametro CtrlCfg per visualizzare parametri aggiuntivi.
 - Immettere i seguenti valori nella colonna Valore iniziale di ciascun parametro.

Parametri	Valore iniziale
A_CtrlCfg.Cancel	Lasciare vuoto
A_CtrlCfg.TriggerType	300
A_CtrlCfg.StrMode	Lasciare vuoto

6. Aggiungere valori SymCfg iniziali:
 - Nella pagina **Variabili locali** espandere il parametro SymCfg per visualizzare parametri aggiuntivi.
 - Immettere i seguenti valori nella colonna Valore iniziale di ciascun parametro.

Parametri	Valore iniziale
A_SymCfg.Service	1
A_SymCfg.Symbol	'UDINT_FromA'
A_SymCfg.Count	Lasciare vuoto
A_SymCfg.DataType	200
A_SymCfg.Offset	Lasciare vuoto

Il Selettore di variabili locali deve avere l'aspetto raffigurato nella seguente immagine.

Nome	Alias	Tipo di dati	Dimensione	Valore progett	Valore iniziale
MSG_CIPSYMBOLIC_1		MSG_CIPGEN	
A_CtrlCfg		CIPCONTROL	
A_CtrlCfg.Cancel		BOOL			
A_CtrlCfg.TriggerType		UDINT			300
A_CtrlCfg.StrMode		USINT			
A_SymCfg		CIPSYMBOLIC	
A_SymCfg.Service		USINT			1
A_SymCfg.Symbol		STRING			'UDINT_FromA'
A_SymCfg.Count		UINT			
A_SymCfg.DataType		USINT			200
A_SymCfg.Offset		USINT			

7. Configurazione dei valori iniziali per TargetCfg

- Nella POU del ladder diagram fare doppio clic sulla variabile A_TarCfg per aprire il selettore di variabili globali.
- Espandere il parametro TargetCfg per visualizzare parametri aggiuntivi.
- Immettere i seguenti valori nella colonna Valore iniziale di ciascun parametro.

Parametri	Valore iniziale
A_TarCfg.Path	'4,192.168.1.19'
A_TarCfg.CipConnMode	1
A_TarCfg.UcmmTimeout	0
A_TarCfg.ConnMsgTimeout	0
A_TarCfg.ConnClose	Lasciare vuoto

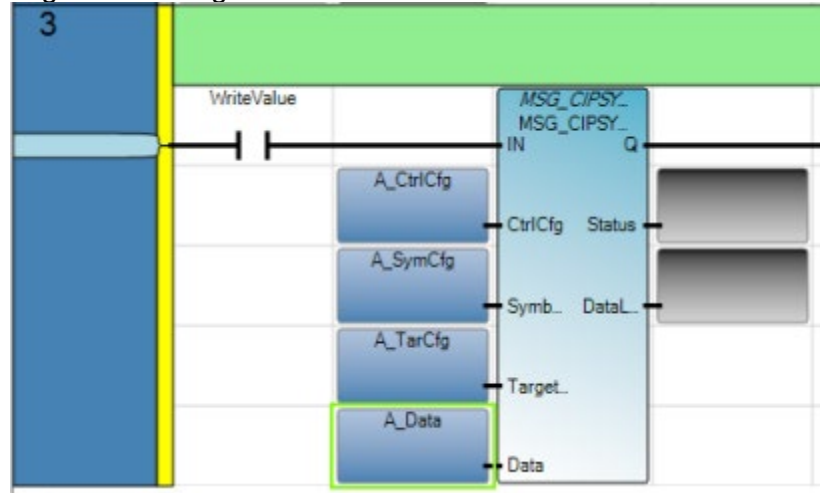
Il selettore di variabili globali create dall'utente deve avere l'aspetto raffigurato nella seguente immagine.

Nome	Alias	Tipo di dati	Dimensione	Valore progett	Valore iniziale
A_TarCfg		CIPTARGETCF	
A_TarCfg.Path		STRING			'4,192.168.1.19'
A_TarCfg.CipConnMode		USINT			
A_TarCfg.UcmmTimeout		UDINT			
A_TarCfg.ConnMsgTimeout		UDINT			
A_TarCfg.ConnClose		BOOL			

8. Aggiungere un contatto:

- Nella **Casella degli strumenti**, selezionare **Contatto diretto**, quindi trascinarlo e rilasciarlo a sinistra dell'ingresso del blocco funzione MSG_CIPSYMBOLIC sul terzo ramo del ladder.
- Nel **Selettore di variabili**, assegnare la variabile **WriteValue** al contatto.

Il terzo piolo del programma ladder diagram per la messaggistica MSG_CIPSYMBOLIC deve avere l'aspetto raffigurato nella seguente immagine.



Vedere anche

[Aggiunta di un operatore Equal e di una bobina](#) a pagina 228

[Esempio: Come creare un programma di generazione messaggi MSG_CIPSYMBOLIC per scrivere i dati in un controllore](#) a pagina 226

[Verifica della corretta configurazione IP sul controllore B](#) a pagina 225

Verifica della corretta configurazione IP sul controllore B

Per verificare se le impostazioni relative all'indirizzo IP del controllore B sono corrette, attenersi alla seguente procedura.

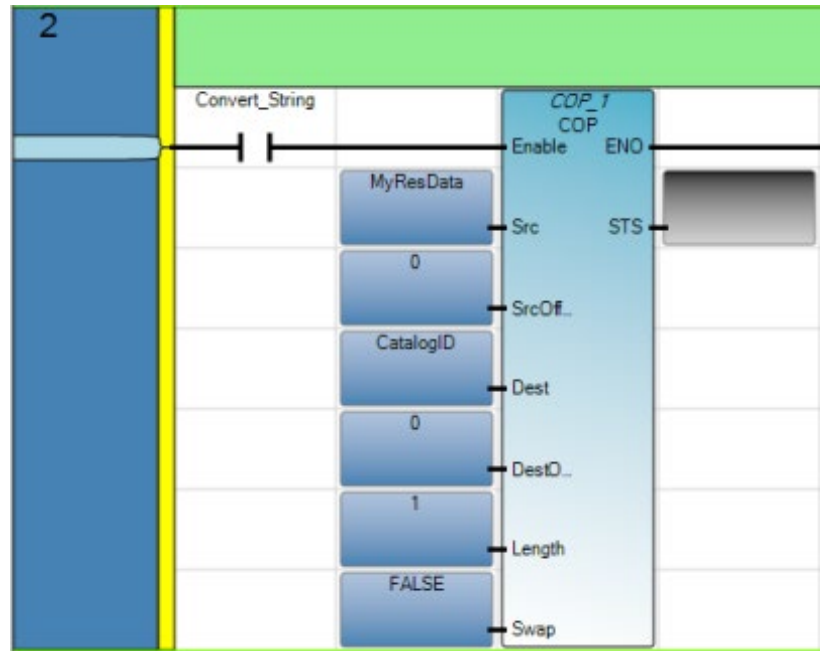
Per verificare l'indirizzo IP:

1. Aprire lo Spazio di Lavoro dell'applicazione di un controllore:
2. In **Organizzatore progetto** fare doppio clic sul controllore per aprire lo spazio di lavoro del controllore.
3. Nello Spazio di Lavoro del controllore, espandere **Ethernet** nella struttura ad albero del **controllore**, quindi fare clic su Protocollo IP per visualizzare la pagina di configurazione del controllore.
4. Verificare che le impostazioni relative all'indirizzo IP siano corrette, come indicato nella seguente tabella.

Opzione di configurazione IP	Valore
Indirizzo IP	192.168.1.19
Subnet mask	255.255.255.0
Indirizzo gateway	192.168.1.1

Risultati

Le opzioni relative al Protocollo IP nella pagina di configurazione del controllore devono avere l'aspetto raffigurato nella seguente immagine.



Vedere anche

[Aggiunta di un blocco funzione COP, di variabili e contatti a pagina 223](#)

[Aggiunta di un blocco funzione MSG_CIPSYMBOLIC, di variabili e di un contatto a pagina 230](#)

[Creazione di una variabile globale per il controllore B a pagina 233](#)

[Esempio: Come creare un programma di generazione messaggi MSG_CIPSYMBOLIC per scrivere i dati in un controllore a pagina 226](#)

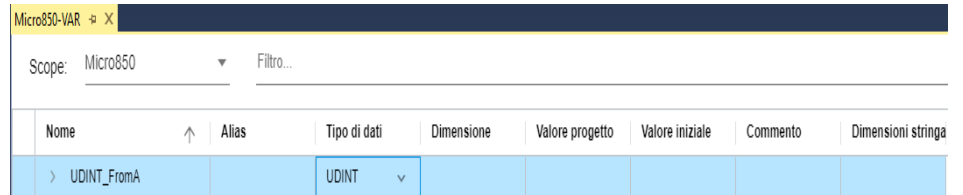
Creazione di una variabile globale per il controllore B

Per creare una variabile globale per il controllore B, attenersi alla seguente procedura.

Per creare una variabile globale:

1. Nell'**Organizzatore progetto**, fare doppio clic su **Variabili globali** per visualizzare il selettore di **variabili globali**.
2. Immettere UDINT_FromA nella colonna **Nome**.
3. Configurare gli altri parametri:
 - Verificare che il tipo di dati sia UDINT.
 - Verificare che l'attributo di **lettura/scrittura** sia selezionato.

Il selettore di **variabili globali** deve avere l'aspetto raffigurato nella seguente immagine.



Vedere anche

[Verifica della corretta configurazione IP sul controllore B](#) a pagina 225

[Revisione dei risultati del programma completo](#) a pagina 234

Revisione dei risultati del programma completo

Nell'esempio seguente è illustrato il programma di messaggistica MSG_CIPSYMBOLIC completo dopo la sua esecuzione.

Organizzatore progetto

Nome: QS_CIPG_CtrB

Dispositivi Tendenze

- Micro870
 - Programmi
 - Variabili globali
 - UDFB (Blocchi funzione definiti)
 - Funzioni definite dall'utente
 - Tipi di dati

Micro870-VAR

Scope: Micro870 Filtro...

Nome	Alias	Valore logico	Valore fisico
> UDINT_FromA		0	N/D
_IO_EM_DI_00		<input type="checkbox"/>	<input type="checkbox"/>
_IO_EM_DI_01		<input type="checkbox"/>	<input type="checkbox"/>
_IO_EM_DI_02		<input type="checkbox"/>	<input type="checkbox"/>
_IO_EM_DI_03		<input type="checkbox"/>	<input type="checkbox"/>
_IO_EM_DI_04		<input type="checkbox"/>	<input type="checkbox"/>

The screenshot shows a ladder logic program with two rungs. The first rung contains a COP (Copy) instruction. The second rung contains a MSG (Message) instruction.

Risultati controllore B

Le due immagini seguenti mostrano i valori del controllore B prima e dopo l'esecuzione del programma di messaggistica.

Prima dell'esecuzione del programma

Nome	Alias	Valore logico	Valore fisico
> UDINT_FromA		0	N/D
_IO_EM_DI_00		<input type="checkbox"/>	<input type="checkbox"/>
_IO_EM_DI_01		<input type="checkbox"/>	<input type="checkbox"/>
_IO_EM_DI_02		<input type="checkbox"/>	<input type="checkbox"/>
_IO_EM_DI_03		<input type="checkbox"/>	<input type="checkbox"/>
_IO_EM_DI_04		<input type="checkbox"/>	<input type="checkbox"/>

Dopo l'esecuzione del programma

Nome	Alias
> UDINT_FromA	
_IO_EM_DI_00	
_IO_EM_DI_01	
_IO_EM_DI_02	
_IO_EM_DI_03	
_IO_EM_DI_04	

Vedere anche

[Creazione di una variabile globale per il controllore B](#) a pagina 233

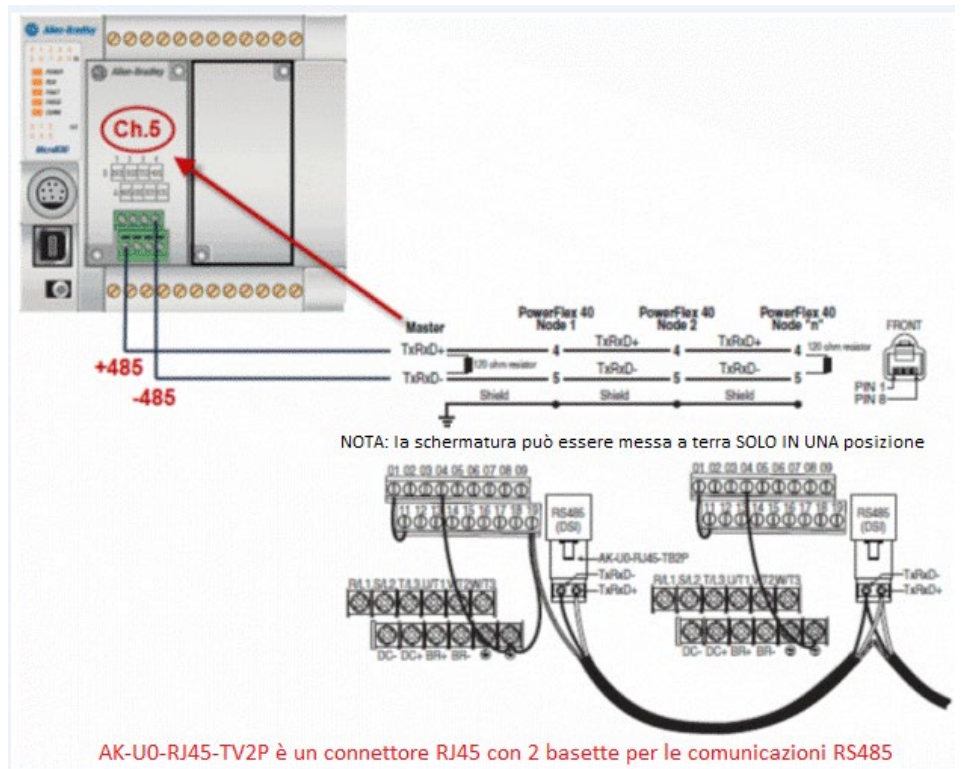
Esempio: Come configurare una comunicazione Modbus per la lettura da e la scrittura in una unità

Questi esempi descrivono come configurare le comunicazioni Modbus per leggere i dati dello stato da una unità PowerFlex 4 e scrivervi i dati di controllo utilizzando il blocco istruzione MSG_MODBUS.

Cablaggio Micro830

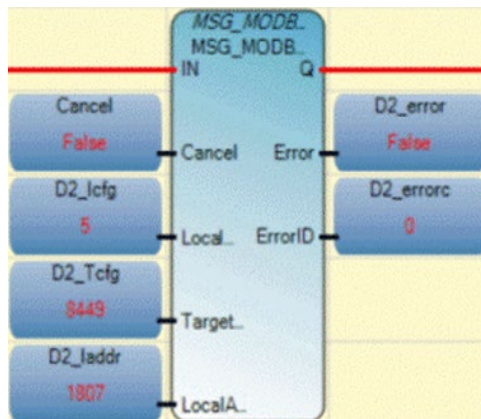
Questo esempio usa un controllore Micro830 con modulo SERIALISOL collegato nel primo slot (Canale 5). È collegato un PowerFlex 40

singolo, ma il diagramma seguente mostra come cablare per multi-drop. Per maggiori informazioni sul cablaggio, fare riferimento al manuale utente.



Esempio di lettura Modbus

La seguente istruzione MSG_MODBUS può essere usata per leggere i dati dello stato dall'unità PowerFlex 40.



Stato dell'unità

Il valore "1807" indica che l'unità è

- Pronta (bit 0 ON)
- Attiva (bit 1 ON)
- Marcia av. coman. (bit 2 ON)
- Rotaz. avanti (bit 3 ON)
- Stato di alcuni ingressi digitali dell'unità

Il valore "278" indica 27,8 Hz.

Per maggiori informazioni sui bit della parola stato logico, sulle descrizioni dei codici di errore, sulle velocità effettive e comandate e altri codici di stato, fare riferimento al manuale utente PowerFlex.

Configurazione lettura MSG_MODBUS

Nell'immagine seguente sono illustrate le opzioni relative al blocco di istruzioni MSG_MODBUS usato per leggere i dati dello stato dall'unità PowerFlex 40.

	Nome	Tipo di dati	Direction	Dimensione
+	MSG_MODBUS_1	MSG_MODI	Var	
-	D2_Icfg	MODBUSLC	Var	
	D2_Icfg.Channel	UINT	Var	
	D2_Icfg.TriggerType	USINT	Var	
	D2_Icfg.Cmd	USINT	Var	
	D2_Icfg.ElementCnt	UINT	Var	
-	D2_Tcfg	MODBUSF	Var	
	D2_Tcfg.Addr	UDINT	Var	
	D2_Tcfg.Node	USINT	Var	
▶	D2_Jaddr	MODBUSLC	Var	
	D2_Jaddr[1]	WORD	Var	
	D2_Jaddr[2]	WORD	Var	
	D2_Jaddr[3]	WORD	Var	
	D2_Jaddr[4]	WORD	Var	
	D2_Jaddr[5]	WORD	Var	

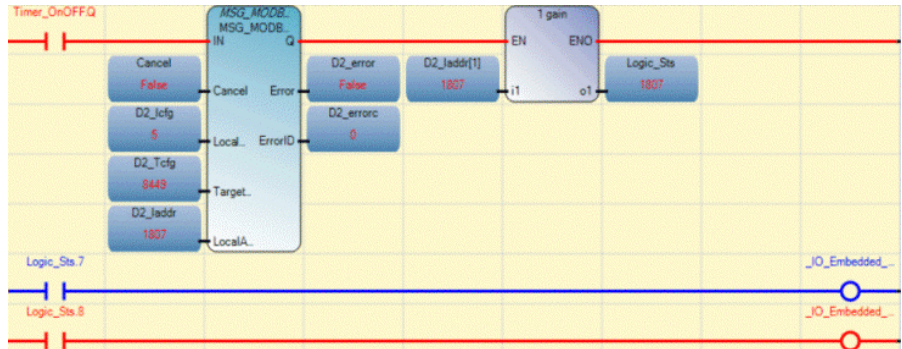
Variabili lettura MSG_MODBUS

Nella seguente tabella sono riportate le variabili e i valori utilizzati per configurare l'istruzione MSG_MODBUS in modo tale da leggere i dati dello stato dall'unità PowerFlex 40.

Variabile	Valore	Descrizione
*.Channel	5	Canale 5: posizione del modulo SERIALISOL
*.TriggerType	0	Trigger per transizioni da False a True
*.Cmd	3	Codice funzione Modbus "03": lettura registri di mantenimento
*.ElementCnt	4	Lunghezza
*.Addr	8449	Indirizzo parola stato logico PowerFlex + 1
*.Node	2	Indirizzo nodo PowerFlex
*_laddr[1]	{data}	Parola stato logico PowerFlex
*_laddr[2]	{data}	Codice di errore PowerFlex
*_laddr[3]	{data}	Velocità comandata PowerFlex (riferimento velocità)
*_laddr[4]	{data}	Feedback velocità PowerFlex (velocità effettiva)

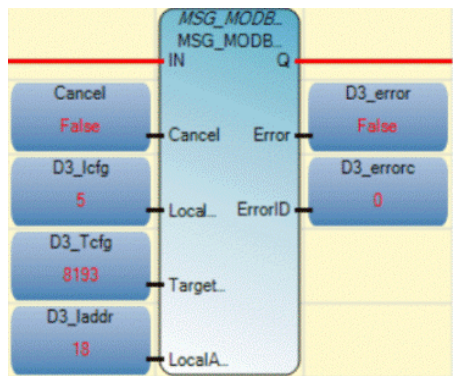
Esempio di istruzione MOV

Nella seguente immagine è illustrato un esempio d'uso dell'istruzione MOV per spostare il valore dell'array *_I[1] in una parola, in modo tale da accedere direttamente ai bit individuali.



Esempio di scrittura Modbus

La seguente istruzione MSG_MODBUS consente di scrivere i dati di controllo nell'unità PowerFlex 40.



Configurazione scrittura MSG_MODBUS

Nella seguente immagine sono illustrate le variabili e i valori utilizzati per configurare l'istruzione MSG_MODBUS in modo tale da scrivere i dati di controllo nell'unità PowerFlex 40.

Nome	Tipo di dati	Direction	Dimensione
D3_Icfg	MODBUSLK	Var	
D3_Icfg.Channel	UINT	Var	
D3_Icfg.TriggerType	USINT	Var	
D3_Icfg.Cmd	USINT	Var	
D3_Icfg.ElementCnt	UINT	Var	
D3_Tcfg	MODBUSLK	Var	
D3_Tcfg.Addr	LDINT	Var	
D3_Tcfg.Node	USINT	Var	
D3_Iaddr	MODBUSLK	Var	
D3_Iaddr[1]	WORD	Var	
D3_Iaddr[2]	WORD	Var	
D3_Iaddr[3]	WORD	Var	

Variabili di scrittura MSG_MODBUS

La tabella seguente elenca le variabili con i relativi valori e ne descrive lo scopo.

Variabile	Valore	Descrizione
*.Channel	5	Canale 5: posizione del modulo SERIALISOL
*.TriggerType	0	Trigger per transizioni da False a True
*.Cmd	16	Codice funzione Modbus "16": scrittura registri di mantenimento
*.ElementCnt	2	Lunghezza
*.Addr	8193	Indirizzo parola stato logico PowerFlex + 1
*.Node	2	Indirizzo nodo PowerFlex
*._laddr[1]	{data}	Parola comando logico PowerFlex
*._laddr[2]	{data}	Parola riferimento velocità PowerFlex

Vedere anche

[MSG_MODBUS](#) a pagina 196

[Utilizzo dei blocchi funzione di comunicazione \(messaggistica\)](#) a pagina 215

Supporto per protocolli di comunicazione

I blocchi funzione MSG_CIP supportano diversi protocolli di comunicazione, come descritto in questa sezione.

Supporto del blocco funzione per i protocolli di comunicazione dei messaggi.

Protocollo di messaggistica	Supporto per comunicazione	Utilizzare questo blocco funzione
Server e client Modbus/RTU	Tramite una porta seriale configurata come Modbus RTU.	MSG_MODBUS a pagina 196
Server e client Modbus/TCP	Tramite Ethernet, invece di una porta seriale.	MSG_MODBUS2 a pagina 202
Server e client Ethernet IP	Mediante un canale Ethernet integrato.	MSG_CIPSYMBOLIC a pagina 191 MSG_CIPGENERIC a pagina 182
Server e client CIP seriale	Cavo Ethernet o seriale.	MSG_CIPSYMBOLIC a pagina 191
ASCII	Tramite una porta RS-232 configurata con un driver ASCII.	Vedere le istruzioni della porta seriale ASCII.

Modbus RTU

Modbus è un protocollo per le comunicazioni master-slave, half-duplex che permette a un singolo master di comunicare con un massimo di 247 dispositivi slave. Il master di rete Modbus legge e scrive i bit e i registri. Modbus RTU è supportato configurando la porta seriale come Modbus RTU.

Per maggiori informazioni sul protocollo Modbus, fare riferimento alle relative specifiche, disponibili all'indirizzo <http://www.modbus.org>.

Modbus/TCP

Il protocollo di comunicazione server Modbus/TCP utilizza le stesse funzioni di mapping Modbus di Modbus RTU, ma è supportato via Ethernet, anziché tramite una porta seriale.

Il controllore Micro850 supporta fino a 16 connessioni server Modbus/TCP simultanee. L'unica configurazione di protocollo richiesta è quella della tabella di mapping Modbus.

EtherNet/IP

I controllori Micro820 e Micro850 supportano fino a 16 connessioni server EtherNet/IP simultanee tramite un canale di comunicazione Ethernet integrato. Questo canale consente di connettere un controllore Micro850 a vari dispositivi tramite una rete locale con velocità di trasferimento 10 Mbps/100 Mbps.

Common Industrial Protocol (CIP)

Il protocollo CIP è un protocollo di trasporto, a livello di applicazione, impiegato per la messaggistica su reti EtherNet/IP, ControlNet e DeviceNet. Questo protocollo permette di connettere facilmente i moduli HMI a un controllore Micro830 o Micro850.

Messaggistica esplicita CIP

Per la messaggistica esplicita si utilizza il protocollo CIP. La messaggistica esplicita (configurazione, raccolta dati e diagnostica) serve a trasferire i dati che non richiedono l'aggiornamento costante. La messaggistica esplicita è supportata con la messaggistica CIP client generica mediante il blocco funzione MSG_CIPGENERIC e con la messaggistica CIP client simbolica mediante il blocco funzione MSG_CIPSYMBOLIC.

CIP seriale

Il CIP seriale utilizza il protocollo Full-Duplex DF1 e fornisce una connessione da punto a punto tra due dispositivi. Combina la trasparenza dei dati (specifica American National Standards Institute, ANSI - X3.28-1976 sottocategoria D1) e la trasmissione simultanea a 2 vie con risposte integrate (sottocategoria F1)

I controllori Micro830 e Micro850 supportano il CIP seriale tramite una connessione RS-232 ai dispositivi esterni, quali computer con software RSLinx Classic in esecuzione, terminali del componente PanelView (revisioni del firmware 1.70 e successive) o altri controllori che supportano il CIP seriale via DF1 full-duplex, come i controllori ControlLogix e CompactLogix dotati di porte seriali integrate.

Il driver della porta seriale può essere configurato come CIP seriale, Modbus RTU, ASCII o Shutdown tramite la struttura di configurazione del dispositivo.

Vedere anche

[Istruzioni porta seriale ASCII](#) a pagina 111

[Canali di comunicazione integrati](#) a pagina 242

Canali di comunicazione integrati

I controllori Micro820, Micro830 e Micro850 dispongono dei seguenti canali di comunicazione aggiuntivi integrati:

- Una porta combinata RS-232/485 non isolata
- Una porta di programmazione USB non isolata (solo Micro830 e Micro850)
- Una porta Ethernet RJ-45 (solo Micro820 e Micro850)

Vedere anche

[Supporto per protocolli di comunicazione](#) a pagina 240

Istruzioni per il confronto

Utilizzare le istruzioni Compare per confrontare i valori Integer, Real, Time, Date e String utilizzando un'espressione o un'istruzione di confronto specifica.

Istruzione	Descrizione
(=) Equal a pagina 243	Confronta il primo ingresso con il secondo per stabilire l'uguaglianza. Per i tipi di dati Integer, Real, Time, Date e String.
(>) Greater Than a pagina 246	Confronta i valori di ingresso per determinare se il primo è maggiore del secondo.
(>=) Greater Than or Equal a pagina 247	Confronta i valori di ingresso per determinare se il primo è maggiore o uguale al secondo.
(<) Less Than a pagina 249	Confronta i valori di ingresso per determinare se il primo è minore del secondo.
(<=) Less Than or Equal a pagina 250	Confronta i valori di ingresso per determinare se il primo è minore o uguale al secondo.
(<>) Not Equal a pagina 251	Confronta i valori di ingresso per determinare se il primo non è uguale al secondo.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

Uguale

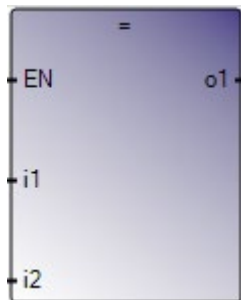
Esegue il confronto tra il primo ingresso e il secondo per stabilire l'uguaglianza di tipi di dati Integer, Real, Time, Date e String.

Dettagli operazione:

- La prova di uguaglianza dei valori Time non è raccomandata per i blocchi istruzione TON, TP e TOF.
- Il tipo di dati Real è sconsigliato durante il confronto dei valori per l'uguaglianza, poiché i numeri nell'operazione matematica sono arrotondati in maniera diversa rispetto a quelli visualizzati sul display di Uscita delle variabili. Di conseguenza, due valori di uscita possono sembrare uguali sul display, tuttavia la valutazione risulterà comunque falsa. Ad esempio, dal confronto di 23,500001 con 23,499999, entrambi vengono visualizzati come 23,5 sul display di ingresso delle variabili, ma nel controllore non saranno uguali.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

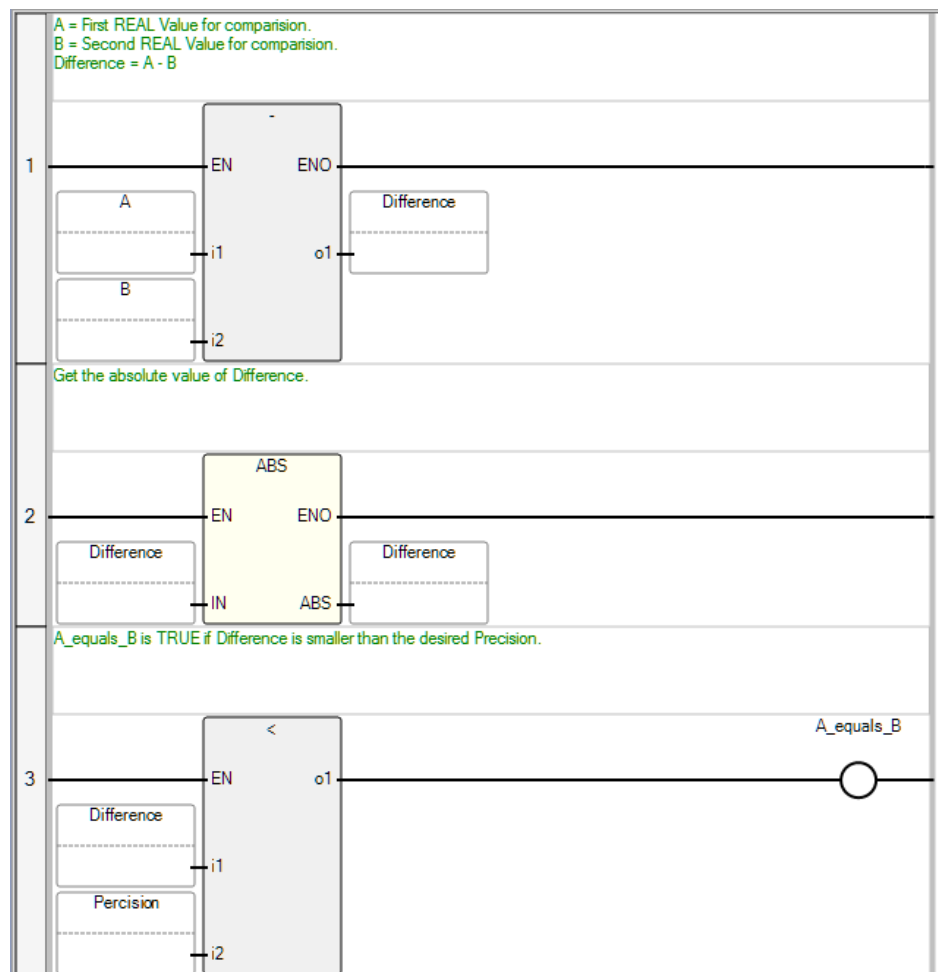
Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitare funzione. TRUE: esegue il confronto di uguaglianza. FALSE: il confronto non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Tutti gli ingressi devono essere dello stesso tipo di dati. L'inserimento del valore Durata si applica ai linguaggi Testo strutturato, Diagramma ladder e Programmazione a blocchi funzionali. il tipo di dati Real non è raccomandato.
i2	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	
o1	Uscita	BOOL	TRUE se i1 = i2.

Esempio di confronto dei valori reali mediante sottrazione (-) ABS e Less than (<)

Il tipo di dati Reale non è consigliabile per confrontare l'uguaglianza dei valori a causa delle differenze nell'arrotondamento dei numeri. Due valori di uscita potrebbero sembrare uguali su un display Connected Components Workbench, ma comunque valutati come false.

Ad esempio, 23,500001 e 23,499999 saranno visualizzati entrambi come 23,5 in sul display di ingresso delle variabili, tuttavia nel controllore non saranno uguali.

Per verificare se due dati di tipo reale sono uguali, è possibile usare l'istruzione Sottrazione per ottenere la differenza tra i valori e quindi determinare se la differenza è inferiore a un valore con precisione stabilita. Vedere l'esempio seguente di programma LD per il confronto tra i valori di due dati di tipo reale.



Esempio di testo strutturato Equal (=)

(* Equivalenza ST: *)

```
areult := (10 = 25); (* areult è FALSE *)
```

```
mresult := ('ab' = 'ab'); (* mresult è TRUE *)
```

Vedere anche

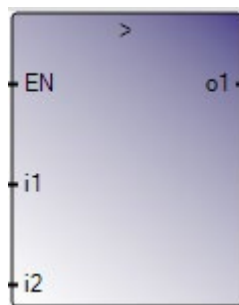
[Istruzioni per il confronto](#) a pagina 243

Greater than

Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo è maggiore del secondo.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il confronto dell'ingresso. FALSE: il confronto non viene eseguito. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Tutti gli ingressi devono essere dello stesso tipo di dati.

i2	Ingresso	SINT USINT - BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	
o1	Uscita	BOOL	TRUE se i1 > i2.

Esempio di Testo strutturato maggiore di (>)

(* Equivalenza ST: *)

```

aresult := (10 > 25); (* aresult è FALSE *)
mresult := ('ab' > 'a'); (* mresult è TRUE *)

```

Vedere anche

[Istruzioni per il confronto](#) a pagina 243

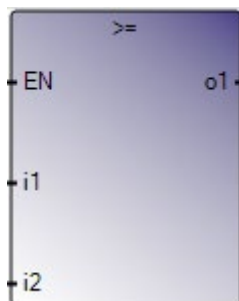
Greater than or equal

Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo è maggiore o uguale al secondo.

Per TON, TP e TOF, la prova di uguaglianza dei valori Time non è consigliata.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il confronto dell'ingresso. FALSE: il confronto non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Tutti gli ingressi devono essere dello stesso tipo di dati. L'inserimento del valore Durata si applica ai linguaggi Testo strutturato, Diagramma ladder e Programmazione a blocchi funzionali.
i2	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	
o1	Uscita	BOOL	TRUE se i1 >= i2.

Esempio di Testo strutturato maggiore o uguale (>=)

(* Equivalenza ST: *)

```
aresult := (10 >= 25); (* aresult è FALSE *)
mresult := ('ab' >= 'ab'); (* mresult è TRUE *)
```

Vedere anche

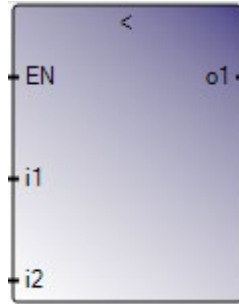
[Istruzioni per il confronto](#) a pagina 243

Less than

Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo è inferiore al secondo.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il confronto dell'ingresso. FALSE: il confronto non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Tutti gli ingressi devono essere dello stesso tipo di dati.

i2	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	
o1	Uscita	BOOL	TRUE se il < i2.

Esempio di testo strutturato minore di (>)

(* Equivalenza ST: *)

```

aresult := (10 < 25); (* aresult è TRUE *)
mresult := ('z' < 'B'); (* mresult è FALSE *)
    
```

(* Equivalenza IL: *)

LD	10
LT	25
ST	aresult
LD	'z'
LT	'B'
ST	mresult

Vedere anche

[Istruzioni per il confronto](#) a pagina 243

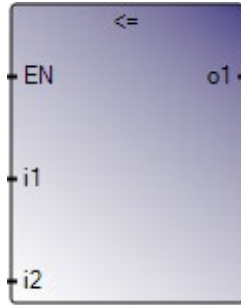
Less than or equal

Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo è inferiore o uguale al secondo.

Per TON, TP e TOF, la prova di uguaglianza dei valori Time non è consigliata.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il confronto dell'ingresso. FALSE: il confronto non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	SINT - USINT - BYTE - INT - UINT - WORD - DINT - UDINT - DWORD - LINT - ULINT - LWORD - REAL - LREAL - TIME - DATE - STRING	Tutti gli ingressi devono essere dello stesso tipo di dati. L'inserimento del valore Durata si applica ai linguaggi Testo strutturato, Diagramma ladder e Programmazione a blocchi funzionali.
i2	Ingresso	SINT - USINT - BYTE - INT - UINT - WORD - DINT - UDINT - DWORD - LINT - ULINT - LWORD - REAL - LREAL - TIME - DATE - STRING	
o1	Uscita	BOOL	TRUE se i1 <= i2.

Esempio di testo strutturato minore o uguale (> =)

(* Equivalenza ST: *)

```
aresult := (10 <= 25); (* aresult è TRUE *)
mresult := ('ab' <= 'ab'); (* mresult è TRUE *)
```

Vedere anche

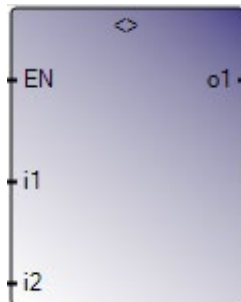
[Istruzioni per il confronto](#) a pagina 243

Not equal

Per i valori Integer, Real, Time, Date e String, confronta i valori di ingresso per stabilire se il primo non è uguale al secondo.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo di confronto corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Tutti gli ingressi devono essere dello stesso tipo di dati.
i2	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	
o1	Uscita	BOOL	TRUE: se il primo <math><></math> secondo.

Esempio di Testo strutturato Not equal (<>)

(* Equivalenza ST: *)

```
aresult := (10 <> 25); (* aresult è TRUE *)  
mresult := ('ab' <> 'ab'); (* mresult è FALSE *)
```

Vedere anche

[Istruzioni per il confronto](#) a pagina 243

Istruzioni contatore

Utilizzare le istruzioni contatore per controllare le operazioni in base al numero di eventi.

Istruzione	Descrizione
CTD a pagina 255	Conta gli Integer da un valore dato fino a 0, 1 a 1.
CTU a pagina 257	Conta gli Integer da 0 fino a un valore dato, 1 a 1.
CTUD a pagina 259	Conta gli Integer da 0 fino a un valore dato, 1 a 1, o da un valore dato a 0, 1 a 1.

Vedere anche

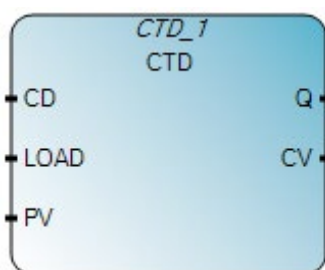
[Istruzione impostate in ordine alfabetico](#) a pagina 18

CTD (conteggio decrescente)

Conta gli Integer da un valore dato fino a 0, 1 a 1.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

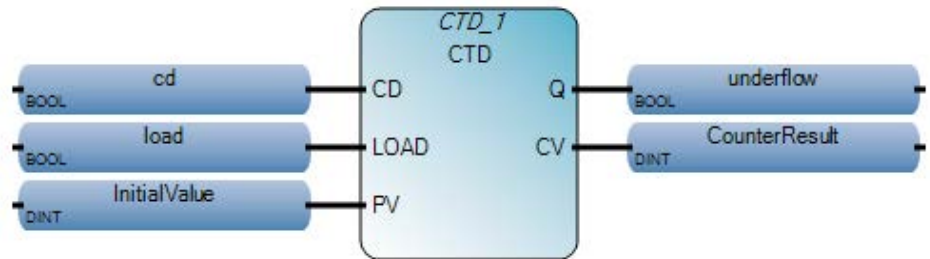


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

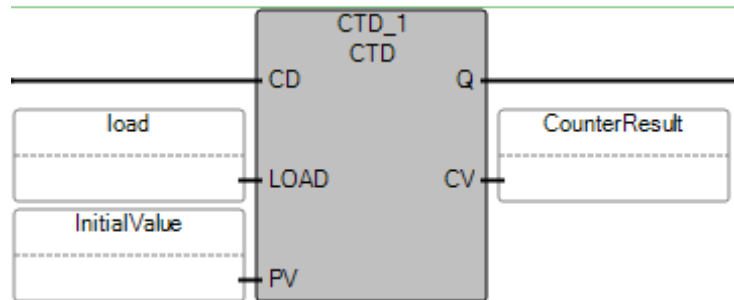
Parametro	Tipo di parametro	Tipo di dati	Descrizione
CD	Ingresso	BOOL	Esegue il conteggio decrescente. TRUE: fronte di salita rilevato, conteggio decrescente con incrementi di uno. FALSE: fronte di discesa rilevato, tenere premuto il valore del contatore con lo stesso valore.
LOAD	Ingresso	BOOL	Carica verifica il valore PV rispetto al valore del conteggio decrescente. TRUE - set CV=PV. FALSE: continua incrementando il conteggio decrescente di uno.
PV	Ingresso	DINT	Valore massimo programmato del contatore.

Q	Uscita	BOOL	Indica se l'istruzione del conteggio decrescente ha generato un numero minore o uguale al valore massimo del contatore. TRUE: risultato contatore <= 0 (condizione di Underflow). FALSE: risultato contatore >0.
CV	Uscita	DINT	Valore del contatore corrente.

Esempio di programmazione a blocchi funzionali CTD



Esempio di Diagramma ladder CTD



Esempio di testo strutturato CTD

```
CTD_1 {
    void CTD_1(BOOL CD, BOOL LOAD, DINT PV)
    Tipo: CTD, Contatore down
```

```
1 | InitialValue := 10;
2 | CTD_1(cd, load, InitialValue);
```

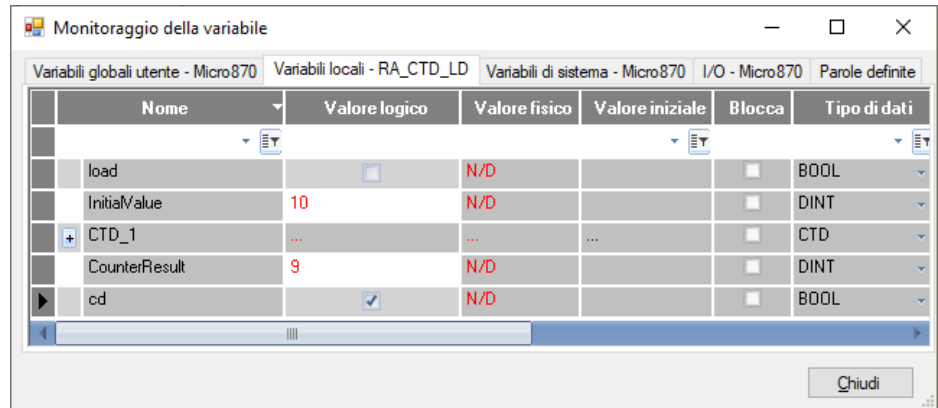
(*Equivalenza ST: CTD1 è un'istanza del blocco *)

CTD1(trigger,load_cmd,100);

underflow := CTD1.Q;

result := CTD1.CV;

Risultati



Vedere anche

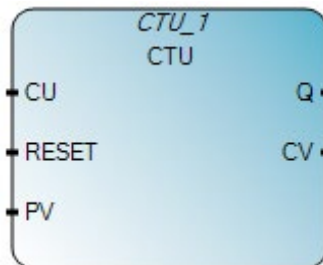
[Istruzioni contatore](#) a pagina 255

CTU (conteggio crescente)

CTU conta (numeri interi) da 0 fino a un valore dato, 1 a 1.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

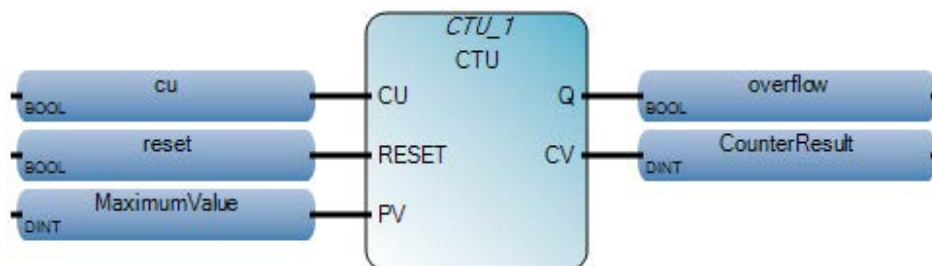
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



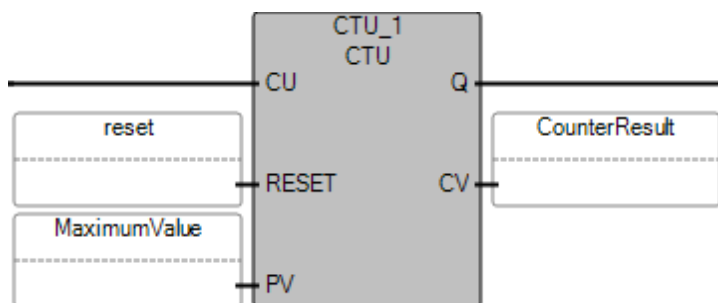
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
CU	Ingresso	BOOL	Esegue il conteggio crescente. TRUE: fronte di salita rilevato, conteggio crescente con incrementi di uno. FALSE: fronte di discesa rilevato, tenere premuto il valore del contatore con lo stesso valore.
RESET	Ingresso	BOOL	Reset verifica il valore PV rispetto al valore del conteggio crescente. TRUE: imposta il valore CV a zero. FALSE: continuare incrementando il conteggio crescente di uno.
PV	Ingresso	DINT	Valore massimo programmato del contatore.
Q	Uscita	BOOL	Indica se l'istruzione del conteggio crescente ha generato un numero maggiore o uguale al valore massimo del contatore. TRUE: risultato contatore =>PV (condizione di Overflow). FALSE: risultato contatore < PV
CV	Uscita	DINT	Risultato del contatore corrente.

Esempi di programmazione a blocchi funzionali CTU



Esempio di Diagramma ladder CTU



Esempio di testo strutturato CTU

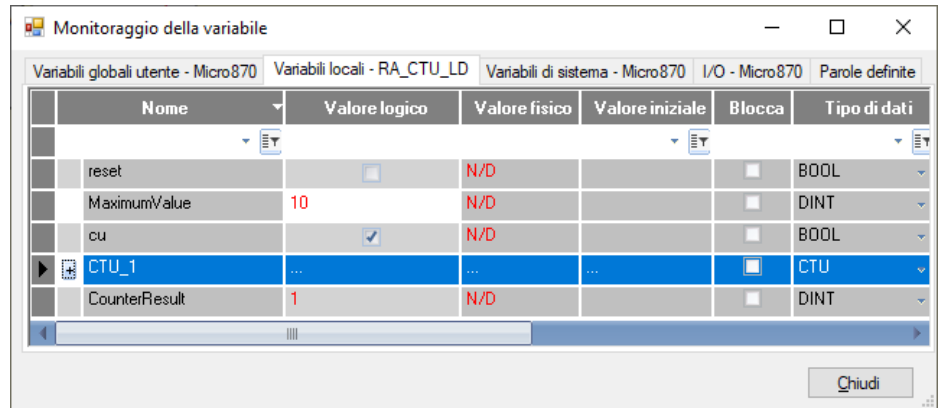
```
CTU_1(
void CTU_1(BOOL CU, BOOL RESET, DINT PV)
Tipo: CTU, Contatore up
```

```
1 MaximumValue := 10;
2 CTU_1(cu, reset, MaximumValue);
```

(* Equivalenza ST: CTU1 è un'istanza del blocco CTU*)

```
CTU1(trigger,NOT(auto_mode),100);
overflow := CTU1.Q;
result := CTU1.CV;
```

Risultati



Vedere anche

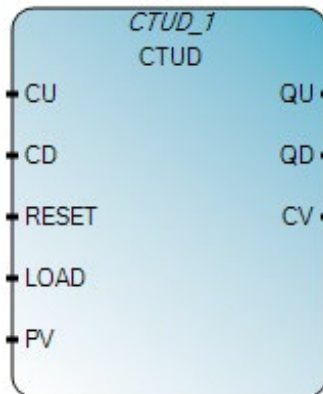
[Istruzioni contatore](#) a pagina 255

CTUD (conteggio crescente decrescente)

Conta gli Integer da 0 fino a un valore dato, 1 a 1, o da un valore dato a 0, 1 a 1.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

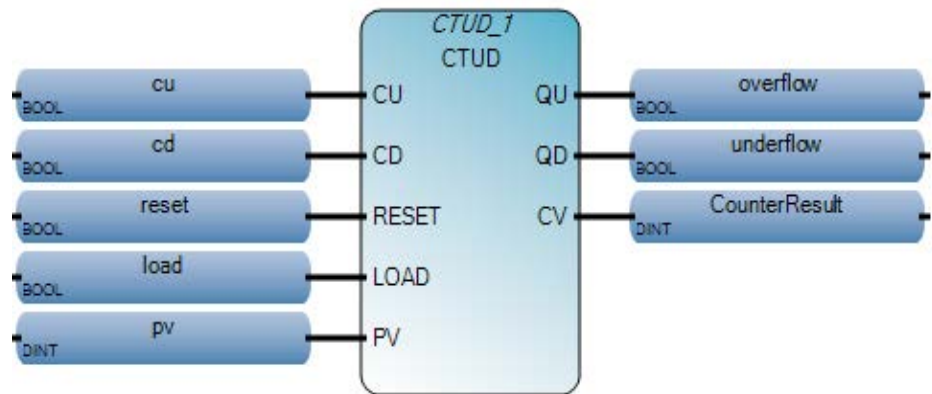


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

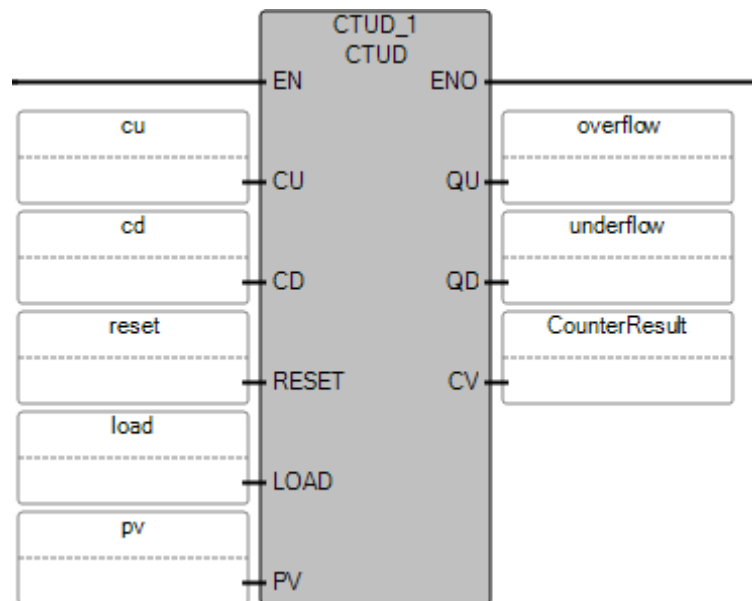
Parametro	Tipo di parametro	Tipo di dati	Descrizione
CU	Ingresso	BOOL	TRUE: fronte di salita rilevato, conteggio crescente.
CD	Ingresso	BOOL	TRUE: fronte di salita rilevato, conteggio decrescente.
RESET	Ingresso	BOOL	Ripristina comando dominante (priorità massima durante la definizione del comportamento delle istruzioni). (CV = 0 quando RESET è TRUE).
LOAD	Ingresso	BOOL	Carica comando. TRUE: imposta CV = PV.
PV	Ingresso	DINT	Valore massimo programmato.

QU	Uscita	BOOL	Overflow. TRUE: se CV >= PV.
QD	Uscita	BOOL	Underflow. TRUE: se CV <= 0.
CV	Uscita	DINT	Risultato contatori.

Esempio di programmazione a blocchi funzionali CTUD



Esempio di diagramma ladder CTUD



Esempio di testo strutturato CTUD

```

1| cu := TRUE;
2| cd := TRUE;
3| reset := FALSE;
4| load := FALSE;
5| pv := 10;
6| CTUD_1(cu, cd, reset, load, pv);

```

CTUD_1(|

```

void CTUD_1(BOOL CU, BOOL CD, BOOL RESET, BOOL LOAD, DINT PV)
Tipo: CTUD, Contatore up/down

```

(* Equivalenza ST: supponiamo che CTUD1 sia un'istanza del blocco*)

```

CTUD1(trigger1, trigger2, reset_cmd, load_cmd,100);
full := CTUD1.QU;
empty := CTUD1.QD;
nb_elt := CTUD1.CV;

```

Vedere anche

[Istruzioni contatore](#) a pagina 255

Istruzioni per la conversione dati

Utilizzare le istruzioni di conversione dei dati per convertire il tipo di dati di una variabile in un tipo diverso.

Istruzione	Descrizione
ANY_TO_BOOL a pagina 263	Converte un valore non booleano in uno booleano.
ANY_TO_BYTE a pagina 264	Converte un valore in un byte.
ANY_TO_DATE a pagina 265	Converte un tipo di dati String, Integer, Real o Time in un tipo di dati Date.
ANY_TO_DINT a pagina 267	Converte un valore in Double Integer.
ANY_TO_DWORD a pagina 268	Converte un valore in un valore Double Word.
ANY_TO_INT a pagina 269	Converte un valore in Integer.
ANY_TO_LINT a pagina 270	Converte un valore in Long Integer.
ANY_TO_LREAL a pagina 272	Converte un valore in Long Real.
ANY_TO_LWORD a pagina 273	Converte un valore in Long Word.
ANY_TO_REAL a pagina 274	Converte un valore in Real.
ANY_TO_SINT a pagina 275	Converte un valore in Short Integer.
ANY_TO_STRING a pagina 276	Converte un valore in String.
ANY_TO_TIME a pagina 277	Converte un valore in un tipo di dati Time.
ANY_TO_UDINT a pagina 278	Converte un valore in Unsigned Double Integer.
ANY_TO_UINT a pagina 279	Converte un valore in Unsigned Integer.
ANY_TO_ULINT a pagina 280	Converte un valore in Unsigned Long Integer.
ANY_TO_USINT a pagina 281	Converte un valore in Unsigned Short Integer.
ANY_TO_WORD a pagina 282	Converte un valore in Word.

Vedere anche

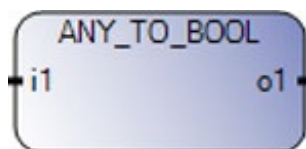
[Istruzione impostate in ordine alfabetico](#) a pagina 18

ANY_TO_BOOL

Converte un valore non booleano in un valore booleano.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Booleano. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
i1	Ingresso	SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore non Booleano.
o1	Uscita	BOOL	Valore Booleano.

Esempio di testo strutturato ANY_TO_BOOL

(* Equivalenza ST: *)

```
ares := ANY_TO_BOOL (I0);           (* ares è TRUE *)
tres := ANY_TO_BOOL (t#0s);        (* tres è FALSE *)
mres := ANY_TO_BOOL ('FALSE');     (* mres è FALSE *)
```

Vedere anche

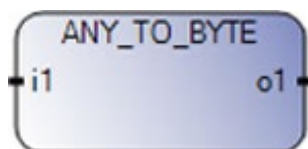
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_BYTE

Converte un valore in Byte a 8 bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Byte. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore non Byte.
o1	Uscita	BYTE	Un valore Byte a 8 bit.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_BYTE

(* Equivalenza ST: *)

```
bres := ANY_TO_BYTE (true);           (* bres è 1 *)
tres := ANY_TO_BYTE (t#0s46ms);      (* tres è 46 *)
mres := ANY_TO_BYTE ('0198');        (* mres è 198 *)
```

Vedere anche

[Istruzioni per la conversione dati](#) a pagina 263

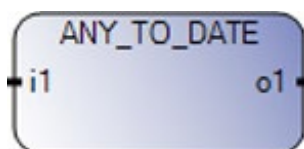
ANY_TO_DATE

Converte un tipo di dati String, Integer, Real o Time in un tipo di dati Date.

I tipi di dati Boolean, Byte e Word sono supportati ma non consigliati.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

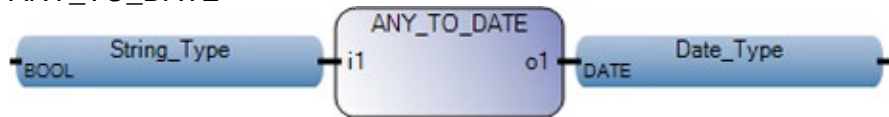


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

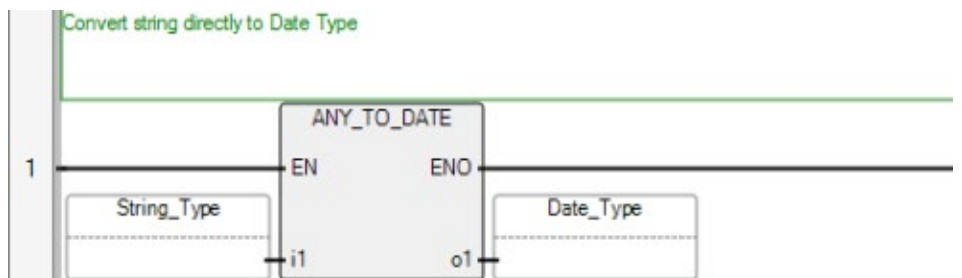
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Date. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME STRING	<ul style="list-style-type: none"> Le stringhe vengono convertite direttamente nel tipo di dati DATE e devono essere nel formato YYYY-MM-DD, DATE#YYYY-MM-DD o D#YYYY-MM-DD. I tipi Integer e Real, che devono essere positivi, sono in unità di secondi e vengono aggiunti alla data 1970-01-01. Time viene aggiunto alla data 1970-01-01.
o1	Uscita	DATE	Valore della data convertita. Gli errori durante la conversione, ad esempio una data negativa, generano un risultato NON VALIDO.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

ANY_TO_DATE esempio di programmazione a blocchi funzionali

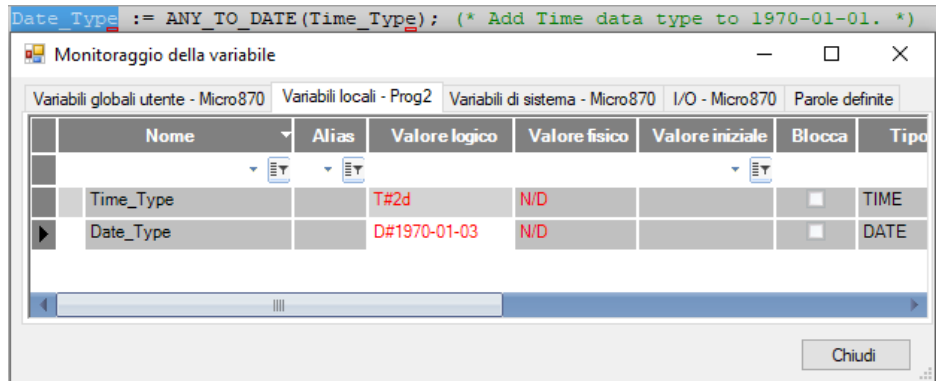
ANY_TO_DATE



Esempio di diagramma ladder ANY_TO_DATE



Esempio di testo strutturato ANY_TO_DATE



Vedere anche

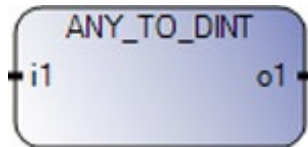
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_DINT

Converte un valore in Double Integer a 32 bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione a Double Integer a 32 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.

i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore diverso da Double Integer.
o1	Uscita	DINT	Un valore Double Integer a 32 bit.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_DINT

(* Equivalenza ST: *)

```

bres := ANY_TO_DINT (true);           (* bres è 1 *)
tres := ANY_TO_DINT (t#1s46ms);      (* tres è 1046 *)
mres := ANY_TO_DINT ('0198');        (* mres è 198 *)
  
```

Vedere anche

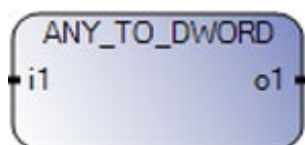
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_DWORD

Converte un valore in Double Word a 32 bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Double Word a 32 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore diverso da double Word.
o1	Uscita	DWORD	Un valore double Word a 32 bit.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_DWORD

(* Equivalenza ST: *)

```
bres := ANY_TO_DWORD (true);           (* bres è 1 *)
tres := ANY_TO_DWORD (t#1s46ms);      (* tres è 1046 *)
mres := ANY_TO_DWORD ('0198');        (* mres è 198 *)
```

Vedere anche

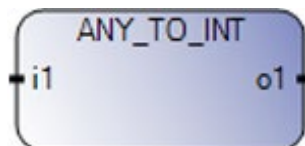
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_INT

Converte un valore in Integer a 16 bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Integer a 16 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
iI	Ingresso	BOOL SINT USINT BYTE UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore Integer non a 16 bit.
oI	Uscita	INT	Un valore Integer a 16 bit.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_INT

(* Equivalenza ST: *)

```

bres := ANY_TO_INT (true);           (* bres è 1 *)
tres := ANY_TO_INT (t#0s46ms);      (* tres è 46 *)
mres := ANY_TO_INT ('0198');        (* mres è 198 *)
    
```

Vedere anche

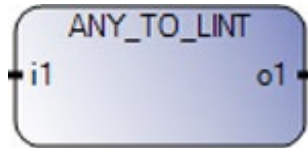
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_LINT

Converte un valore in Long Integer a 64 bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Long Integer a 64 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore diverso da Long Integer.
o1	Uscita	LINT	Un valore Long Integer a 64 bit.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_LINT

(* Equivalenza ST: *)

```

bres := ANY_TO_LINT (true);           (* bres è 1 *)
tres := ANY_TO_LINT (t#0s46ms);      (* tres è 46 *)
mres := ANY_TO_LINT ('0198');        (* mres è 198 *)
    
```

Vedere anche

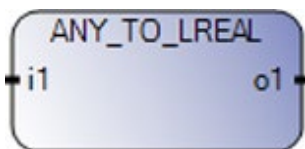
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_LREAL

Converte un valore in Long Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Long Real. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL TIME DATE STRING	Qualsiasi valore diverso da long Real.
o1	Uscita	LREAL	Un valore long Real.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_LREAL

(* Equivalenza ST: *)

```

bres := ANY_TO_LREAL (true);           (* bres è 1,0 *)
tres := ANY_TO_LREAL (t#1s46ms);      (* tres è 1046,0 *)
ares := ANY_TO_LREAL (198);           (* ares è 198,0 *)
    
```

Vedere anche

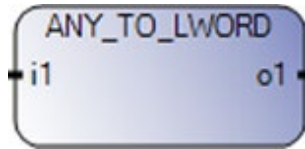
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_LWORD

Converte un valore in Long Word a 64 bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abitazione istruzioni. TRUE: esegue il calcolo della conversione in Long Word a 64 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT REAL LREAL IME DATE STRING	Qualsiasi valore diverso da Long Word.
o1	Uscita	LWORD	Un valore Long Word a 64 bit.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_LWORD

(* Equivalenza ST: *)

```
bres := ANY_TO_LWORD (true); (* bres è 1 *)
tres := ANY_TO_LWORD (t#0s46ms); (* tres è 46 *)
mres := ANY_TO_LWORD ('0198'); (* mres è 198 *)
```

Vedere anche

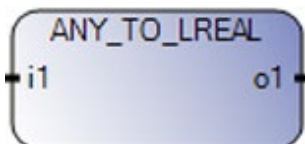
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_REAL

Converte un valore in Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Real. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD LREAL TIME DATE STRING	Qualsiasi valore diverso da Real.
o1	Uscita	REAL	Un valore Real.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_REAL

(* Equivalenza ST: *)

```

bres := ANY_TO_REAL (true);           (* bres è 1,0 *)
tres := ANY_TO_REAL (t#1s46ms);      (* tres è 1046,0 *)
ares := ANY_TO_REAL (198);           (* ares è 198,0 *)
    
```

Vedere anche

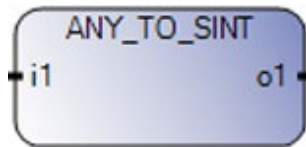
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_SINT

Converte un valore in Short Integer.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Short Integer a 8 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore non-Short Integer.
o1	Uscita	SINT	Qualsiasi valore Short Integer.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_SINT

(* Equivalenza ST: *)

```
bres := ANY_TO_SINT (true); (* bres è 1 *)
tres := ANY_TO_SINT (t#0s46ms); (* tres è 46 *)
mres := ANY_TO_SINT ('0198'); (* mres è 198 *)
```

Vedere anche

[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_STRING

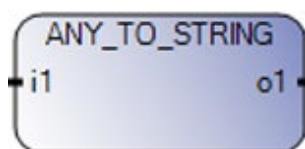
Converte un valore in String.

Dettagli operazione:

- Durante la conversione di un tipo di dati REAL in una stringa, l'istruzione ANY_TO_STRING utilizza il formato di IEEE 754.
 - ANY_TO_STRING converte da 125.0 a 1.25000E + 02
- Se la lunghezza della stringa di destinazione è di 5 caratteri:
 - ANY_TO_STRING converte da 125.0 a 1.25000E+02 e le uscite da 1.250 alla stringa di destinazione.
 - Considerare la creazione di un blocco funzione definito dall'utente per convertire dalla notazione esponenziale al numero.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in String. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE	Qualsiasi valore diverso da String.
o1	Uscita	STRING	Se IN è Boolean, "FALSE" o "TRUE". Se IN è Integer o Real, rappresentazione decimale. Se IN è TIME, può essere: TIME time1 STRING s1 time1 :=13 ms; s1:=ANY_TO_STRING(time1); (* s1 = '0s13' *).

ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
-----	--------	------	--

Esempio di testo strutturato ANY_TO_STRING

(* Equivalenza ST: *)

```
bres := ANY_TO_STRING (TRUE);           (* bres è 'TRUE' *)
ares := ANY_TO_STRING (125);           (* ares è '125' *)
```

Vedere anche

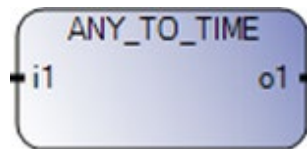
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_TIME

Converte un valore diverso da Time o Date in un valore Time.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Time. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL STRING	Qualsiasi valore positivo diverso da un tipo di dati Time o Date. IN (o parte di integer di IN se è real) è il numero di millisecondi. STRING (numero di millisecondi, ad esempio, un valore di 300032 rappresenta 5 minuti e 32 millisecondi).
o1	Uscita	TIME	Valore Time rappresentato da IN. Un valore di 1193h2m47s295ms non è valido.

ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
-----	--------	------	--

Esempio di testo strutturato ANY_TO_TIME

(* Equivalenza ST: *)

```
ares := ANY_TO_TIME (1256); (* ares := t#1s256ms *)
rres := ANY_TO_TIME (1256.3); (* rres := t#1s256ms *)
```

Vedere anche

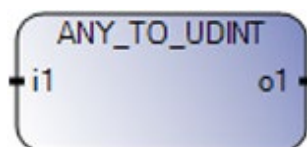
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_UDINT

Converte un valore in Unsigned Double Integer a 32 bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione a Double Integer a 32 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore diverso da Unsigned Double Integer.
o1	Uscita	UDINT	Un valore Unsigned Double Integer a 32 bit.

ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
-----	--------	------	--

Esempio di testo strutturato ANY_TO_UDINT

(* Equivalenza ST: *)

```
bres := ANY_TO_UDINT (true);           (* bres è 1 *)
tres := ANY_TO_UDINT (t#1s46ms);      (* tres è 1046 *)
mres := ANY_TO_UDINT ('0198');        (* mres è 198 *)
```

Vedere anche

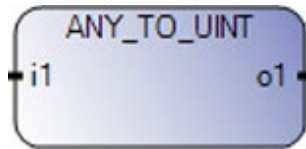
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_UINT

Converte un valore in Unsigned Integer.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Unsigned Integer a 16 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT USINT BYTE INT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore non-Unsigned Integer.

o1	Uscita	UINT	Un valore Unsigned Integer.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_UINT

(* Equivalenza ST: *)

```
bres := ANY_TO_UINT (true);           (* bres è 1 *)
tres := ANY_TO_UINT (t#0s46ms);      (* tres è 46 *)
mres := ANY_TO_UINT ('0198');        (* mres è 198 *)
```

Vedere anche

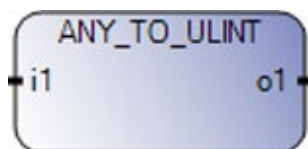
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_ULINT

Converte un valore in Unsigned Long Integer a 64 bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Unsigned Long Integer a 64 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.

i1	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT DWORD LINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore diverso da Unsigned Long Integer.
o1	Uscita	ULINT	Un valore Unsigned Long Integer a 64 bit.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_ULINT

(* Equivalenza ST: *)

```

bres := ANY_TO_ULINT (true);           (* bres è 1 *)
tres := ANY_TO_ULINT (t#0s46ms);      (* tres è 46 *)
mres := ANY_TO_ULINT ('0198');        (* mres è 198 *)
  
```

Vedere anche

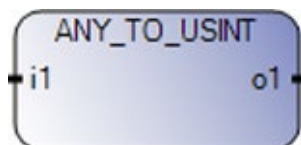
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_USINT

Converte un valore in Unsigned Short Integer.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Unsigned Short Integer a 8 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
i1	Ingresso	BOOL SINT BYTE INT UINT WORD DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore non-Short Integer.
o1	Uscita	USINT	Un valore Unsigned Short Integer.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_USINT

(* Equivalenza ST: *)

```

bres := ANY_TO_USINT (true);           (* bres è 1 *)
tres := ANY_TO_USINT (t#0s46ms);      (* tres è 46 *)
mres := ANY_TO_USINT ('0198');        (* mres è 198 *)
    
```

Vedere anche

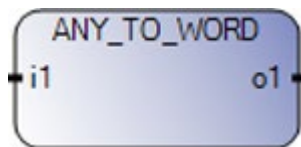
[Istruzioni per la conversione dati](#) a pagina 263

ANY_TO_WORD

Converte un valore in Word a 16 bit.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo della conversione in Word a 16 bit. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
il	Ingresso	BOOL SINT USINT BYTE INT DINT UDINT DWORD LINT ULINT LWORD REAL LREAL TIME DATE STRING	Qualsiasi valore diverso da un valore WORD.
o1	Uscita	WORD	Un valore Word.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di testo strutturato ANY_TO_WORD

(* Equivalenza ST: *)

bres := ANY_TO_WORD (true);	(* bres è 1 *)
tres := ANY_TO_WORD (t#0s46ms);	(* tres è 46 *)
mres := ANY_TO_WORD ('0198');	(* mres è 198 *)

Vedere anche

[Istruzioni per la conversione dati](#) a pagina 263

Istruzioni per l'elaborazione dei dati

Utilizzare le istruzioni per l'elaborazione dei dati per modificare i dati di uscita e lo stato senza modificare il programma.

Istruzione	Descrizione
AVERAGE a pagina 285	Calcola una media in esecuzione su un numero di campioni definito.
COP a pagina 287	Copia i dati binari nell'elemento di origine per l'elemento di destinazione.
MAX a pagina 294	Calcola il valore massimo tra due valori interi.
MIN a pagina 292	Calcola il valore minimo tra due valori interi.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

MEDIA

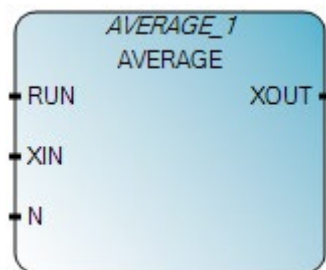
Calcola una media in movimento su un numero di campioni definito e memorizza il valore per ogni ciclo.

Dettagli operazione:

- Il numero definito di campioni (N) non deve superare 127.
- Quando si imposta o si cambia il valore di N, impostare RUN su FALSE, quindi nuovamente su TRUE.
- Se il comando RUN è FALSE (modalità reset), il valore di uscita è uguale al valore di ingresso.
- Al raggiungimento del numero massimo di valori archiviati, il primo valore archiviato viene cancellato dall'ultimo.
- usando i tipi di dati in virgola mobile, i calcoli potrebbero essere imprecisi a causa delle limitazioni nell'arrotondamento con le operazioni matematiche in virgola mobile.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

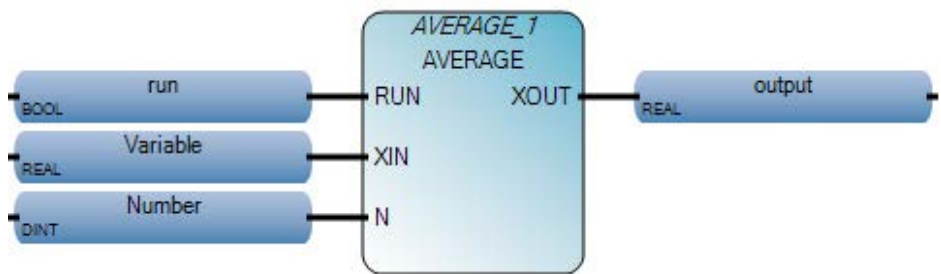
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



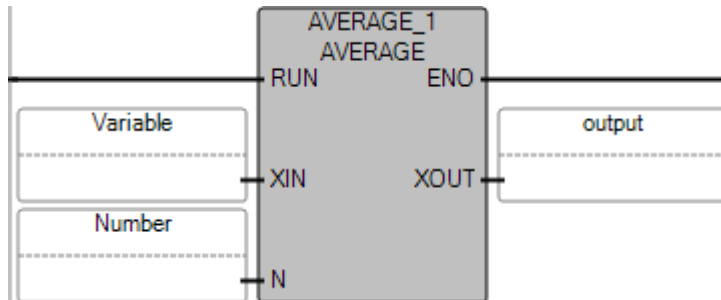
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
RUN	Ingresso	BOOL	TRUE = esecuzione FALSE = reset
XIN	Ingresso	REAL	Qualsiasi variabile reale.
N	Ingresso	DINT	Numero di campioni definito dall'applicazione.
XOUT	Uscita	REAL	Esecuzione della media dei valori XIN.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali AVERAGE



Esempio di diagramma ladder AVERAGE



Esempio di Testo strutturato AVERAGE

```
AVERAGE_1 {
void AVERAGE_1(BOOL RUN, REAL XIN, DINT N)
Tipo: AVERAGE, Media mobile su N campioni
```

```
1 | AVERAGE_1(run, Variable, Number);
2 | output := AVERAGE_1.XOUT;
```

(* Equivalenza ST: AVERAGE1 un'istanza di un blocco AVERAGE *)

```
AVERAGE1((auto_mode & store_cmd), sensor_value, 100);
ave_value := AVERAGE1.XOUT;
```

Vedere anche

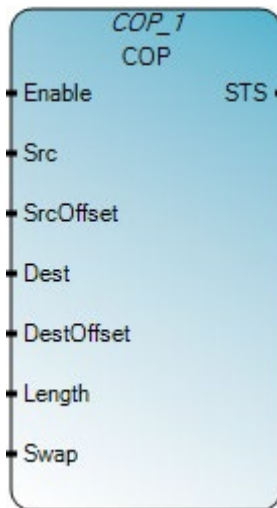
[Istruzioni per l'elaborazione dei dati](#) a pagina 285

COP

Copia i dati binari nell'elemento di origine per l'elemento di destinazione. L'elemento di origine rimane invariato.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione																		
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. COP è attivato in base al livello. TRUE: esegue la copia. FALSE: il blocco funzione non è attivo.																		
Src	Ingresso	<table border="0"> <tr> <td>BOOL</td> <td>DWORD</td> </tr> <tr> <td>SINT</td> <td>REAL</td> </tr> <tr> <td>USINT</td> <td>TIME</td> </tr> <tr> <td>BYTE</td> <td>DATE</td> </tr> <tr> <td>INT</td> <td>STRING</td> </tr> <tr> <td>UINT</td> <td>LWORD</td> </tr> <tr> <td>WORD</td> <td>ULINT</td> </tr> <tr> <td>DINT</td> <td>LINT</td> </tr> <tr> <td>UDINT</td> <td>LREAL</td> </tr> </table>	BOOL	DWORD	SINT	REAL	USINT	TIME	BYTE	DATE	INT	STRING	UINT	LWORD	WORD	ULINT	DINT	LINT	UDINT	LREAL	Elemento iniziale da copiare. Se l'origine è un tipo di dati STRINGA, la destinazione deve essere un tipo di dati STRINGA o un USINT (UCHAR e BYTE). In caso contrario, viene segnalata una mancata corrispondenza tra tipi di dati.
BOOL	DWORD																				
SINT	REAL																				
USINT	TIME																				
BYTE	DATE																				
INT	STRING																				
UINT	LWORD																				
WORD	ULINT																				
DINT	LINT																				
UDINT	LREAL																				
SrcOffset	Ingresso	UINT	L'offset elemento di origine è utilizzato con i tipi di dati del vettore per identificare la posizione nel vettore di origine da cui copiare i dati. Impostare l'offset su 0: <ul style="list-style-type: none"> • Se non è un tipo di dati array oppure • Per copiare dal primo elemento per un tipo di dati array. 																		

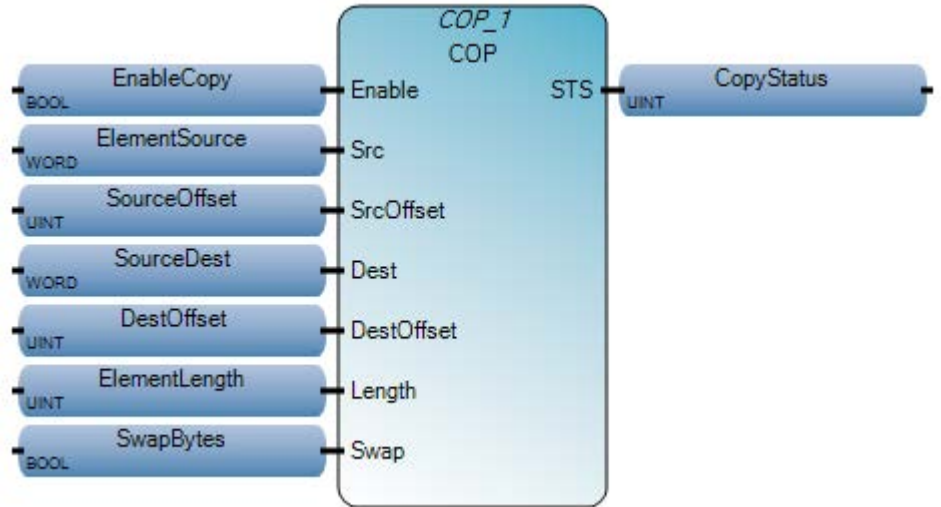
Dest	Ingresso	BOOL SINT USINT BYTE INT UINT WORD DINT UDINT	DWORD REAL TIME DATE STRING LWORD ULINT LINT	Elemento iniziale da sovrascrivere dall'origine. Se la destinazione è un tipo di dati STRINGA, l'origine deve essere un tipo di dati STRINGA o un USINT (UCHAR e BYTE). In caso contrario, viene segnalata una mancata corrispondenza tra tipi di dati.
DestOffset	Ingresso	UINT		L'offset elemento di destinazione è utilizzato con i tipi di dati del vettore per identificare la posizione nel vettore di destinazione in cui copiare i dati. Impostare l'offset su 0: <ul style="list-style-type: none"> • Se non è un tipo di dati array oppure • Per copiare dal primo elemento per un tipo di dati array.
Lunghezza	Ingresso	UINT		Numero di elementi di destinazione da copiare. Quando la destinazione è un tipo di dati STRINGA, indica il numero di stringhe da copiare.
Swap	Ingresso	BOOL		Utilizzato per scambiare i dati dagli elementi di origine e destinazione, in modo che i dati di destinazione sostituiscono i dati di origine e i dati di origine sostituiscono i dati di destinazione. TRUE: scambio dei byte in base al tipo di dati. Un'operazione di scambio non avviene se: <ul style="list-style-type: none"> • Il tipo di dati di origine o di destinazione è una STRINGA oppure • Sia i dati di origine che quelli di destinazione hanno lunghezza 1 byte.
Sts	Uscita	UINT		Stato dell'operazione di copia. Le definizioni del parametro Sts sono definite nei codici di stato COP.
ENO	Uscita	BOOL		Abilita l'uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Codici di stato (Sts) del COP

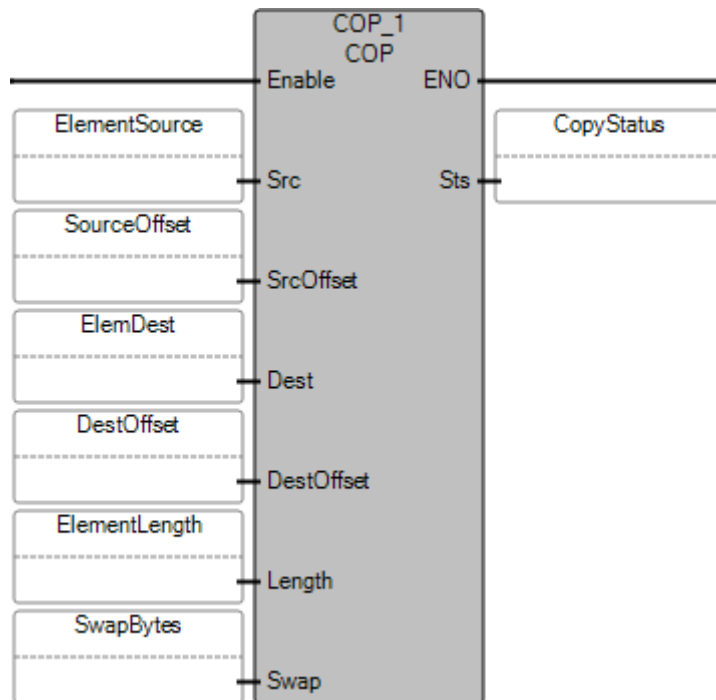
La tabella seguente descrive i codici di stato del COP.

Codice dello stato COP	Descrizione stato
0x00	Nessuna azione intrapresa (non abilitato).
0x01	Blocco funzione COP con successo.
0x02	La destinazione presenta byte residui durante la copia dalla stringa.
0x03	I dati di origine sono troncati.
0x04	La lunghezza di copia non è valida.
0x05	Il tipo di dati non corrisponde con il tipo di dati stringa come origine o destinazione.
0x06	La dimensione dei dati di origine è troppo ridotta per la copia.
0x07	La dimensione dei dati di destinazione è troppo ridotta per la copia.
0x08	Offset dei dati di origine non valido.
0x09	Offset dei dati di destinazione non valido.
0x0A	I dati non sono validi o nell'origine o nella destinazione.

Esempio di programmazione a blocchi funzionali COP



Esempio di diagramma ladder COP



Esempio di testo strutturato COP

```

1 COP_1(EnableCopy, ElementSource, SourceOffset, SourceDest,
2   DestOffset, ElementLength, SwapBytes);
3 output :=COP_1.STS;

```

Vedere anche

[Esempio di array della stringa COP](#) a pagina 290

[Istruzioni per l'elaborazione dei dati](#) a pagina 285

[Copia in un tipo di dati differente](#) a pagina 290

Copia in un tipo di dati differente

Quando viene eseguita una copia da o in un tipo di dati stringa, per i dati nell'array USINT è utilizzato il formato di stringa breve ODVA.

Quando la COP è utilizzata tra qualsiasi altra coppia di tipi di dati, l'operazione di copia è valida, anche se il tipo di dati nell'origine non è dello stesso tipo di quello nella destinazione e anche quando il formato di entrambi non è valido. La logica deve essere convalidata a livello di applicazione.

Da un array USINT a un array stringa

Per copiare un array USINT a un array stringa, i dati nell'array USINT devono essere in questo formato:

- Byte1: lunghezza della prima stringa
- Byte2: primo carattere del byte
- Byte3: secondo carattere del byte
- Byte n: ultimo carattere del byte
- Byte (n+1): lunghezza della seconda stringa
- Byte (n+2): primo carattere del byte della seconda stringa

Vedere anche

[COP](#) a pagina 287

Esempio di array della stringa COP

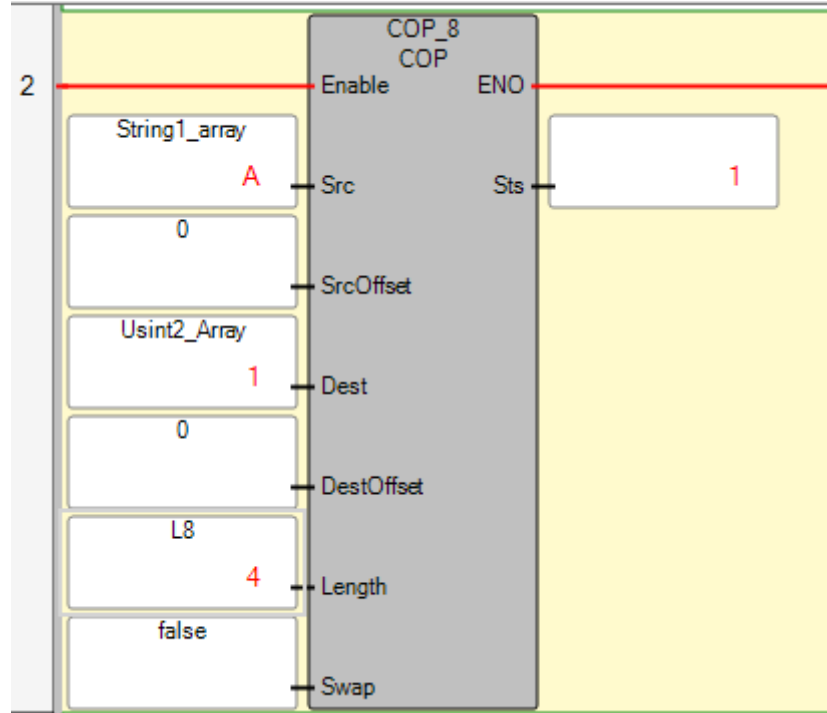
L'esempio che segue mostra un'istruzione COP che copia un vettore stringa in un vettore Usint. L'istruzione COP ignora tutti gli elementi dell'array sorgente di lunghezza pari a zero.

In questo esempio:

- La lunghezza specificata è pari a 4 byte.
- Il numero degli elementi di destinazione da copiare è 4 byte.
- Tutti gli elementi dell'array di lunghezza pari a zero (vuoti) vengono ignorati.
- L'istruzione COP trova un elemento di lunghezza diversa da zero nell'array stringa[1]; questo viene copiato nell'array USINT designato[1] con 1. Uno corrisponde alla lunghezza della stringa nell'array stringa[1] e l'array USINT[2] è 65, ossia il codice ASCII corrispondente ad "A".

- L'istruzione COP trova un elemento di lunghezza diversa da zero nell'array stringa[10] che corrisponde ad "a"; questo viene copiato nell'array USINT di destinazione[3] con 1. Uno corrisponde alla lunghezza della stringa nell'array stringa[10] e l'array USINT[4] è 97, ossia il codice ASCII corrispondente ad "a".

Ladder diagram



Elementi dell'array con valori logici

	Nome	Alias	Valore logico
-	String_String_1.String1_array		...
	String_String_1.String1_array[1]		A
	String_String_1.String1_array[2]		
	String_String_1.String1_array[3]		
	String_String_1.String1_array[4]		
	String_String_1.String1_array[5]		
	String_String_1.String1_array[6]		
	String_String_1.String1_array[7]		
	String_String_1.String1_array[8]		
	String_String_1.String1_array[9]		
	String_String_1.String1_array[10]		a
-	String_String_1.USint2_Array		...
	String_String_1.USint2_Array[1]		1
	String_String_1.USint2_Array[2]		65
	String_String_1.USint2_Array[3]		1
▶	String_String_1.USint2_Array[4]		97
	String_String_1.USint2_Array[5]		0
	String_String_1.USint2_Array[6]		0
	String_String_1.USint2_Array[7]		0
	String_String_1.USint2_Array[8]		0
	String_String_1.USint2_Array[9]		0
	String_String_1.USint2_Array[10]		0
	String_String_1.USint2_Array[11]		0
	String_String_1.USint2_Array[12]		0
	String_String_1.USint2_Array[13]		0
	String_String_1.L8		4
+	String_String_1.COP_8		...

Vedere anche

[COP](#) a pagina 287

MIN (minimo)

Calcola il valore minimo tra due valori interi.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

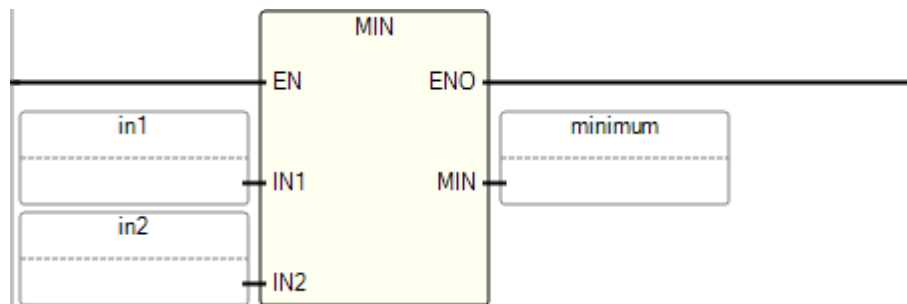
Parametro	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo del valore intero minimo. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN1	Ingresso	DINT	Qualunque valore intero con segno.
IN2	Ingresso	DINT	Non può essere reale.
MIN	Uscita	DINT	Il minimo dei due valori di ingresso.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali MIN



Esempio di Diagramma ladder MIN



Esempio di Testo strutturato MIN

```
MIN(
  DINT MIN(DINT IN1, DINT IN2)
  Minimo
```

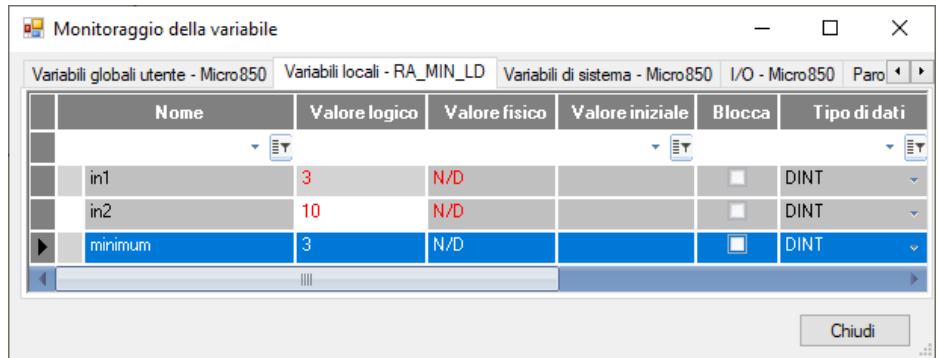
```
1 in1 := 3;
2 in2 := 10;
3 minimum := MIN(in1, in2);
```

(* Equivalenza ST: *)

nuovo_valore := MAX (MIN (valore_max, valore), valore_min);

(* vincola il valore al gruppo [valore_min..valore_max] *)

Risultati



Vedere anche

[Istruzioni per l'elaborazione dei dati](#) a pagina 285

MAX (massimo)

Calcola il valore massimo tra due valori interi.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



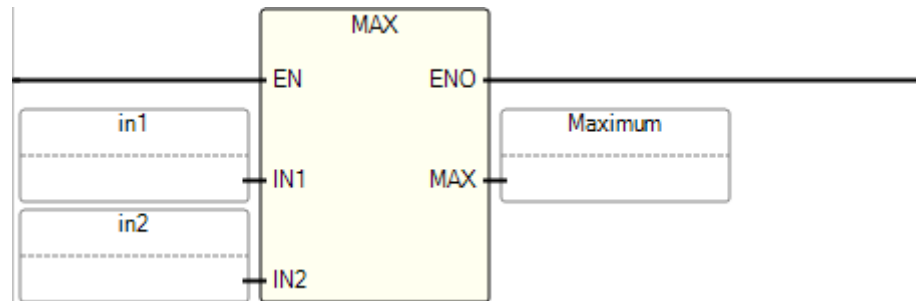
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo del valore intero massimo. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
IN1	Ingresso	DINT	Qualunque valore intero con segno.
IN2	Ingresso	DINT	Non può essere reale.
MAX	Uscita	DINT	Il massimo dei due valori di ingresso.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali MAX



Esempio di Diagramma ladder MAX



Esempio di Testo strutturato MAX

```

MAX {
  DINT MAX(DINT IN1, DINT IN2)
  Massimo
}

1 IN1 := 3;
2 IN2 := 10;
3 Maximum := MAX(IN1, IN2);

```

(* Equivalenza ST: *)

nuovo_valore := MAX (MIN (valore_max, valore), valore_min);

(* vincola il valore al gruppo [valore_min..valore_max] *)

Risultati

Monitoraggio della variabile						
Variabili globali utente - Micro850						
Variabili locali - RA_MAX_LD						
Variabili di sistema - Micro850						
I/O - Micro850						
Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	
IN1	3	N/D		<input type="checkbox"/>	DINT	
IN2	10	N/D		<input type="checkbox"/>	DINT	
Maximum	10	N/D		<input checked="" type="checkbox"/>	DINT	

Vedere anche

[Istruzioni per l'elaborazione dei dati](#) a pagina 285

Istruzioni per l'High Speed Counter (HSC)

Monitorare e controllare l'High Speed Counter attenendosi alle relative istruzioni.

Istruzione	Descrizione
HSC a pagina 298	HSC applica all'high-speed counter valori alti e bassi preimpostati e valori di origine per l'uscita.
HSC_SET_STS a pagina 314	HSC_SET_STS permette l'impostazione o il ripristino manuale dello stato del conteggio dell'HSC.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

Che cos'è l'High-Speed Counter?

Un high-speed counter rileva e conta impulsi stretti (veloci), quindi invia istruzioni specifiche per avviare le operazioni di controllo, quando i conteggi rilevati raggiungono i relativi valori preimpostati. Tra le operazioni di controllo sono inclusi l'esecuzione automatica e immediata della routine di interruzione dell'high-speed counter e l'aggiornamento immediato delle uscite in base al modello di maschera e all'origine configurata.

Funzionalità dell'high-speed counter

Poiché le istruzioni dell'HSC richiedono elevate prestazioni, il loro funzionamento è garantito da una circuiteria personalizzata che opera parallelamente al processore di sistema principale. Tra le funzionalità avanzate degli High-Speed Counter (HSC) sono incluse:

- Funzionamento a 100 kHz del controllo diretto dell'alta velocità per le uscite
- Dati in formato numero intero con segno a 32 bit (intervallo di conteggio $\pm 2.147.483.647$)
- Valori alto e basso preimpostati e programmabili
- SetPoint di overflow e underflow
- Elaborazione delle interruzioni automatiche in base al conteggio accumulato
- Operatività delle istruzioni dell'HSC con parametri di runtime modificabili (dal programma di controllo dell'utente)

Supporto del controllore Micro800 per l'HSC

Tutti i controllori Micro830, Micro850 e Micro870, tranne 2080-LCxx-AWB, supportano fino a sei ingressi HSC. La funzionalità HSC è implementata nei controllori Micro800 tramite l'hardware dell'high-speed counter (ingressi integrati nel controllore) e l'istruzione HSC nell'applicazione. L'istruzione HSC configura l'hardware dell'high-speed counter e aggiorna l'accumulatore dell'immagine.

IMPORTANTE è possibile utilizzare la funzione HSC solo con l'I/O integrato del controllore. Non può essere utilizzata con moduli di espansione I/O.

Vedere anche

[Istruzioni per l'High-Speed Counter \(HSC\)](#) a pagina 297

[Utilizzo delle istruzioni dell'High-Speed Counter](#) a pagina 317

HSC (High Speed Counter)

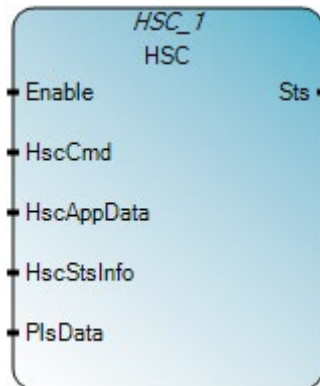
HSC applica all'high-speed counter valori alti e bassi preimpostati e valori di origine per l'uscita.

Dettagli operazione:

- L'interruttore di fine corsa programmabile (PLS) viene abilitato impostando il parametro HSCAppData.PLSEnable su True.
- Il parametro PLSPosition viene reimpostato al termine di un ciclo completo e al raggiungimento del valore HSCSTS.HP.

Questa istruzione si applica ai controllori Micro830 e Micro850.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

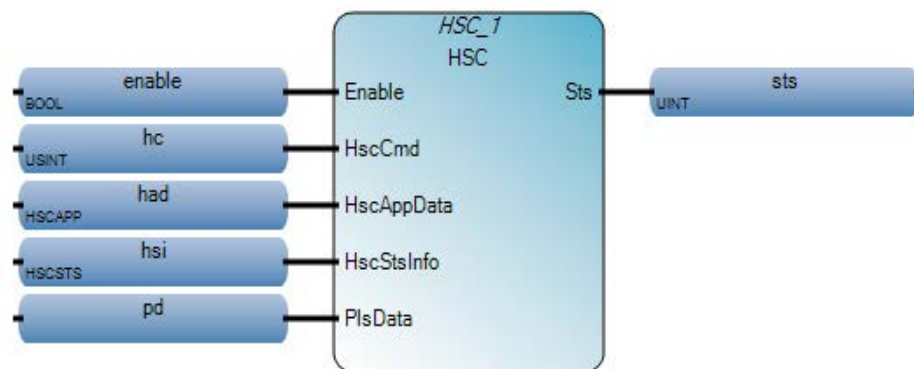


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

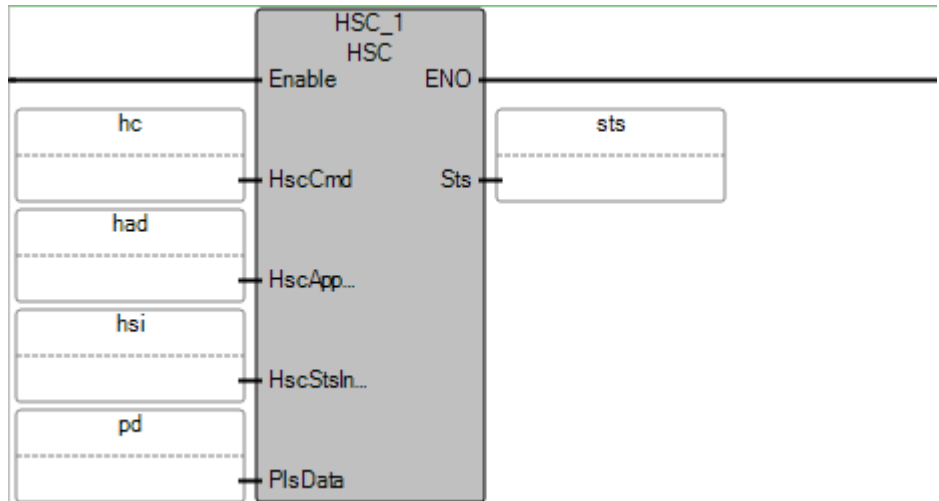
Parametro	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

EN	Ingresso	BOOL	Stato del ramo del blocco istruzione. TRUE: il timer inizia a incrementare. FALSE: il blocco funzione non è attivo. Si consiglia di non usare il parametro EN con il blocco funzione HSC perché il timer continua a funzionare anche quando EN è impostato su FALSE. Applicabile solo alle programmazioni in Ladder Diagram.
Enable	Ingresso	BOOL	Abilitare il blocco istruzione. TRUE: esegue l'operazione HSC specificata nel parametro di comando HSC. FALSE: non viene emesso nessun comando HSC.
HscCmd	Ingresso	USINT	Invia comandi all'HSC.
HSCAppData	Ingresso	HSCAPP	Configurazione dell'applicazione HSC; di solito necessaria solo una volta.
HSCStsInfo	Ingresso	HSCSTS	Stato dinamico dell'HSC. Di norma, è aggiornato continuamente durante il conteggio HSC.
PlsData	Ingresso	DINT UDINT	Struttura dati PLS (interruttore fine corsa programmabile).
Sts	Uscita	UINT	Stato dell'esecuzione dell'HSC. Codici di stato dell'HSC: <ul style="list-style-type: none"> • 0x00: nessuna azione intrapresa (non abilitato). • 0x01: esecuzione HSC completata. • 0x02: comando HSC non valido. • 0x03: ID HSC fuori intervallo. • 0x04: errore di configurazione HSC.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali HSC



Esempio di diagramma ladder HSC



Esempio di testo strutturato HSC

```
HSC_1(
void HSC_1(BOOL Enable, USINT HscCmd, HSCAPP HscAppData, HSCSTS HscStsInfo, PLS[1..1] PlsData)
Tipo: HSC, Applicare preimpostazioni alte/basse e origine di uscita all'HSC.

1 | HSC_1(enable, hc, had, hsi, pd);
2 | sts := HSC_1.Sts;
```

Vedere anche

[Istruzioni per l'High-Speed Counter \(HSC\)](#) a pagina 297

[Valori HSCCmd](#) a pagina 300

[Tipo di dati HSCAPP](#) a pagina 301

[Tipo di dati HSCSTS](#) a pagina 304

[Tipo di dati PLS](#) a pagina 311

Valori HSCCmd

Nella seguente tabella sono descritti i comandi dell'HSC per ciascuno dei valori di comando.

Comando HSC	Descrizione comando
0x01	HSC RUN <ul style="list-style-type: none"> • Avvia l'HSC (se HSC è in modalità inattiva ed è abilitato il piolo). • Aggiorna solo le informazioni sullo stato dell'HSC (se HSC è in modalità di esecuzione ed è abilitato il piolo).
0x02	HSC Stop arresta un conteggio HSC (se HSC è in modalità di esecuzione ed è abilitato il piolo).
0x03	HSC Load/Set: ricarica i dati dell'applicazione HSC (se è abilitato il piolo) per 6 elementi di input: HPSetting, LPSetting, HPOutput, LPOutput, OFSetting, UFSettingand e UFSetting. Nota: questo comando non ricarica i seguenti elementi di input: accumulatore HSC.
0x04	Ripristino accumulatore HSC (se il rung è abilitato).

Risultati del comando HSC

Valore comando	Risultato	Condizioni
HscCmd =1	Avvia il meccanismo dell'HSC e fa passare le transizioni dell'HSC alla modalità di esecuzione.	L'impostazione del parametro di ingresso Enable su False non arresta il conteggio in modalità di esecuzione. HscCmd =2 deve essere emesso per arrestare il conteggio.
	Il meccanismo dell'HSC aggiorna automaticamente i valori.	HSC AppData.Accumulator è aggiornato con HSC Sts.Accumulator
HscCmd =4 (ripristino)	Imposta il valore Acc. dell'HSC al valore AppData.Accumulator dell'HSC.	HscCmd =4 non arresta il conteggio HSC. Se HSC è in fase di conteggio quando viene emesso HscCmd =4, è possibile che alcuni conteggi vadano persi. Per impostare un valore Acc dell'HSC specifico durante il conteggio, scrivere il valore in AppData.Accumulator dell'HSC immediatamente prima dell'emissione di HscCmd =4.

Vedere anche

[HSC](#) a pagina 298

Tipo di dati HSCAPP

Utilizzare il tipo di dati HSCAPP per definire il parametro HSCAppData nell'istruzione dell'HSC. I parametri del tipo di dati HSCApp servono a definire i dati di configurazione dell'HSC.

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati HSCAPP.

Parametri	Tipo di dati	Formato dati	Accesso al programma utente	Descrizione
PLSEnable	BOOL	bit	lettura/scrittura	Abilitare o disabilitare l'interruttore di fine corsa programmabile (PLS) dell'High Speed Counter.
HSCID	UINT	word	lettura/scrittura	Definisce l'HSC.
HSCMode	UINT	word	lettura/scrittura	Definisce la modalità dell'HSC.
Accumulator	DINT	parola lunga	lettura/scrittura	Valore accumulatore iniziale. HSCApp.Accumulator imposta il valore iniziale dell'accumulatore quando si attiva l'High Speed Counter. Quando l'High-Speed Counter è in modalità conteggio, l'accumulatore viene automaticamente aggiornato dal sottosistema dell'HSC, per propagare il valore effettivo dell'accumulatore dell'HSC.
HPSetting	DINT	parola lunga	lettura/scrittura	Impostazione valore alto preimpostato. Il parametro HSCApp.HPSetting imposta il setpoint superiore (nei conteggi) che definisce quando il sottosistema dell'HSC genera un interrupt. Il numero dei dati caricati nell'impostazione valore alto preimpostato deve essere inferiore o uguale ai dati residenti nel parametro di overflow (HSCAPP.OFSetting). In caso contrario, viene generato un errore di HSC.
LPSSetting	DINT	parola lunga	lettura/scrittura	Impostazione valore basso preimpostato. HSCApp.LPSSetting imposta il setpoint inferiore (nei conteggi) che definisce quando il sottosistema dell'HSC genera un interrupt. Il numero dei dati caricati nell'impostazione valore basso preimpostato deve essere superiore o uguale ai dati residenti nel parametro di underflow (HSCAPP.UFSetting). In caso contrario, viene generato un errore di HSC. Se i valori di underflow e dell'impostazione valore basso preimpostato sono numeri negativi, il valore basso preimpostato deve essere un numero con un valore assoluto inferiore al valore di underflow.

OFSetting	DINT	parola lunga	lettura/scrittura	<p>Impostazione overflow.</p> <p>L'impostazione di overflow di HSCApp.OFSetting definisce il limite di conteggio superiore del contatore.</p> <p>Se il valore accumulato del contatore sale oltre il valore specificato in OFSetting, viene generata un'interruzione per overflow.</p> <p>Quando viene generata l'interruzione per overflow, il sottosistema dell'HSC reimposta il valore accumulato sul valore di underflow e il contatore continua a contare dal valore di underflow (i conteggi non vanno persi durante questa transizione).</p> <p>I valori di OFSetting devono essere:</p> <ul style="list-style-type: none"> • Tra -2.147.483.648 e 2.147.483.647. • Maggiori del valore di underflow. • Maggiori o uguali ai dati residenti nell'impostazione valore alto preimpostato (HSCAPP.HPSetting). In caso contrario, viene generato un errore di HSC.
UFSetting	DINT	parola lunga	lettura/scrittura	<p>Impostazione underflow.</p> <p>Impostazione di underflow di HSCApp.UFSetting che definisce il limite di conteggio inferiore del contatore.</p> <p>Se il valore accumulato del contatore scende sotto il valore specificato in UFSetting, viene generata un'interruzione per underflow.</p> <p>Quando viene generata l'interruzione per underflow, il sottosistema dell'HSC reimposta il valore accumulato sul valore di overflow e il contatore avvia il conteggio dal valore di overflow (i conteggi non sono persi nella transizione).</p> <p>I valori di UFSetting devono essere:</p> <ul style="list-style-type: none"> • Tra -2.147.483.648 e 2.147.483.647. • Inferiori al valore di overflow. • Inferiori o uguali ai dati residenti nell'impostazione valore basso preimpostato (HSCAPP.LPSetting), altrimenti viene generato un errore di HSC.
OutputMask	UDINT	word	lettura/scrittura	<p>Maschera esterna per uscita.</p> <p>HSCApp.OutputMask definisce le uscite integrate nel controllore, direttamente controllabili dall'High Speed Counter. Il sottosistema HSC è in grado di impostare le uscite su ON oppure OFF senza l'interazione del programma di controllo in base ai valori preimpostati alto o basso dell'accumulatore dell'HSC.</p> <p>Il modello di bit memorizzato in HSCApp.OutputMask definisce quali uscite vengono controllate dall'HSC e quali no.</p> <p>Il modello di bit HSCAPP.OutputMask corrisponde ai bit di uscita del controllore ed è possibile configurarlo durante l'installazione iniziale.</p> <p>I bit impostati (1) sono abilitati ed possibile impostarli su On oppure Off tramite il sottosistema dell'HSC.</p> <p>I bit cancellati (0) non possono essere impostati su On oppure Off tramite sottosistema dell'HSC.</p> <p>Ad esempio, per utilizzare l'HSC per controllare le uscite 0, 1, 3, assegnare:</p> <ul style="list-style-type: none"> • HscAppData.OutputMask = 2#1011 oppure • HscAppData.OutputMask = 11
HPOutput	UDINT	parola lunga	lettura/scrittura	<p>Impostazione uscita 32 bit per raggiungimento valore alto preimpostato.</p> <p>HSCApp.HPOutput definisce lo stato (1 = ON oppure 0 = OFF) delle uscite del controllore quando viene raggiunto il valore alto preimpostato. Per maggiori informazioni su come impostare direttamente su On oppure Off in base al valore alto preimpostato.</p> <p>Configurare il modello di bit uscita alta durante l'impostazione iniziale o utilizzare il blocco funzione HSC per caricare i nuovi parametri durante il funzionamento del controllore.</p>
LPOutput	UDINT	parola lunga	lettura/scrittura	<p>Impostazione uscita 32 bit per raggiungimento valore basso preimpostato.</p> <p>HSCApp.LPOutput definisce lo stato (1 = "on" oppure 0 = "off") delle uscite del controllore quando viene raggiunto il valore basso preimpostato. Per maggiori informazioni su come impostare direttamente le uscite su ON oppure OFF in base al valore basso preimpostato.</p> <p>Configurare il modello di bit uscita bassa durante l'impostazione iniziale o utilizzare il blocco funzione HSC per caricare i nuovi parametri durante il funzionamento del controllore.</p>

Confronto tra impostazioni HSCApp e impostazioni PLSData

Quando la funzione PLS è abilitata, le impostazioni HSCApp pertinenti sono sostituite dalle impostazioni PLSData corrispondenti, come descritto nella seguente tabella.

Impostazione HSCApp	Impostazione PLSData
HSCAPP.HpSetting	HSCHP
HSCAPP.LpSetting	HSCLP
HSCAPP.HPOutput	HSCHPOOutput
HSCAPP.LPOutput	HSCLPOutput

HSCApp.HSCID

Il parametro HSCApp.HSCID identifica l'High Speed Counter.

Nella tabella seguente sono elencati i valori per l'HSCID:

Selezione uscita	Bit	Descrizione
Prima parola dati funzione HSC	15-13	Tipo di modulo HSC: <ul style="list-style-type: none"> • 0x00 - Integrato. • 0x01 - Espansione. • 0x02 - Porta plug-in.
	12-8	ID slot del modulo: <ul style="list-style-type: none"> • 0x00 - Integrato. • 0x01-0x1F - ID del modulo di espansione. • 0x01-0x05 - ID della porta plug-in.
	7-0	ID HSC all'interno del modulo: <ul style="list-style-type: none"> • 0x00-0x0F - Integrato. • 0x00-0x07 - ID di HSC per Espansione. • 0x00-0x07 - ID di HSC per porta plug-in. per la versione iniziale di Connected Components Workbench sono supportati solo ID 0x00-0x05.

HSCApp.HSCMode

Il parametro HSCApp.HSCMode imposta l'High Speed Counter su uno dei 10 tipi di modalità di conteggio. Il valore della modalità è configurato tramite il dispositivo di programmazione ed è accessibile nel programma di controllo.

Per ulteriori informazioni sulle modalità operative HSC e le assegnazioni degli input, vedere gli schemi dei cablaggi e degli input HSC nel *Manuale dell'utente per i controllori programmabili Micro830 e Micro850*.

Modalità operativa dell'HSC, HSC principale e quello secondario supportano modalità differenti.

- Gli High Speed Counter principali supportano 10 tipi di modalità operative.
- Gli High Speed Counter secondari supportano 5 tipi di modalità operative (modalità 0, 2, 4, 6, 8).

- Se l'High Speed Counter principale è impostato in modalità 1, 3, 5, 7 o 9, l'High Speed Counter secondario verrà disabilitato.

HSCMode	Modalità conteggio
0	Contatore su. L'accumulatore è cancellato (0) immediatamente al raggiungimento del valore alto preimpostato. Non è possibile definire un valore basso preimpostato in questa modalità.
1	Contatore su con ripristino esterno e tenuta. L'accumulatore è cancellato (0) immediatamente al raggiungimento del valore alto preimpostato. Non è possibile definire un valore basso preimpostato in questa modalità.
2	Contatore con direzione esterna.
3	Contatore con direzione esterna, ripristino e tenuta.
4	Contatore con due ingressi (su e giù).
5	Contatore con due ingressi (su e giù), ripristino esterno e tenuta.
6	Contatore in quadratura (ingressi in fase A e B).
7	Contatore in quadratura (ingressi in fase A e B) con ripristino esterno e tenuta.
8	Contatore in quadratura X4 (ingressi in fase A e B).
9	Contatore in quadratura X4 (ingressi in fase A e B) con ripristino esterno e tenuta.

Esempio di parametri HSCAppData

Nella seguente immagine sono rappresentati i parametri HSCAppData nel **Selettore di variabili**.

Nome	Alias	Tipo di dati	Dimensione	Valore progett.	Valore iniziale
HSC_1		HSC	
HSC_1.Enable		BOOL			
HSC_1.HscCmd		USINT			
HSC_1.HscAppData		HSCAPP	
HSC_1.HscAppData.PlsEnable		BOOL			
HSC_1.HscAppData.HscID		UINT			
HSC_1.HscAppData.HscMode		UINT			
HSC_1.HscAppData.Accumulator		DINT			
HSC_1.HscAppData.HPSSetting		DINT			
HSC_1.HscAppData.LPSetting		DINT			
HSC_1.HscAppData.OFSetting		DINT			
HSC_1.HscAppData.UFSetting		DINT			
HSC_1.HscAppData.OutputMask		UDINT			
HSC_1.HscAppData.HPOutput		UDINT			
HSC_1.HscAppData.LPOutput		UDINT			

Vedere anche

[HSC](#) a pagina 298

Tipo di dati HSCSTS

HSCSTSInfo (tipo di dati HSCSTS) visualizza lo stato dell'High-Speed Counter.

Azioni dello stato HSCSTSInfo

Durante il conteggio dell'HSC si verificano le seguenti azioni di stato HSC.

- Se il blocco di funzione dell'HSC sta eseguendo il conteggio con il comando 0x01, lo stato dell'HSC viene aggiornato costantemente.
- Se si verifica un errore, il flag Error_Detected viene attivato e viene impostato un codice di errore.

Parametri HSCSTSInfo

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente	Descrizione
CountEnable	BOOL	0...9	sola lettura	Conteggio abilitato.
ErrorDetected	BOOL	0...9	lettura/scrittura	Diverso da zero indica un errore rilevato.
CountUpFlag	BOOL	0...9	sola lettura	Flag conteggio crescente.
CountDwnFlag	BOOL	2...9	sola lettura	Flag conteggio decrescente.
ModeIDone	BOOL	0 oppure 1	lettura/scrittura	HSC è Mode 1A o Mode 1B; l'accumulatore conta fino al valore HP.
OVF	BOOL	0...9	lettura/scrittura	Rilevato un overflow.
UNF	BOOL	0...9	lettura/scrittura	Rilevato un underflow.
CountDir	BOOL	0...9	sola lettura	1: conteggio crescente; 0: conteggio decrescente.
HPReached	BOOL	2...9	lettura/scrittura	Raggiunto valore alto preimpostato.
LPReached	BOOL	2...9	lettura/scrittura	Raggiunto valore basso preimpostato.
OFCauseInter	BOOL	0...9	lettura/scrittura	Overflow causato da interruzioni dell'HSC.
UFCauseInter	BOOL	2...9	lettura/scrittura	Underflow causato da interruzioni dell'HSC.
HPCauseInter	BOOL	0...9	lettura/scrittura	Raggiunto valore alto preimpostato che ha causato un'interruzione nell'HSC.
LPCauseInter	BOOL	2...9	lettura/scrittura	Raggiunto valore basso preimpostato che ha causato un'interruzione nell'HSC.
PLsPosition	UINT	0...9	sola lettura	Posizione del interruttore di fine corsa programmabile. Il parametro PLsPosition viene reimpostato al termine di un ciclo completo e al raggiungimento del valore HP.
ErrorCode	UINT	0...9	lettura/scrittura	Visualizza i codici di errore rilevati dal sottosistema dell'HSC.
Accumulator	DINT		lettura/scrittura	Letture attuale accumulatore.
HP	DINT		sola lettura	Impostazione valore alto preimpostato più recente.
LP	DINT		sola lettura	Impostazione valore basso preimpostato più recente.
HPOutput	UDINT		lettura/scrittura	Impostazione uscita valore alto preimpostato più recente.
LPOutput	UDINT		lettura/scrittura	Impostazione uscita valore basso preimpostato più recente.

Dettagli dei parametri HSCSTSInfo

I parametri HSCSTSInfo (tipo di dati HSCSTS) sono utilizzati per determinare lo stato dell'High-Speed Counter.

CountEnable

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.CountEnable	BOOL	0...9	sola lettura

Indica lo stato dell'High-Speed Counter, a prescindere che il conteggio sia abilitato (1) o disabilitato (0, impostazione predefinita).

ErrorDetected

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.ErrorDetected	BOOL	0...9	lettura/scrittura

Rileva se è presente un errore nel sottosistema dell'HSC. Gli errori di configurazione sono i tipi di errore più comuni segnalati da ErrorDetectedr. Quando il bit è impostato (1), controllare lo specifico codice di errore nel parametro HSCSTS.ErrorCode, che è mantenuto dal controllore. Se necessario, è possibile cancellare il bit ErrorDetected.

CountUpFlag

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.CountUpFlag	BOOL	0...9	sola lettura

Utilizzato con tutti gli HSC (modalità 0...9). Se il bit HSCSTS.CountEnable è impostato, il bit Count Up è impostato (1). Se il bit HSCSTS.CountEnable è cancellato, il bit Count Up è cancellato (0).

CountDownFlag

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.CountDownFlag	BOOL	2...9	sola lettura

Utilizzato con i contatori bidirezionali (modalità 2...9). Se è impostato il bit HSCSTS.CountEnable, il bit Count Down è impostato (1). Se il bit HSCSTS.CountEnable viene cancellato, il bit Count Down è cancellato (0).

Mode1Done

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.Mode1Done	BOOL	0 oppure 1	lettura/scrittura

Il sottosistema dell'HSC imposta il flag di stato HSCSTS.Mode1Done su (1) quando l'HSC è configurato per il comportamento in modalità 0 oppure 1 e l'accumulatore esegue il conteggio a crescere fino al valore alto preimpostato.

OVF

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.OVF	BOOL	0...9	lettura/scrittura

Il sottosistema dell'HSC imposta il flag di stato HSCSTS.OVF su (1) ogniqualvolta sia stato eseguito il conteggio del valore accumulato (HSCSTS.Accumulator) fino alla variabile di overflow (HSCAPP.OFSetting). Il bit è transitorio e viene impostato dal sottosistema dell'HSC. L'utilizzo, il tracciamento e la cancellazione (0) di una condizione di overflow sono gestiti dal programma di controllo.

Le condizioni di overflow non generano errori del controllore.

UNF

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.UNF	BOOL	0...9	lettura/scrittura

Il sottosistema dell'HSC imposta il flag di stato HSCSTS.UNF su (1) ogniqualvolta venga eseguito il conteggio del valore accumulato (HSCSTS.Accumulator) fino alla variabile di underflow (HSCAPP.UFSetting). Il bit è transitorio e viene impostato dal sottosistema dell'HSC. L'utilizzo, il tracciamento e la cancellazione (0) di una condizione di underflow sono gestiti dal programma di controllo.

Le condizioni di underflow non generano errori del controllore.

CountDir

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.CountDir	BOOL	0...9	sola lettura

Il sottosistema dell'HSC controlla il flag di stato HSCSTS.CountDir. Quando l'accumulatore dell'HSC esegue il conteggio crescente, il flag di direzione è impostato (1). Quando l'accumulatore dell'HSC esegue il conteggio decrescente, il flag di direzione è cancellato (0).

Se il valore accumulato si interrompe, il bit di direzione mantiene il valore. Il flag di direzione cambia solo se viene invertita la direzione del conteggio accumulato.

Questo bit è costantemente aggiornato dal sottosistema dell'HSC se il controllore è in una modalità di esecuzione.

HPReached

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.HPReached	BOOL	2...9	lettura/scrittura

Il sottosistema dell'HSC imposta il flag di stato HSCSTS.HPReached su (1) ogniqualvolta il valore accumulato (HSCSTS.Accumulator) sia superiore o uguale alla variabile del valore alto preimpostato (HSCAPP.HPSSetting).

Questo bit è costantemente aggiornato dal sottosistema dell'HSC ogniqualvolta il controllore si trovi in una modalità di esecuzione. La scrittura in questo elemento non è consigliata.

LPReached

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.LPReached	BOOL	2...9	sola lettura

Il sottosistema dell'HSC imposta il flag di stato HSCSTS.LPReached su (1) ogniqualvolta il valore accumulato (HSCSTS.Accumulator) sia inferiore o uguale alla variabile del valore basso preimpostato (HSCAPP.LPSetting).

Questo bit è costantemente aggiornato dal sottosistema dell'HSC ogniqualvolta il controllore si trovi in una modalità di esecuzione. La scrittura in questo elemento non è consigliata.

OFCauseInter

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.OFCauseInter	BOOL	0...9	lettura/scrittura

Il bit di stato dell'interruzione per overflow è impostato (1) quando l'accumulatore dell'HSC esegue il conteggio fino al valore di overflow e l'interruzione dell'HSC viene attivata. Questo bit può essere utilizzato nel programma di controllo per identificare la variabile di overflow che ha causato l'interruzione dell'HSC. Se il programma di controllo deve eseguire un'azione di controllo specifica in base all'overflow, questo bit è utilizzato come logica condizionale.

Questo bit può essere cancellato (0) dal programma di controllo e viene cancellato anche dal sottosistema dell'HSC in presenza delle seguenti condizioni:

- Esecuzione dell'interruzione con valore basso preimpostato
- Esecuzione dell'interruzione con valore alto preimpostato
- Esecuzione dell'interruzione per underflow

UFCauseInter

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.UFCauseInter	BOOL	2...9	lettura/scrittura

Il bit di stato dell'interruzione per underflow è impostato (1) quando l'accumulatore dell'HSC esegue il conteggio fino al valore di underflow e l'interruzione dell'HSC viene attivata. Questo bit può essere utilizzato nel programma di controllo per identificare la condizione di underflow che ha causato l'interruzione dell'HSC. Se il programma di controllo deve eseguire un'azione di controllo specifica in base all'underflow, questo bit è utilizzato come logica condizionale.

Questo bit può essere cancellato (0) dal programma di controllo e viene cancellato anche dal sottosistema dell'HSC in presenza delle seguenti condizioni:

- Verificarsi dell'interruzione con valore basso preimpostato
- Verificarsi dell'interruzione con valore alto preimpostato
- Verificarsi dell'interruzione per overflow

HPCauseInter

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.HPCauseInter	BOOL	0...9	lettura/scrittura

Il bit di stato dell'interruzione con valore alto preimpostato è impostato (1) quando l'accumulatore dell'HSC raggiunge il valore alto preimpostato e l'interruzione dell'HSC viene attivata. Questo bit può essere utilizzato nel programma di controllo per identificare la condizione di valore alto preimpostato che ha causato l'interruzione dell'HSC. Se il programma di controllo deve eseguire un'azione di controllo specifica in base al valore alto preimpostato, questo bit è utilizzato come logica condizionale.

Questo bit può essere cancellato (0) dal programma di controllo e viene cancellato anche dal sottosistema dell'HSC in presenza delle seguenti condizioni:

- Verificarsi dell'interruzione con valore basso preimpostato
- Verificarsi dell'interruzione per underflow
- Verificarsi dell'interruzione per overflow

LPCauseInter

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.LPCauseInter	BOOL	2...9	lettura/scrittura

Il bit di stato dell'interruzione con valore basso preimpostato è impostato (1) quando l'accumulatore dell'HSC raggiunge il valore basso preimpostato e l'interruzione dell'HSC viene attivata. Questo bit può essere utilizzato nel programma di controllo per identificare la condizione di valore basso preimpostato che ha causato l'interruzione dell'HSC. Se il programma di controllo deve eseguire un'azione di controllo specifica in base al valore basso preimpostato, questo bit è utilizzato come logica condizionale.

Questo bit può essere cancellato (0) dal programma di controllo e viene cancellato anche dal sottosistema dell'HSC in presenza delle seguenti condizioni:

- Verificarsi dell'interruzione con valore alto preimpostato
- Verificarsi dell'interruzione per underflow
- Verificarsi dell'interruzione per overflow

PLSPosition

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.PLSPosition	UINT	0...9	sola lettura

Quando l'HSC è in modalità di conteggio ed è abilitata la funzione PLS, questo parametro indica l'elemento di PLS in uso per l'attuale configurazione dell'HSC.

ErrorCode

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.ErrorCode	BOOL	0...9	sola lettura

Visualizza i codici di errore rilevati dal sottosistema dell'HSC.

Sottoelemento del codice di errore	Codice di errore conteggio HSC	Accesso al programma utente
Bit 15-8 (byte alto)	0-255	Il valore non nullo per byte alto indica che l'errore HSC è causato dalle impostazioni dei dati PLS. Il valore del byte alto indica quale elemento dei dati PLS fa verificare l'errore.
Bit 7-0 (byte basso)	0x00	Nessun errore.
	0x01	Modalità conteggio HSC non valida.
	0x02	Valore alto preimpostato non valido.
	0x03	Overflow non valido.
	0x04	Underflow non valido.
	0x05	Nessun dato PLS.

Accumulator

Parametri	Tipo di dati	Accesso al programma utente
HSCApp.Accumulator	DINT	lettura/scrittura

Imposta il valore iniziale dell'accumulatore quando si attiva l'High-Speed Counter. Quando l'High-Speed Counter è in modalità conteggio, l'accumulatore viene automaticamente aggiornato dal sottosistema dell'HSC, per propagare il valore effettivo dell'accumulatore dell'HSC.

HP

Parametri	Tipo di dati	Accesso al programma utente
HSCSTS.HP	DINT	sola lettura

HSCSTS.HP è il SetPoint superiore (nei conteggi) che definisce quando il sottosistema dell'HSC genera un'interruzione.

Il numero dei dati caricati nell'impostazione valore alto preimpostato deve essere inferiore o uguale ai dati residenti nel parametro di overflow (HSCAPP.OFSetting). In caso contrario, viene generato un errore di HSC.

Questa è l'impostazione più recente del valore alto preimpostato che può essere aggiornata tramite la funzione PLS dal blocco di dati PLS.

LP

Parametri	Tipo di dati	Modalità HSC	Accesso al programma utente
HSCSTS.LP	DINT		sola lettura

HSCSTS.LP è il SetPoint inferiore (nei conteggi) che definisce quando il sottosistema dell'HSC genera un'interruzione.

Il numero dei dati caricati nell'impostazione valore basso preimpostato deve essere superiore o uguale ai dati residenti nel parametro di underflow (HSCAPP.UFSetting). In caso contrario, viene generato un errore di HSC. Se i valori di underflow e dell'impostazione valore basso preimpostato sono numeri negativi, il valore basso preimpostato deve essere un numero con un valore assoluto inferiore.

Questa è l'impostazione più recente del valore basso preimpostato che può essere aggiornata tramite la funzione PLS dal blocco di dati PLS.

HPOutput

Parametri	Tipo di dati	Accesso al programma utente
HSCApp.HPOutput	UDINT	lettura/scrittura

Definisce lo stato (1 = ON oppure 0 = OFF) delle uscite del controllore quando viene raggiunto il valore alto preimpostato . Per maggiori informazioni su come impostare direttamente su On oppure Off in base al valore alto preimpostato.

È possibile configurare il modello di bit uscita alta durante l'impostazione iniziale o utilizzare il blocco di funzione HSC per caricare i nuovi parametri durante il funzionamento del controllore.

LPOutput

Parametri	Tipo di dati	Accesso al programma utente
HSCApp.LPOutput	UDINT	lettura/scrittura

LPOutput (HSCApp.LPOutput) definisce lo stato (1 = "on" oppure 0 = "off") delle uscite del controllore quando viene raggiunto il valore basso preimpostato. Per maggiori informazioni su come impostare direttamente le uscite su ON oppure OFF in base al valore basso preimpostato.

È possibile configurare il modello di bit uscita bassa durante l'impostazione iniziale o utilizzare il blocco di funzione HSC per caricare i nuovi parametri durante il funzionamento del controllore.

Esempio di parametri HSCSTSInfo

Nella seguente immagine sono rappresentati i parametri HSCSTSInfo nel **Selettore di variabili**.

Nome	Alias	Tipo di dati	Dimensione	Valore progett	Valore iniziale
HSC_1		HSC	
HSC_1.Enable		BOOL			
HSC_1.HscCmd		USINT			
HSC_1.HscAppData		HSCAPP	
HSC_1.HscSTSInfo		HSCSTS	
HSC_1.HscSTSInfo.CountEnable		BOOL			
HSC_1.HscSTSInfo.ErrorDetected		BOOL			
HSC_1.HscSTSInfo.CountUpFlag		BOOL			
HSC_1.HscSTSInfo.CountDwnFlag		BOOL			
HSC_1.HscSTSInfo.Mode1Done		BOOL			
HSC_1.HscSTSInfo.OVF		BOOL			
HSC_1.HscSTSInfo.JNF		BOOL			
HSC_1.HscSTSInfo.CountDir		BOOL			
HSC_1.HscSTSInfo.HPRReached		BOOL			
HSC_1.HscSTSInfo.LPReached		BOOL			
HSC_1.HscSTSInfo.OFCauseInter		BOOL			
HSC_1.HscSTSInfo.JFCauseInter		BOOL			
HSC_1.HscSTSInfo.HPCauseInter		BOOL			
HSC_1.HscSTSInfo.LPCauseInter		BOOL			
HSC_1.HscSTSInfo.PlsPosition		UINT			
HSC_1.HscSTSInfo.ErrorCode		UINT			
HSC_1.HscSTSInfo.Accumulator		DINT			
HSC_1.HscSTSInfo.HP		DINT			
HSC_1.HscSTSInfo.LP		DINT			
HSC_1.HscSTSInfo.HPOutput		UDINT			
HSC_1.HscSTSInfo.LPOutput		UDINT			

Vedere anche

[HSC](#) a pagina 298

Tipo di dati PLS

PLSData (tipo di dati PLS) è utilizzato per configurare l'interruttore di fine corsa programmabile .

Elementi della struttura PLSData

La struttura dati PLS è un array flessibile con i seguenti elementi.

Elemento	Ordine elemento	Tipo di dati	Descrizione elemento
HSCHP	Parola 0...1	DINT	Valore alto preimpostato
HSCLP	Parola 2...3	DINT	Valore basso preimpostato
HSCHPOutput	Parola 4...5	UDINT	Dati uscita valore alto
HSCLPOutput	Parola 6...7	UDINT	Dati uscita valore basso

Il numero totale degli elementi di una struttura dati PLS non deve superare 255.

Parametri PLSData

Nella seguente tabella sono elencati i dettagli relativi ai parametri PLSData.

Elemento	Tipo di dati	Formato dati	Modalità HSC	Accesso al programma utente	Descrizione
HSCHP	DINT	Intero con segno a 32 bit	0	lettura/scrittura	Valore alto preimpostato
HSCLP	DINT	Intero con segno a 32 bit	0	lettura/scrittura	Valore basso preimpostato
HSCHPOutput	UDINT	Binario a 32 bit	0	lettura/scrittura	Dati uscita valore alto
HSCLPOutput	UDINT	Binario a 32 bit	0	lettura/scrittura	Dati uscita valore basso

Confronto tra impostazioni HSCApp e impostazioni PLSData

Quando la funzione PLS è abilitata, le impostazioni HSCApp pertinenti sono sostituite dalle impostazioni PLSData corrispondenti, come descritto nella seguente tabella.

Impostazione HSCApp	Impostazione PLSData
HSCAPP.HpSetting	HSCHP
HSCAPP.LpSetting	HSCLP
HSCAPP.HPOutput	HSCHPOutput
HSCAPP.LPOutput	HSCLPOutput

Esempio di parametri PLSData

Nella seguente figura sono rappresentati i parametri PLSData nel Selettore di variabili.

Nome	Alias	Tipo di dati	Dimensione	Valore predef.	Valore iniziale	Commento	Dimensione stringa
HSC_1		HSC					
HSC_1.Enable		BOOL				Abilita blocco funzione	
HSC_1.HscCmd		USINT				Consultare Valori del comando HSC.	
HSC_1.HscAppData		HSCAPP				Configurazione applicazione HSC.	
HSC_1.HscStlInfo		HSCSTS				Stato HSC dinamico.	
HSC_1.PlsData						Struttura dati dell'interruttore di fine corsa programmabile (PLS)	
HSC_1.PlsData[1]		PLS					
HSC_1.PlsData[1].HschP		DINT					
HSC_1.PlsData[1].HschLP		DINT					
HSC_1.PlsData[1].HschPOutPut		UDINT					
HSC_1.PlsData[1].HschLPOutPut		UDINT					
HSC_1.Sts		UINT				Stato esecuzione. Consultare Valori dello stato HSC.	

Vedere anche

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Istruzioni per l'High-Speed Counter \(HSC\)](#) a pagina 297

Tipo di dati HSCE_CHANNEL

Descrive il tipo di dati the HSCE_CHANNEL:

```
@typedef struct struct_HSCE_channel
{
```

```

USINT   ModuleType;
USINT   SlotID;
USINT   HSCID;
} HSCE_CHANNEL;
    
```

La descrizione è riportata di seguito:

Byte	Descrizione
Tipo di modulo	0x00: Incorporato 0x01: Espansione (sezionato) 0x02: Porta universale
ID slot	0x00: Incorporato 0x01-0x1F: ID del modulo di espansione (sezionato) 0x01-0x05: ID della porta universale
HSCID	0x00-0x0F: Incorporato 0x00-0x07: ID di HSC per espansione 0x00-0x01: ID di HSC per porta universale

Vedere anche

[HSC](#) a pagina 298

Tipo di dati HSCE_STS

Di seguito è riportata la descrizione del tipo di dati HSCE_STS:

Parametro	Tipo di dati	Modalità HSC	Descrizione
CountEnable	BOOL	0...13	Conteggio abilitato.
ErrorDetected	BOOL	0...13	Diverso da zero indica un errore rilevato.
CountUpFlag	BOOL	0...13	Flag conteggio crescente.
CountDwnFlag	BOOL	2...13	Flag conteggio decrescente.
Mode1Done	BOOL	0 oppure 1	HSC è Mode 1A o Mode 1B; l'accumulatore conta fino al valore HP.
OVF	BOOL	0...13	Rilevato un overflow.
UNF	BOOL	0...13	Rilevato un underflow.
CountDir	BOOL	0...13	1: conteggio crescente; 0: conteggio decrescente.
HPReached	BOOL	2...13	Raggiunto valore alto preimpostato.
LPReached	BOOL	2...13	Raggiunto valore basso preimpostato.
OFCauseInter	BOOL	0...13	Overflow causato da interruzioni dell'HSC.
UFCauseInter	BOOL	2...13	Underflow causato da interrupt di HSC.
HPCauseInter	BOOL	0...13	Raggiunto valore alto preimpostato che ha causato un'interruzione nell'HSC.
LPCauseInter	BOOL	2...13	Raggiunto valore basso preimpostato che ha causato un'interruzione nell'HSC.
StatelInfo	USINT		Informazioni macchina di istruzioni contatore HSCE
PlsPosition	UINT	0...9	Posizione del interruttore di fine corsa programmabile. Il parametro PLSPosition viene reimpostato al termine di un ciclo completo e al raggiungimento del valore HP.
ErrorCode	UINT	0...13	Visualizza i codici di errore rilevati dal sottosistema dell'HSC.
Accumulator	DINT		Letture attuale accumulatore.
HP	DINT		Impostazione valore alto preimpostato più recente.
LP	DINT		Impostazione valore basso preimpostato più recente.
HPOutput	UDINT		Impostazione uscita valore alto preimpostato più recente.
LPOutput	UDINT		Impostazione uscita valore basso preimpostato più recente.

Vedere anche

[HSC](#) a pagina 298

Tipo di dati PLS_HSCE

La funzione interruttore di fine corsa programmabile (PLS) è un insieme aggiuntivo di modalità operative per l'High Speed Counter. In queste modalità i valori dei dati preimpostati di uscita vengono aggiornati usando i dati forniti dall'utente ogni volta che viene raggiunto uno dei valori preimpostati. Per programmare queste modalità, utilizzare un file PLS con i set di dati da usare. La struttura di dati PLS_HSCE è un vettore flessibile in cui ciascun elemento è definito come segue:

Elemento	Tipo di dati	Descrizione elemento
HighPreset	LINT	Impostazione valore alto preimpostato
LowPreset	LINT	Impostazione valore basso preimpostato
HiPresetOutput	UDINT	Dati uscita valore alto preimpostato
LoPresetOutput	UDINT	Dati uscita valore basso preimpostato

```
// struttura per elemento PLS
typedef struct PLS_HSCE_EleStruct
{
    LINT HighPreset;      // valore alto preimpostato HSC
    LINT LowPreset;      // valore basso preimpostato di
                        HSC
    UDINT HiPresetOutput; // uscita valore alto
                        preimpostato di HSC
    UDINT LoPresetOutput; // uscita valore basso
                        preimpostato di HSC
} PLS_HSCE_EleStruct;
```

Il numero totale degli elementi per una serie di dati PLS_HSCE non deve essere maggiore di 24 per plug-in HSC.

Vedere anche

[HSC](#) a pagina 298

HSC_SET_STS (stato di impostazione dell'High-Speed Counter)

HSC_SET_STS permette l'impostazione o il ripristino manuale dello stato del conteggio dell'HSC.

Dettagli operazione:

- Il blocco funzione dell'HSC deve essere interrotto (conteggio fermo) per consentire l'impostazione o il ripristino dello stato HTS da parte del blocco funzione HSC_SET_STC. Se la funzione HSC non viene interrotta, i parametri di ingresso continuano a essere aggiornati e le modifiche eseguite utilizzando che HSC_SET_STS verranno ignorate.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830 e Micro850.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

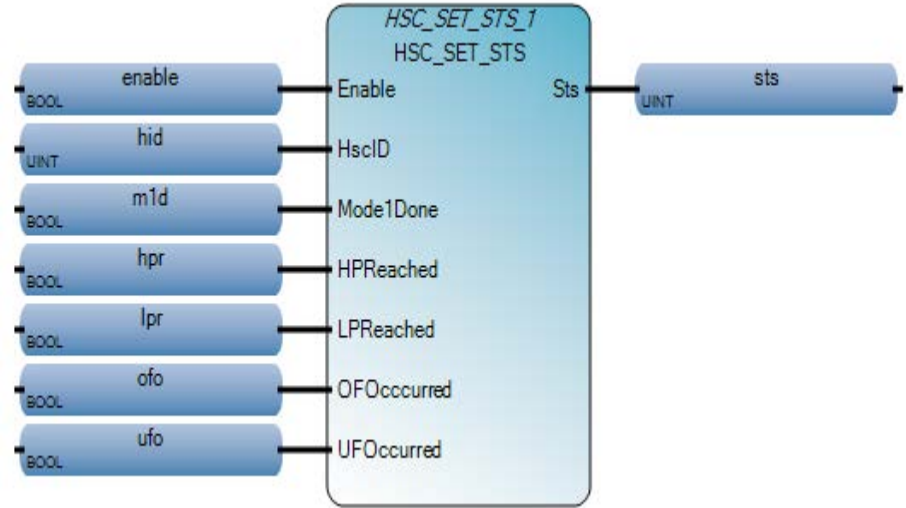
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: imposta/segue il reset dello stato dell'HSC. FALSE: nessuna modifica allo stato HSC.
HscID	Ingresso	UINT	Permette l'impostazione o il ripristino manuale dello stato dell'HSC.
Mode1Done	Ingresso	BOOL	Conteggio modalità 1A o 1B completato. Può essere impostato o ripristinato quando HSC non è in conteggio.
HPReached	Ingresso	BOOL	Raggiunto valore alto preimpostato. Può essere impostato o ripristinato quando HSC non è in conteggio.
LPReached	Ingresso	BOOL	Raggiunto valore basso preimpostato. Può essere impostato o ripristinato quando HSC non è in conteggio.
OFOccurred	Ingresso	BOOL	Si è verificato un overflow. Può essere impostato o ripristinato quando HSC non è in conteggio.
UFOccurred	Ingresso	BOOL	Si è verificato un underflow. Può essere impostato o ripristinato quando HSC non è in conteggio.
Sts	Uscita	UINT	I codici di stato sono definiti nei codici di stato HSC (Sts).
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Codici di stato (Sts) dell'HSC

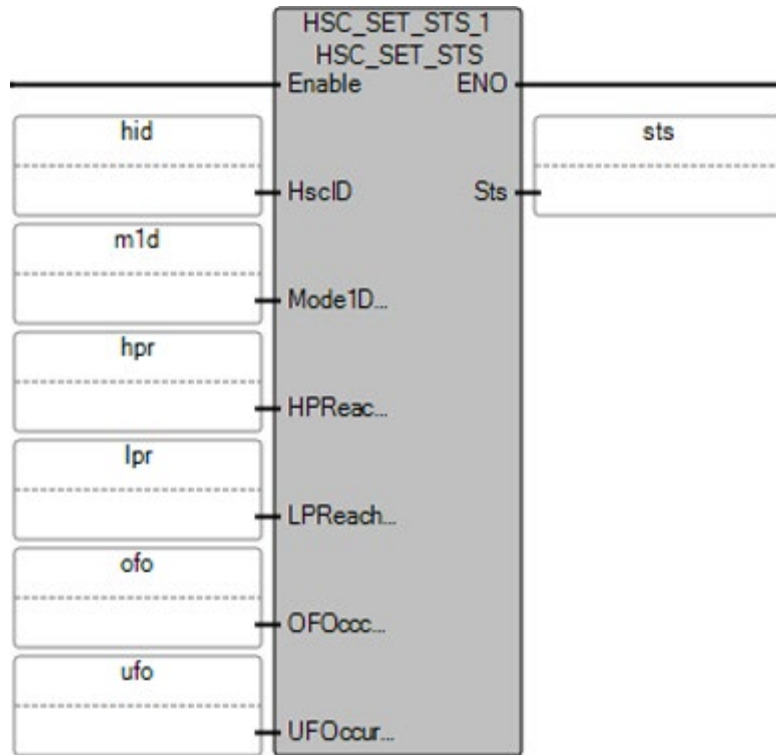
Nella tabella seguente sono descritti i codici di stato relativi al blocco funzione HSC.

Codice stato	Descrizione stato
0x00	Nessuna azione intrapresa (non abilitato).
0x01	Esecuzione dell'HSC completata.
0x02	Comando HSC non valido.
0x03	ID HSC fuori intervallo.
0x04	Errore di configurazione HSC.

Esempio di programmazione a blocchi funzionali HSC_SET_STS



Esempio di diagramma ladder HSC_SET_STS



Esempio di testo strutturato HSC_SET_STS

```
HSC_SET_STS_1 {
  void HSC_SET_STS_1(BOOL Enable, UINT HscID, BOOL Mode1Done, BOOL HPReached, BOOL LPReached, BOOL OFOccurred, BOOL UFOccurred)
  Tipo: HSC_SET_STS, Impostare/reimpostare manualmente lo stato dell'HSC.

  1 HSC_SET_STS_1(enable, hid, m1d, hpr, lpr, ofo, ufo);
  2 sts := HSC_SET_STS_1.Sts;
```


Vedere anche

[Istruzioni per l'High-Speed Counter \(HSC\)](#) a pagina 297

[Utilizzo delle istruzioni dell'High-Speed Counter](#) a pagina 317

Utilizzo delle istruzioni dell'High-Speed Counter

Questa sezione fornisce dettagli ed esempi specifici per l'utilizzo delle istruzioni dell'high-speed counter nei programmi logici, inclusi i seguenti argomenti:

[Aggiornamento dei dati dell'applicazione HSC](#) a pagina 317

[Configurazione di un Interrupt utente dell'High Speed Counter \(HSC\)](#) a pagina 318

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Esempio: come creare un'High-Speed Counter \(HSC\)](#) a pagina 324

[Esempio: come aggiungere una funzione interruttore di fine corsa programmabile \(PLS\)](#) a pagina 334

[Esempio: interruttore fine corsa programmabile \(PLS\) abilitato](#) a pagina 335

Aggiornamento dei dati dell'applicazione HSC

La configurazione dell'HSC è definita nei dati dell'applicazione HSC e di norma viene eseguita solo una volta prima di programmare l'istruzione dell'HSC. Mentre l'HSC sta eseguendo il conteggio, le modifiche apportate ai dati dell'applicazione HSC (parametro HSCAppData) vengono ignorate.

Per aggiornare la configurazione dell'HSC

1. Aggiornare HSCAppData.
2. Chiamare l'istruzione dell'HSC con il comando 0x03 (set/reload).

Vedere anche

[Configurazione di un Interrupt utente dell'High Speed Counter \(HSC\)](#) a pagina 318

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Esempio: come creare un programma High-Speed Counter \(HSC\)](#) a pagina 324

[Esempio: interruttore fine corsa programmabile \(PLS\) abilitato](#) a pagina 335

[HSC \(High Speed Counter\)](#) a pagina 298

Finestra di dialogo Interruzione da parte dell'utente dell'High-Speed Counter (HSC)

Come aprire la finestra di dialogo Interruzione da parte dell'utente dell'High-Speed Counter (HSC)

Tra i tipi di interruzione, selezionare **Interruzione da parte dell'utente dell'HSC (High Speed Counter)**.

La finestra di dialogo per l'interruzione dell'HSC può essere utilizzata per le seguenti operazioni:

- Configurazione delle proprietà di interruzione, quali ID e programma da utilizzare.
- Configurazione dei parametri dell'interruzione.

Vedere anche

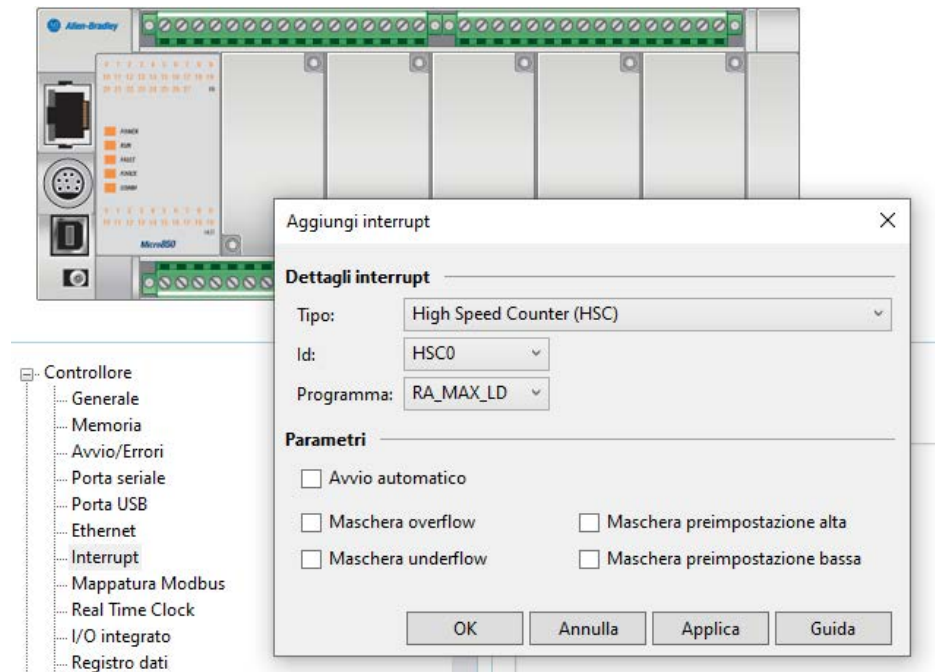
[Configurazione di un Interrupt utente dell'High Speed Counter \(HSC\)](#) a pagina 318

[Istruzioni per l'High-Speed Counter](#) a pagina 297

Configurazione di un Interrupt utente dell'High Speed Counter (HSC)

Un'interruzione da parte dell'utente causa la sospensione dell'attività in esecuzione nel controllore, l'esecuzione di un'attività diversa e infine il ritorno all'attività precedente, nel punto in cui era stata sospesa.

I controllori Micro830, Micro850 e Micro870 supportano fino a sei interrupt utente HSC utilizzabili per eseguire la logica selezionata dall'utente quando si verifica un evento preconfigurato.



Vedere anche

[Aggiornamento dei dati dell'applicazione HSC](#) a pagina 317

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Esempio: Come creare un programma per l'High Speed Counter \(HSC\)](#) a pagina 324

[Esempio: interruttore fine corsa programmabile \(PLS\) abilitato](#) a pagina 335

[HSC \(High Speed Counter\)](#) a pagina 298

Aggiunta e configurazione di un'interruzione da parte dell'utente dell'High-Speed Counter (HSC)

Per aggiungere e configurare un interrupt HSC dallo spazio di lavoro di configurazione del controllore, attenersi alla seguente procedura.

Per aggiungere un'interruzione HSC:

1. In **Organizzatore progetto** fare doppio clic sul controllore per aprire lo spazio di lavoro del controllore.
2. Nella struttura ad albero del **controllore**, fare clic su **Interrupt** per visualizzare la pagina di configurazione **Interrupt**.
3. Fare clic con il pulsante destro del mouse su una riga vuota, quindi fare clic su **Aggiungi** per visualizzare la finestra di dialogo con le proprietà dell'**Interrupt**.

4. Per configurare un interrupt HSC:
 - Tra i tipi di interruzione, selezionare **Interruzione da parte dell'utente dell'HSC (High Speed Counter)**.
 - Selezionare le proprietà interrupt HSC.
 - Selezionare i parametri interrupt HSC.
5. Chiudere la finestra di dialogo delle proprietà **Interrupt**.

Vedere anche

[Configurazione di un Interrupt utente dell'High Speed Counter \(HSC\)](#) a pagina 318

[Istruzioni per l'High-Speed Counter](#) a pagina 297

[Proprietà interruzione HSC](#) a pagina 320

[Parametri interruzione HSC](#) a pagina 321

Proprietà interruzione HSC

I bit di stato nelle proprietà dell'interruzione HSC indicano gli stati di abilitazione/disabilitazione e di esecuzione e se la condizione dell'interruzione è perduta o meno.

Abilitazione dell'interruzione da parte dell'utente (HSCO.Enabled)

Parametri	Formato dati	Modalità HSC	Accesso al programma utente
HSCO.Enabled	bit	0...9	sola lettura

Il bit Enabled è utilizzato per indicare lo stato di abilitazione o disabilitazione dell'interruzione dell'HSC.

Esecuzione dell'interruzione da parte dell'utente (HSCO.EX)

Parametri	Formato dati	Modalità HSC	Accesso al programma utente
HSCO.Ex	bit	0...9	sola lettura

Il bit EX (esecuzione interruzione da parte dell'utente) è impostato (1) ogniqualvolta il sottosistema dell'HSC inizi a elaborare la sottoroutine dell'HSC a causa di una delle seguenti condizioni:

- Raggiunto valore basso preimpostato
- Raggiunto valore alto preimpostato
- Condizione di overflow - Conteggio crescente fino al valore di overflow
- Condizione di underflow - Conteggio decrescente fino al valore di underflow

Il bit EX dell'HSC può essere utilizzato nel programma di controllo come logica condizionale, per determinare se è in esecuzione un'interruzione dell'HSC.

Il sottosistema dell'HSC cancella (0) il bit EX quando il controllore completa l'elaborazione della sottoroutine dell'HSC.

Interruzione da parte dell'utente in attesa (HSCO.PE)

Parametri	Formato dati	Modalità HSC	Accesso al programma utente
HSCO.PE	bit	0...9	sola lettura

Il flag di stato PE (interruzione da parte dell'utente in attesa) indica che un'interruzione è momentaneamente sospesa. È possibile monitorare il bit di stato PE o utilizzarlo per la logica nel programma di controllo, se è necessario determinare quando non è possibile eseguire immediatamente una sottoroutine. Il bit PE è mantenuto dal controllore e viene impostato e cancellato automaticamente.

Interruzione da parte dell'utente persa (HSCO.LS)

Parametri	Formato dati	Modalità HSC	Accesso al programma utente
HSCO.LS	bit	0...9	sola lettura

Il flag di stato LS (interruzione da parte dell'utente persa) indica che un'interruzione è andata persa. Il controllore può elaborare una condizione attiva di interruzione da parte dell'utente e mantiene un'interruzione in attesa prima dell'impostazione del bit di perdita.

Il bit LS è impostato dal controllore. L'utilizzo e il monitoraggio di una condizione di perdita sono gestiti dal programma di controllo.

Vedere anche

[Configurazione di un Interrupt utente dell'High Speed Counter \(HSC\)](#) a pagina 318

Parametri interruzione HSC

I parametri dell'interruzione HSC servono per configurare le opzioni relative a maschera e avvio.

Avvio automatico (HSCO.AS)

Parametri	Formato dati	Modalità HSC	Accesso al programma utente
HSCO.AS	bit	0...9	sola lettura

L'avvio automatico è configurato con il dispositivo di programmazione e memorizzato come parte del programma utente. Il bit di avvio automatico definisce se la funzione di interruzione dell'HSC viene avviata automaticamente ogniqualvolta che il controllore entra in qualsiasi modalità di esecuzione o di collaudo.

Maschera overflow (HSCO.MV)

Il bit di controllo **MV** (maschera di overflow) viene utilizzato per abilitare (consentire) o disabilitare (non consentire) il verificarsi di un interrupt per overflow. Se il bit è disabilitato (0) e l'**HSC** rileva una condizione di

raggiunto overflow, l'interruzione **HSC** da parte dell'utente non viene eseguita.

Il bit **MV** è controllato dal programma utente e mantiene il proprio valore anche in caso di spegnimento e riaccensione. Il programma utente deve impostare e cancellare il bit **MV**.

Parametri	Formato dati	Modalità HSC	Accesso al programma utente
HSCO.MV	bit	0...9	sola lettura

Maschera underflow (HSCO.MN)

Parametri	Formato dati	Modalità HSC	Accesso al programma utente
HSCO.MN	bit	2...9	sola lettura

Il bit di controllo MN (maschera di underflow) viene utilizzato per abilitare (consentire) o disabilitare (non consentire) un interrupt per underflow. Se il bit è disabilitato (0) e l'HSC rileva una condizione di raggiunto underflow, l'interrupt utente HSC non viene eseguito.

Il bit MN è controllato dal programma utente e mantiene il proprio valore anche in caso di spegnimento e riaccensione. Il programma utente deve impostare e cancellare il bit MN.

Maschera preimpostata alta (HSCO.MH)

Parametri	Formato dati	Modalità HSC	Accesso al programma utente
HSCO.MH	bit	0...9	sola lettura

Il bit di controllo MH (maschera del valore alto preimpostato) è utilizzato per abilitare (consentire) o disabilitare (non consentire) un interrupt per valore alto preimpostato. Se il bit viene cancellato (0) e l'HSC rileva una condizione di raggiunto valore alto preimpostato, l'interrupt utente HSC non viene eseguito.

Il bit MH è controllato dal programma utente e mantiene il proprio valore anche in caso di spegnimento e riaccensione. Il programma utente deve impostare e cancellare il bit MH.

Maschera preimpostata bassa (HSCO.ML)

Parametri	Formato dati	Modalità HSC	Accesso al programma utente
HSCO.ML	bit	2...9	sola lettura

Il bit di controllo ML (maschera valore basso preimpostato) è utilizzato per abilitare (consentire) o disabilitare (non consentire) un interrupt per valore basso preimpostato. Se il bit viene cancellato (0) e viene rilevata dall'HSC una condizione di raggiunto valore basso preimpostato, l'interrupt utente HSC non viene eseguito.

Il bit ML è controllato dal programma utente e mantiene il proprio valore anche in caso di spegnimento e riaccensione. Il programma utente deve impostare e cancellare il bit ML.

Vedere anche

[Configurazione di un Interrupt utente dell'High Speed Counter \(HSC\)](#) a pagina 318

Configurazione di un interruttore di fine corsa programmabile (PLS)

L'high-speed counter prevede modalità operative aggiuntive per l'implementazione di un interruttore di fine corsa programmabile (PLS). La funzione PLS si utilizza per configurare l'High-Speed Counter in modo da farlo funzionare come un PLS o un commutatore rotativo a camme. La funzione PLS supporta fino a 255 coppie di valori alti e bassi preimpostati e può essere utilizzata quando è necessaria più di una coppia di questi valori.

Abilitazione della PLS nell'HSC

La modalità PLS è operativa solo insieme all'HSC del controllore Micro800 e deve essere abilitata nell'istruzione HSC impostando il parametro HSCAppData.PLSEnable su True.

Il parametro PLSPosition viene ripristinato al termine di un ciclo completo e al raggiungimento del valore HSCSTS.HP. Il reset dell'istruzione HSC o lo spostamento dello 0 sul parametro PLSPosition non esegue il reset di PLSPosition.

Operazione HSC con PLS abilitata

La funzione PLS può operare con tutte le altre funzionalità dell'HSC, compresa la possibilità di selezionare gli eventi HSC che generano un'interruzione da parte dell'utente.

Quando è abilitata la funzione PLS e il controllore è in modalità di esecuzione, l'HSC conta gli impulsi in ingresso e si verificano i seguenti eventi.

- Quando il conteggio raggiunge il primo valore preimpostato (HSCHP or HSCLP) definito nei dati PLS, i dati di origine per l'uscita (HSCHPOutput o HSCLPOutput) sono scritti nella maschera dell'HSC (HSCAPP.OutputMask).
- A questo punto i valori preimpostati successivi (HSCHP e HSCLP) definiti nei dati PLS diventano attivi.
- Quando l'HSC esegue il conteggio fino al nuovo valore preimpostato, i nuovi dati di uscita vengono scritti nella maschera dell'HSC.
- Questo processo continua fino al caricamento nel blocco dati PLS dell'ultimo elemento.
- A questo punto l'elemento attivo nel blocco dati PLS viene ripristinato a zero.
- Questo comportamento è definito operazione circolare.

La differenza valore preimpostato PLS tra HSC integrato e modulo plug-in HSC

I comportamenti preimpostati HSCHP e HSCLP PLS sono diversi tra modulo HSC integrato e modulo plug-in HSC. Il bit valore altro preimpostato HSC integrato verrà impostato solo quando viene eseguito l'ultimo PLS, mentre il bit valore alto preimpostato del modulo plug-in HSC verrà impostato quando viene eseguito il primo PLS. Ad esempio:

- HSC PLS0-PLS23 integrato:
Valore altro preimpostato verrà impostato quando il valore PLS23HP=Accumulator
Valore basso preimpostato verrà impostato quando il valore PLS23LP=Accumulator
- PLS0-PLS23 plug-in HSC:
Valore altro preimpostato verrà impostato quando il valore PLS0HP=Accumulator
Valore basso preimpostato verrà impostato quando il valore PLS0LP=Accumulator

Vedere anche

[Configurazione degli interrupt da parte dell'utente dell'High-Speed Counter \(HSC\)](#) a pagina 318

[Esempio: Come creare un programma High Speed Counter \(HSC\)](#) a pagina 324

[Esempio: interruttore fine corsa programmabile \(PLS\) abilitato](#) a pagina 335

[HSC \(High Speed Counter\)](#) a pagina 298

[Aggiornamento dei dati dell'applicazione HSC](#) a pagina 317

Esempio: come creare un programma High-Speed Counter (HSC)

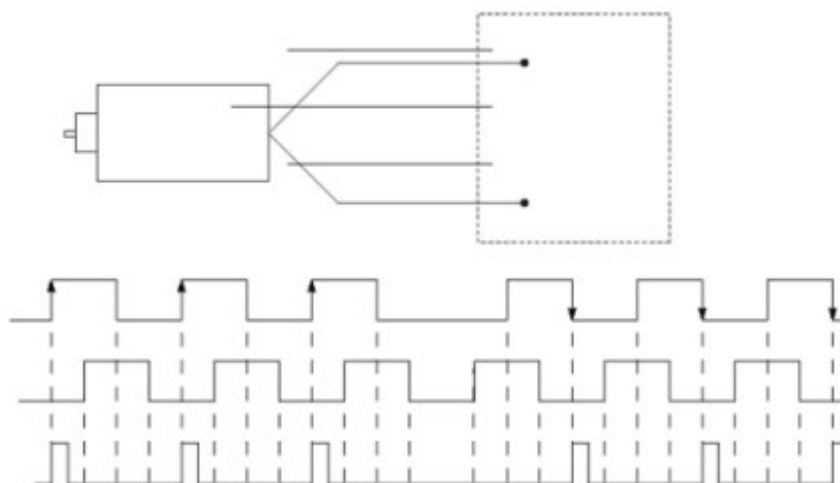
Questo esempio illustra come creare un programma High-Speed Counter (HSC) che impiega un encoder in quadratura e include una funzione interruttore di fine corsa programmabile (PLS).

Encoder in quadratura utilizzato nell'esempio

Nell'esempio di programma, per l'High Speed Counter si utilizza un blocco funzione dell'HSC e un contatore in quadratura con ingressi in fase A e B. L'encoder in quadratura determina la direzione di rotazione e la posizione dello strumento rotante, ad esempio un tornio. Il contatore bidirezionale conta le rotazioni dell'encoder in quadratura.

Il seguente encoder in quadratura è connesso agli ingressi 0 e 1. La direzione del conteggio è determinata dall'angolo di fase tra A e B:

- Se A porta a B, il contatore segna un incremento.
- Se B porta ad A, il contatore segna un decremento.



Creazione di un programma High-Speed Counter (HSC)

Eeguire le seguenti attività per creare, compilare e testare il programma HSC, quindi aggiungere una funzione PLS.

Intestazione tabella	Intestazione tabella
1	Creazione di un ladder diagram e aggiunta di variabili a pagina 325
2	Assegnazione di valori alle variabili HSC a pagina 328
3	Assegnazione di variabili e compilazione del programma a pagina 329
4	Test del programma ed esecuzione dell'High-Speed Counter a pagina 331
5	Aggiunta di una funzione interruttore di fine corsa programmabile (PLS) a pagina 334

Vedere anche

[Aggiornamento dei dati dell'applicazione HSC](#) a pagina 317

[Configurazione degli interrupt da parte dell'utente dell'High-Speed Counter \(HSC\)](#) a pagina 318

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Esempio: interruttore fine corsa programmabile \(PLS\) abilitato](#) a pagina 335

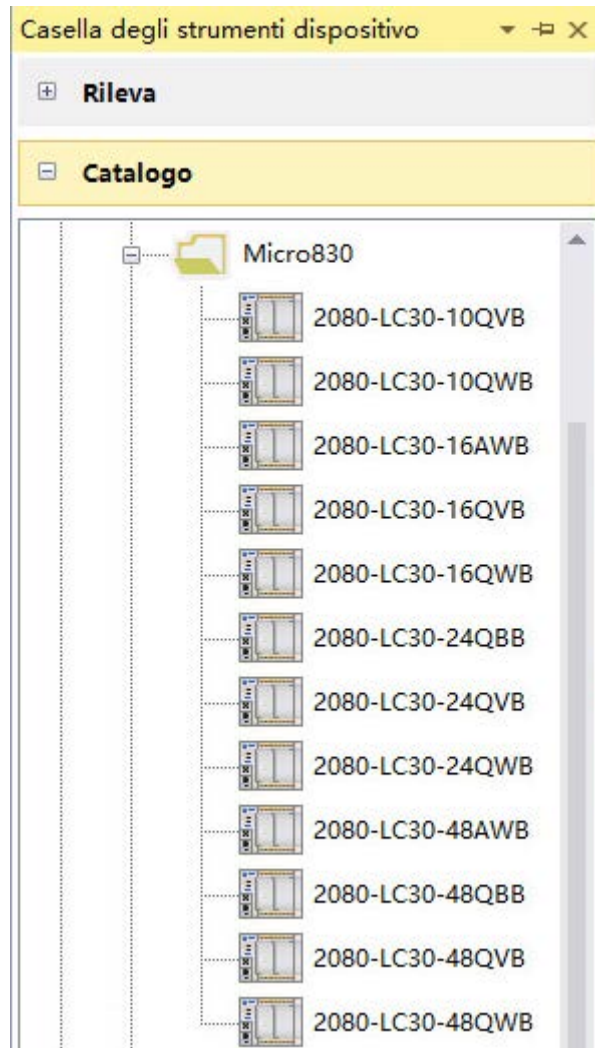
[HSC \(High Speed Counter\)](#) a pagina 298

Creazione di un ladder diagram e aggiunta di variabili

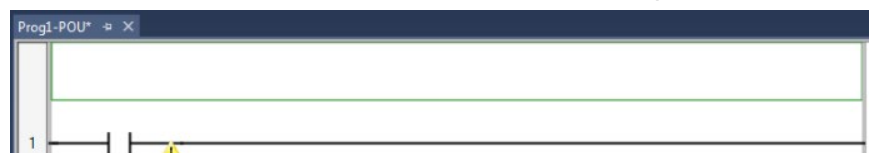
Creare un diagramma ladder e aggiungere variabili locali al ramo. In questo programma esemplificativo si utilizza un controllore 2080-LC50-24QVB. L'HSC è supportato da tutti i controllori Micro830 e Micro850, tranne i tipi di controllore 2080-LCxx-xxAWB.

Per creare un diagramma ladder e aggiungere variabili:

1. Nella **Casella degli strumenti dispositivo** espandere la scheda **Catalogo** per visualizzare le cartelle del dispositivo.
2. Espandere le cartelle Controllori e Micro830 per visualizzare tutti i controllori Micro830. Fare doppio clic su un controllore (2080-LC50-24QVB) per aggiungerlo all'**Organizzatore progetto**.

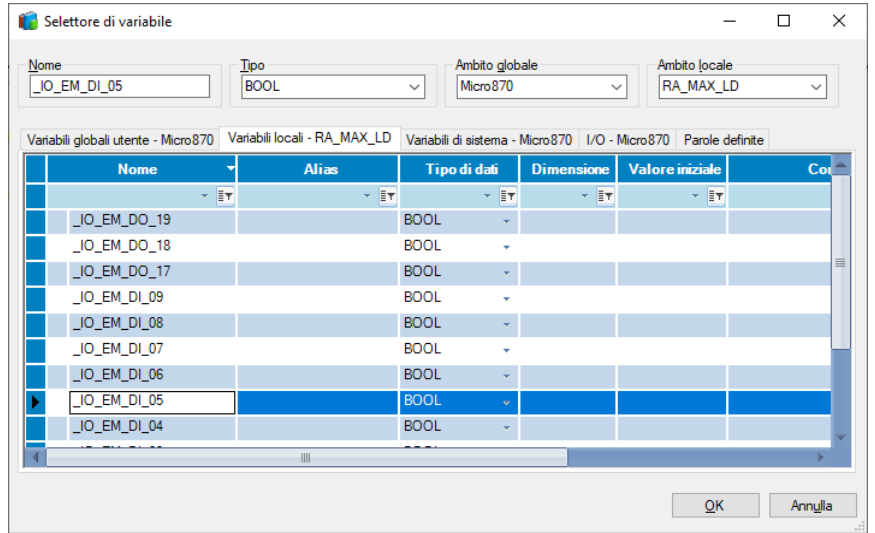


3. Nell'**Organizzatore progetto** fare clic con il pulsante destro del mouse su **Programmi**, scegliere **Aggiungi** e quindi fare clic su **Nuovo LD: ladder diagram** per aggiungere un nuovo programma con logica ladder.
4. Fare clic con il pulsante destro del mouse su **UntitledLD**, quindi scegliere **Apri**.
5. Nella finestra di dialogo **Casella degli strumenti**.
 - Fare doppio clic su **Contatto diretto** per aggiungerlo al piolo oppure
 - Trascinare e rilasciare un **Contatto diretto** sul piolo.

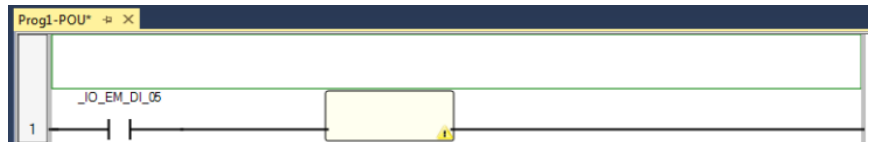


6. Assegnare una variabile al contatto diretto:

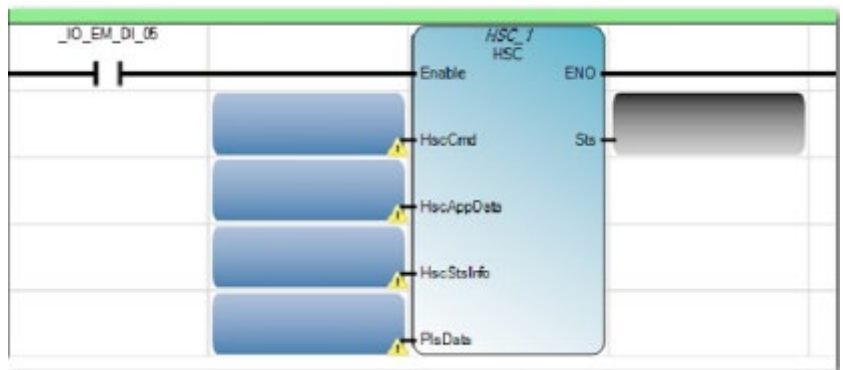
- Fare doppio clic sul contatto diretto per visualizzare il **Selettore di variabili**, quindi fare clic sulla scheda **I/O - Micro830**.
- Fare clic su **_IO_EM_DI_05** e quindi su **OK** per assegnare il contatto diretto all'ingresso 5.



7. Nella finestra di dialogo **Casella degli strumenti**, selezionare un blocco funzione e trascinarlo a destra del contatto diretto, come mostrato nella seguente immagine.



8. Fare doppio clic sul blocco funzione per visualizzare il **Selettore blocco istruzione**.
9. Nel **Selettore blocco istruzione**, selezionare **HSC** e fare clic su **OK**.
10. Verificare che il piolo del ladder risulti simile alla seguente figura.



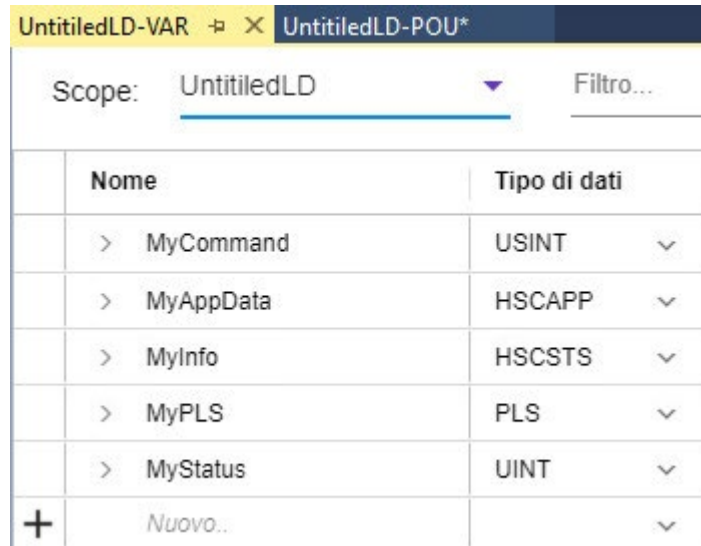
11. Nell'**Organizzatore progetto** fare doppio clic su **Variabili locali** per visualizzare la pagina **Variabili**.
12. Nella pagina **Variabili**, aggiungere le variabili e i tipi di dati riportati di seguito.

Nome variabile	Tipo di dati
MyCommand	USINT
MyAppData	HSCAPP
MyInfo	HSCSTS

MyPLS	PLS
MyStatus	UINT

Risultato

La pagina **Variabili** deve corrispondere all'immagine seguente:



Vedere anche

[Aggiunta di una funzione interruttore di fine corsa programmabile \(PLS\)](#) a pagina 334

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Esempio: Come creare un programma per l'High Speed Counter \(HSC\)](#) a pagina 324

[HSC \(High Speed Counter\)](#) a pagina 298

Assegnazione di valori alle variabili HSC

Una volta aggiunte le variabili, attenersi alla seguente procedura per aggiungere valori alle variabili utilizzando la colonna **Valore iniziale** nel **Selettore di variabili**. Un programma standard di norma utilizza una routine per assegnare valori alle variabili.

Per assegnare valori alle variabili HSC:

1. Espandere **MyAppData** per visualizzare tutte le variabili.
2. Assegnare il valore della modalità HSC:
 - Nel campo **Valore iniziale** della variabile MyAppData.HSCMode digitare 6.
 - Per maggiori informazioni sulla descrizione di ciascun valore, vedere HSCMode in Tipo di dati HSCAPP.

3. Assegnare il resto dei valori alle variabili MyAppData, come mostrato nella seguente figura.
 - Nel campo **Valore iniziale** immettere il valore.
 - Vedere tipo di dati HSCAPP per maggiori informazioni sulla descrizione di ciascun valore.

Nome	Alias	Tipo di dati	Dimensione	Valore predef.	Valore Iniziale
MyAppData		HSCAPP	
MyAppData.PlsEnable		BOOL		FALSE	
MyAppData.HsciD		UINT		0	
MyAppData.IscVoids		UINT	6		
MyAppData.Accumulator		DINT			
MyAppData.HPSSetting		DINT	40		
MyAppData.LPSSetting		DINT	-40		
MyAppData.OFSSetting		DINT	50		
MyAppData.LFSSetting		DINT	-50		
MyAppData.OutputMask		UDINT	3		
MyAppData.HPOutput		UDINT	1		
MyAppData.LPOOutput		UDINT	2		

4. Assegnare il valore comando HSC:
 - Nel campo **Valore iniziale** della variabile MyCommand digitare 1.
 - Per maggiori informazioni sui valori del comando, vedere [Valori HSCCmd](#) a pagina 300.

Vedere anche

[Aggiunta di una funzione interruttore di fine corsa programmabile \(PLS\)](#) a pagina 334

[Assegnazione di variabili e compilazione del programma](#) a pagina 329

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Esempio: come creare un programma High-Speed Counter \(HSC\)](#) a pagina 324

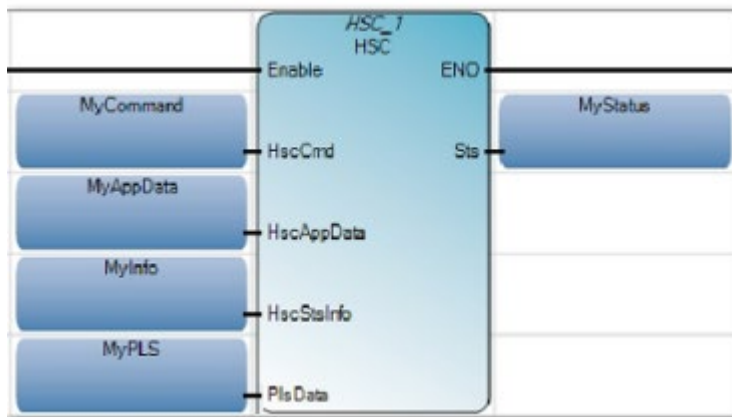
[Tipo di dati HSCAPP](#) a pagina 301

Assegnazione di variabili e compilazione del programma

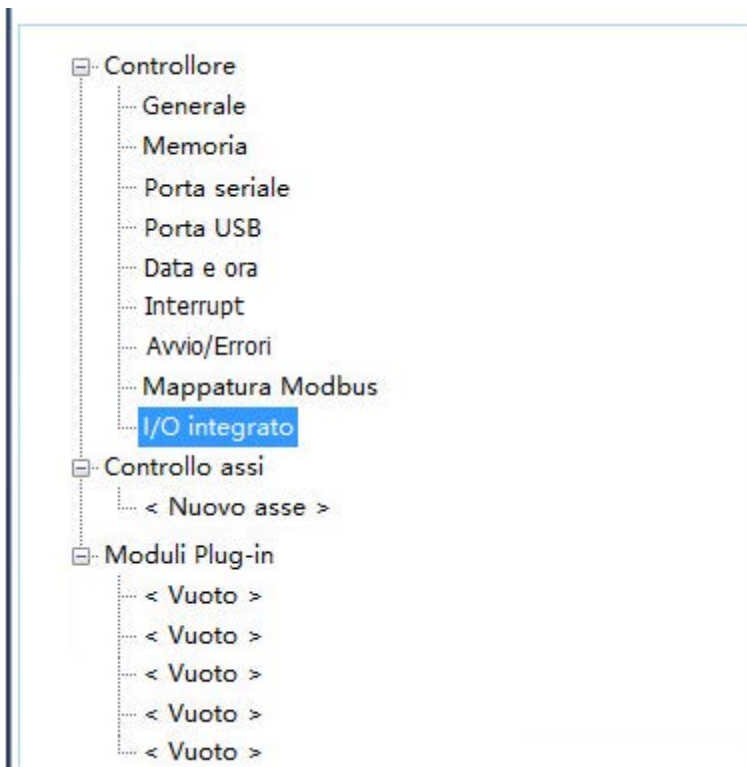
Una volta immessi i valori nelle variabili HSC, attenersi alla seguente procedura per assegnare le variabili al blocco funzione e compilare il programma.

Per assegnare variabili e compilare il programma:

1. Nell'editor di diagramma ladder, assegnare ciascuna variabile all'elemento di blocco funzione HSC come mostrato.



2. Nell'**Organizzatore progetto**, fare clic sul controllore per visualizzare la relativa struttura ad albero.



3. Dalla struttura ad albero del **controllore**, fare clic su **I/O integrato** e selezionare i filtri di ingresso dell'encoder.

Controllore - I/O integrato

Filtro di ingresso		Latch ed EII Edge di ingresso		
Ingressi	Filtro di ingresso	Ingresso	Attiva latch	EII Edge
0-1	Predefinito	0	<input type="checkbox"/>	In discesa
2-3	Predefinito	1	<input type="checkbox"/>	In discesa
4-5	Predefinito	2	<input type="checkbox"/>	In discesa
6-7	Predefinito	3	<input type="checkbox"/>	In discesa
8-9	Predefinito	4	<input type="checkbox"/>	In discesa
10-11	Predefinito	5	<input type="checkbox"/>	In discesa
12-13	Predefinito	6	<input type="checkbox"/>	In discesa
14-15	Predefinito	7	<input type="checkbox"/>	In discesa
16-23	Predefinito	8	<input type="checkbox"/>	In discesa
24-27	Predefinito	9	<input type="checkbox"/>	In discesa
		10	<input type="checkbox"/>	In discesa

4. Verificare che l'encoder sia connesso al controllore Micro830.
5. Avviare il controllore Micro830 e connetterlo al computer.
6. Compilare il programma e scaricarlo nel controllore.

Vedere anche

[Esempio: Come creare un programma per l'High Speed Counter \(HSC\)](#) a pagina 324

[Creazione di un ladder diagram e aggiunta di variabili](#) a pagina 325

[Assegnazione di valori alle variabili HSC](#) a pagina 328

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Aggiunta di una funzione interruttore di fine corsa programmabile \(PLS\)](#) a pagina 334

Test del programma ed esecuzione dell'High-Speed Counter

Una volta scaricato il programma per l'HSC nel controllore, è possibile collaudarlo e quindi eseguire l'High-Speed Counter.

Per collaudare il programma:

1. Connettersi al controllore.
2. Dall'**Organizzatore progetto**, fare doppio clic sul programma HSC, quindi doppio clic su **Variabili locali**.

È possibile visualizzare i valori delle due uscite dell'HSC: STS (MyStatus) e HSCSTS (MyInfo).

3. Fare doppio clic sul contatto diretto **_IO_EM_DI_05** per aprire la finestra **Selettore di variabili**.
4. Fare clic sulla scheda **I/O Micro830**, quindi sulla riga **_IO_EM_DI_05**.
5. Selezionare **Blocco** e **Valore logico** per forzare la posizione **ON** sull'ingresso.

Nome	Alias	Valore logico	Valore fisico	Valore ini	Blocca	Tipo di dati	Dimensio
_IO_EM_DO_00		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DO_01		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DO_02		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DO_03		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DO_04		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DO_05		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DO_06		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DO_07		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DO_08		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DO_09		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DI_00		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DI_01		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DI_02		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DI_03		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DI_04		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DI_05		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	BOOL	-
_IO_EM_DI_06		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DI_07		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DI_08		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-
_IO_EM_DI_09		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	BOOL	-

6. Per visualizzare i risultati, fare clic sulla scheda **Variabili locali** per visualizzare le modifiche alle variabili.
7. Espandere l'elenco variabili **MyAppData** e **MyInfo**.
8. Accendere l'encoder per visualizzare il conteggio crescente/decescente. Ad esempio, se l'encoder è installato in un albero motore, accendere il motore per attivare il conteggio dell'HSC.
9. Verificare che il Valore logico di nella variabile MyStatus sia 1, a indicare che l'HSC è in funzione.
10. Visualizzare il valore del conteggio in MyInfo.Accumulator.
Vedere HSC (High Speed Counter) per l'elenco completo dei codici di stato.

Risultati

In questo esempio, quando MyInfo.Accumulator raggiunge un valore alto preimpostato di 40, l'uscita 0 passa a On e il flag HPRReached

passa a On. Se MyInfo.Accumulator raggiunge un valore basso preimpostato di -40, l'uscita 1 si attiva e il flag LPRached si attivano.

Monitoraggio della variabile

Variabili globali utente - Micro870 Variabili locali - UntitledLD Variabili

Nome	Valore logico	Valore fisico
HSC_1
MyAppData
MyAppData.PlsEnab	<input type="checkbox"/>	N/D
MyAppData.HscID	0	N/D
MyAppData.HscMod	7	N/D
MyAppData.Accumu	40	N/D
MyAppData.HPSettir	40	N/D
MyAppData.LPSettir	-40	N/D
MyAppData.OFSettir	50	N/D
MyAppData.UFSettir	-50	N/D
MyAppData.OutputV	3	N/D
MyAppData.HPOutp	1	N/D
MyAppData.LPOutpi	2	N/D
MyCommand	0	N/D
MyInfo
MyInfo.CountEnable	<input checked="" type="checkbox"/>	N/D
MyInfo.ErrorDetecte	<input type="checkbox"/>	N/D
MyInfo.CountUpFlag	<input checked="" type="checkbox"/>	N/D
MyInfo.CountDownF	<input checked="" type="checkbox"/>	N/D
MyInfo.Mode1Done	<input type="checkbox"/>	N/D
MyInfo.OVF	<input type="checkbox"/>	N/D
MyInfo.UNF	<input type="checkbox"/>	N/D
MyInfo.CountDir	<input checked="" type="checkbox"/>	N/D
MyInfo.HPRached	<input checked="" type="checkbox"/>	N/D
MyInfo.LPRached	<input type="checkbox"/>	N/D
MyInfo.OFCauseInte	<input type="checkbox"/>	N/D
MyInfo.UFCauseInte	<input type="checkbox"/>	N/D
MyInfo.HPCauseInte	<input type="checkbox"/>	N/D
MyInfo.LPCauseInte	<input type="checkbox"/>	N/D
MyInfo.StateInfo	0	N/D
MyInfo.PlsPosition	0	N/D
MyInfo.ErrorCode	0	N/D
MyInfo.Accumulator	40	N/D

Vedere anche

[Aggiunta di una funzione interruttore di fine corsa programmabile \(PLS\)](#) a pagina 334

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Esempio: Come creare un programma per l'High Speed Counter \(HSC\)](#) a pagina 324

[HSC \(High Speed Counter\)](#) a pagina 298

Add a Programmable Limit Switch (PLS) function

L'esempio mostra come aggiungere una funzione Interruttore di fine corsa programmabile (PLS) al programma HSC.

Valori delle variabili per le impostazioni del contatore:

- **MyAppData.PlsEnable** serve per attivare e disattivare le impostazioni PLS. Deve essere impostato a FALSE (disabilitato) se viene utilizzata la variabile MyAppData.
- **MyAppData.HscID** serve per specificare quale input viene utilizzato in base alla modalità e al tipo di applicazione. Vedere gli schemi dei cablaggi e degli input HSC, per riconoscere i diversi ID utilizzabili, così come gli input integrati e le relative caratteristiche.
- Se viene utilizzato ID 0, ID 1 non può essere utilizzato nello stesso controllore, in quanto gli input sono utilizzati da Reset e Hold.
- **MyAppData.HscMode** serve per specificare il tipo di operazione utilizzata da HSC per il conteggio. Vedere la modalità HSC (HSCAPP.HSCMode).

Per abilitare la funzione PLS:

1. In **Organizzatore progetto**, fare doppio clic su **Variabili locali** per visualizzare la pagina **Variabili**.
2. Abilitare la funzione PLS:
 - Nel campo Valore iniziale della variabile MyAppData.PlsEnable selezionare TRUE.
3. Configurare le impostazioni di underflow e overflow:
 - Nel campo Valore iniziale di MyAppData.OFSetting digitare 50.
 - Nel campo Valore iniziale di MyAppData.UFSetting digitare -50.
4. (facoltativo) Configurare la maschera di uscita se si tratta di un'uscita.

I risultati in questo esempio:

- La variabile PLS ha una dimensione di [1..4]. Ciò significa che l'HSC può avere quattro coppie di valori alto e basso preimpostati.

- I valori alti preimpostati devono sempre essere impostati su un valore inferiore rispetto a OFSetting, mentre i valori bassi preimpostati devono essere impostati su un valore superiore rispetto a UFSetting.
- I valori HscHPOutPut e HscLPOutPut determinano le uscite che vengono attivate, al raggiungimento di un valore alto o basso preimpostato.

Nome	Alias	Tipo di dati	Dimensione	Valore predefinito	Valore iniziale
MyPLS		PLS	[1..4]
MyPLS[1]		PLS	
MyPLS[1].HscHP		DINT			10
MyPLS[1].HscLP		DINT			-10
MyPLS[1].HscHPOutPut		UDINT			1
MyPLS[1].HscLPOutPut		UDINT			16
MyPLS[2]		PLS	
MyPLS[2].HscHP		DINT			20
MyPLS[2].HscLP		DINT			-20
MyPLS[2].HscHPOutPut		UDINT			2
MyPLS[2].HscLPOutPut		UDINT			32
MyPLS[3]		PLS	
MyPLS[3].HscHP		DINT			30
MyPLS[3].HscLP		DINT			-30
MyPLS[3].HscHPOutPut		UDINT			4
MyPLS[3].HscLPOutPut		UDINT			64
MyPLS[4]		PLS	
MyPLS[4].HscHP		DINT			40
MyPLS[4].HscLP		DINT			-40
MyPLS[4].HscHPOutPut		UDINT			8
MyPLS[4].HscLPOutPut		UDINT			128

Vedere anche

[Esempio: come creare un programma High-Speed Counter \(HSC\) a pagina 324](#)

[Configurazione di un interruttore di fine corsa programmabile \(PLS\) a pagina 323](#)

Esempio: interruttore fine corsa programmabile (PLS) abilitato

In questo esempio vengono descritti i risultati in caso di funzione PLS abilitata con specifici valori per i parametri HSC e PLSDData.

Valori del parametro HSC

Questo esempio presuppone che i parametri HSC siano impostati sui seguenti valori:

- HSCApp.OutputMask = 31
- HSCApp.HSCMode = 0
- L'HSC controlla solo l'uscita integrata 0...4

Valori del parametro PLSData

Nell'esempio si presuppone che i parametri PLSData relativi alla variabile (HSC_PLS) siano configurati nel modo indicato di seguito.

Nome	Alias	Tipo di dati	Dimensione	Valore progetto	Valore iniziale
HSC_1		HSC	
HSC_PLS		PLS	[1..4]
HSC_PLS[1]		PLS	
HSC_PLS[1].HscHP		DINT			250
HSC_PLS[1].HscLP		DINT			-2
HSC_PLS[1].HscHPOutput		UDINT			3
HSC_PLS[1].HscLPOutput		UDINT			0
HSC_PLS[2]		PLS	
HSC_PLS[2].HscHP		DINT			500
HSC_PLS[2].HscLP		DINT			-2
HSC_PLS[2].HscHPOutput		UDINT			7
HSC_PLS[2].HscLPOutput		UDINT			0
HSC_PLS[3]		PLS	
HSC_PLS[3].HscHP		DINT			750
HSC_PLS[3].HscLP		DINT			-2
HSC_PLS[3].HscHPOutput		UDINT			15
HSC_PLS[3].HscLPOutput		UDINT			0
HSC_PLS[4]		PLS	
HSC_PLS[4].HscHP		DINT			1000
HSC_PLS[4].HscLP		DINT			-2
HSC_PLS[4].HscHPOutput		UDINT			31
HSC_PLS[4].HscLPOutput		UDINT			0

Risultati abilitati dalla PLS

Nell'esempio si verificano i seguenti eventi.

- Alla prima esecuzione della logica ladder: HSCSTS.Accumulator = 1 significa che tutte le uscite sono disattivate.
- Quando HSCSTS.Accumulator = 250, HSC_PLS[1].HscHPOutput è inviato alla maschera HSCAPP.OutputMask e attiva le uscite 0 e 1.
- Si ripete l'invio dell'uscita con valore alto preimpostato alla maschera delle uscite, finché HSCSTS.Accumulator raggiunge 500, 750 e 1000; il controllore attiva rispettivamente le uscite 0...2, 0...3 e 0...4.
- Al termine del funzionamento, il ciclo viene reimpostato e ripetuto da HSCSTS.HP = 250.
- Quando l'intero ciclo è completo e il valore HSCSTS.HP viene raggiunto, il parametro PLSPositon viene ripristinato.

Vedere anche

[Aggiornamento dei dati dell'applicazione HSC](#) a pagina 317

[Configurazione delle interruzioni da parte dell'utente dell'High-Speed Counter \(HSC\)](#) a pagina 318

[Configurazione di un interruttore di fine corsa programmabile \(PLS\)](#) a pagina 323

[Esempio: come creare un programma High-Speed Counter \(HSC\)](#) a pagina 324

[HSC \(High Speed Counter\)](#) a pagina 298

Istruzioni HSCE

Utilizzare le istruzioni HSCE per monitorare e controllare l'High Speed Counter.

Istruzione	Descrizione
HSCE a pagina 337	HSCE: avvio, arresto e lettura del valore dell'accumulatore.
HSCE_CFG a pagina 341	HSCE_CFG è la configurazione dell'High Speed Counter.
HSCE_CFG_PLS a pagina 344	HSCE_CFG_PLS è la configurazione dell'interruttore di fine corsa programmabile (PLS) dell'High Speed Counter.
HSCE_READ_STS a pagina 346	HSCE_READ_STS legge lo stato dell'High Speed Counter.
HSCE_SET_STS a pagina 337	HSCE_SET_STS imposta/segue il reset manuale dello stato dell'High Speed Counter.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

[Tipo di dati HSCE_CHANNEL](#) a pagina 312

[Tipo di dati HSCE_STS](#) a pagina 313

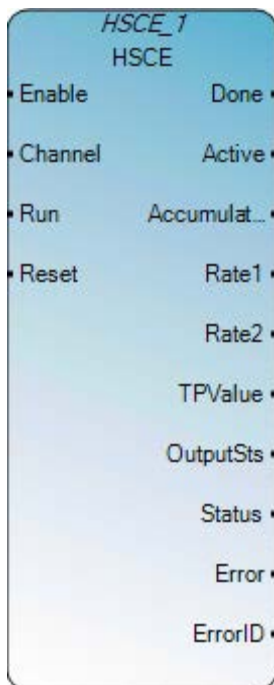
[Tipo di dati PLS_HSCE](#) a pagina 314

HSCE

HSCE è utilizzato per controllare e leggere il contatore HSC.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

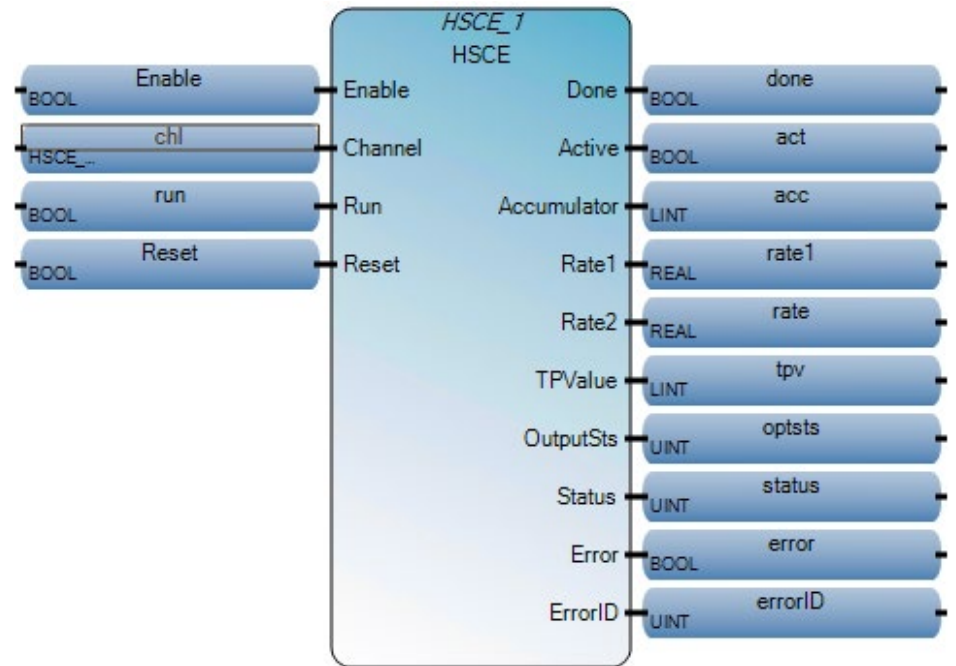
Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.



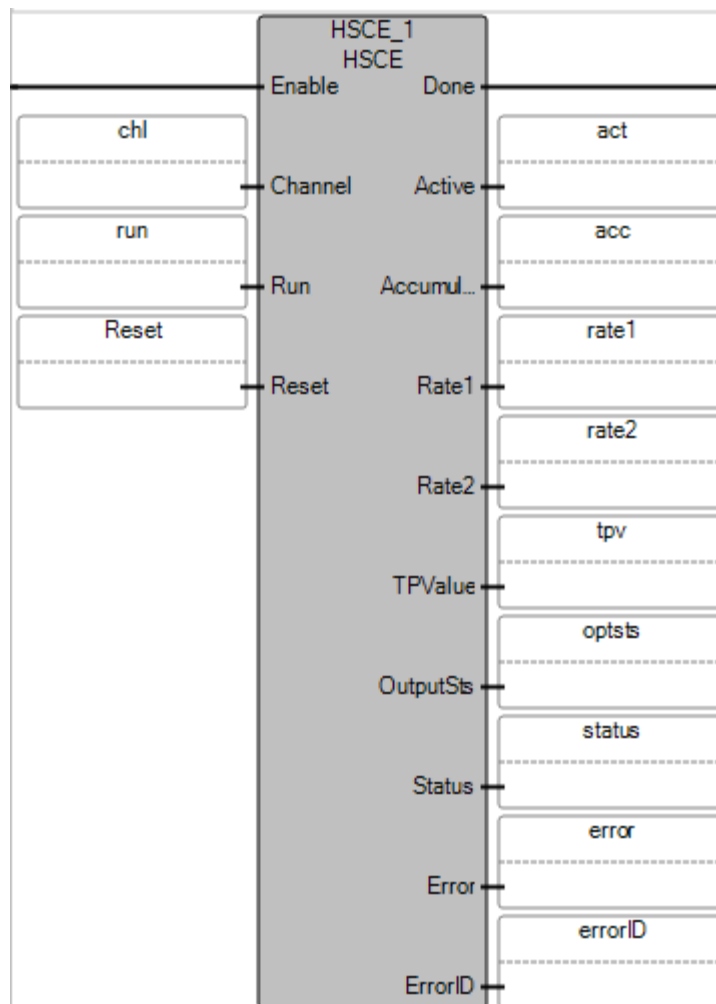
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	TRUE: HSCE avvia il blocco funzione e il modulo plug-in HSC. Accumulatore = InitAcc. Verrà eseguito il reset di HSCE. Rate1 = 0, Rate2 = 0. FALSE: l'accumulatore viene aggiornato con la lettura dal modulo plug-in. Done, Active, Error = FALSE dove ErrorID = 0, Rate1, Rate2 = 0.
Canale	Ingresso	HSCE_CHANNEL	Il canale HSCE.
Run	Ingresso	BOOL	Permette a HSCE di conteggiare lo stato operativo. TRUE: HSCE conteggia gli impulsi. FALSE: HSCE arresta il conteggio.
Reset	Ingresso	BOOL	True: tutte le uscite vengono cancellate e l'accumulatore. Anche Rate1 e Rate2 vengono cancellati e impostati su 0. Cancella lo stato del modulo plug-in HSC. La priorità di Reset input è maggiore del valore immesso per Run
Done	Uscita	BOOL	True: se HSCE Enable è True e non viene rilevato alcun errore False: se HSCE Enable è True ma Run è False.
Attivo	Uscita	BOOL	True: se HSCE Enable è True, Done è True e Run è False. False: se HSCE Enable è True ma Run è False.
Rate1	Uscita	REAL	Frequenza di impulso corrente in unità utente al secondo (metodo Per impulso).
Rate2	Uscita	REAL	Frequenza di impulso corrente in unità utente al secondo (metodo Ciclico).
TPValue	Uscita	LINT	Acquisizione del valore accumulatore quando si attiva il tastatore.
OutputSts	Uscita	UINT	Stato uscita fisica e virtuale del plug-in HSC, applicabile solo al contatore HSC 0. Bit 0: Uscita 0 (stato uscita fisica) Bit da 1 a 15: Da Uscita 1 a Uscita 15 (stato uscita virtuale)
Stato	Uscita	UINT	Informazioni di stato dell'HSC.
Errore	Uscita	BOOL	Indica che si è verificato un errore.
ErrorID	Uscita	UINT	Se si verifica un errore, ErrorID contiene il codice dell'errore.

Esempio di programmazione a blocchi funzionali HSCE



Esempio di diagramma ladder HSCE



Esempio di testo strutturato HSCE

```
HSCE_1(
void HSCE_1(BOOL Enable, HSCE_CHANNEL Channel, BOOL Run, BOOL Reset)
Tipo: HSCE, Avvia, interrompi e leggi il valore dell'accumulatore
```

```

1  HSCE_1(Enable, chl, Run, Reset);
2  Done_HSCE := HSCE_1.Done;
3  Active_HSCE := HSCE_1.Active;
4  Accumulator_HSCE := HSCE_1.Accumulator;
5  Rate1_HSCE := HSCE_1.Rate1;
6  Rate2_HSCE := HSCE_1.Rate2;
7  TPValue_HSCE := HSCE_1.TPValue;
8  Outputsts_HSCE := HSCE_1.OutputSts;
9  Status_HSCE := HSCE_1.Status;
10 Error_HSCE := HSCE_1.Error;
11 ErrorID_HSCE := HSCE_1.ErrorID;
```


Vedere anche

[HSCE_CFG](#) a pagina 341

[HSCE_CFG_PLS](#) a pagina 344

[HSCE_READ_STS](#) a pagina 346

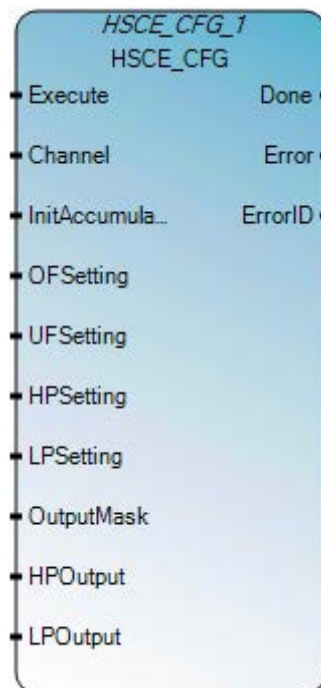
[HSCE_SET_STS](#) a pagina 348

HSCE_CFG

HSCE_CFG è utilizzato per configurare l'High Speed Counter.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.

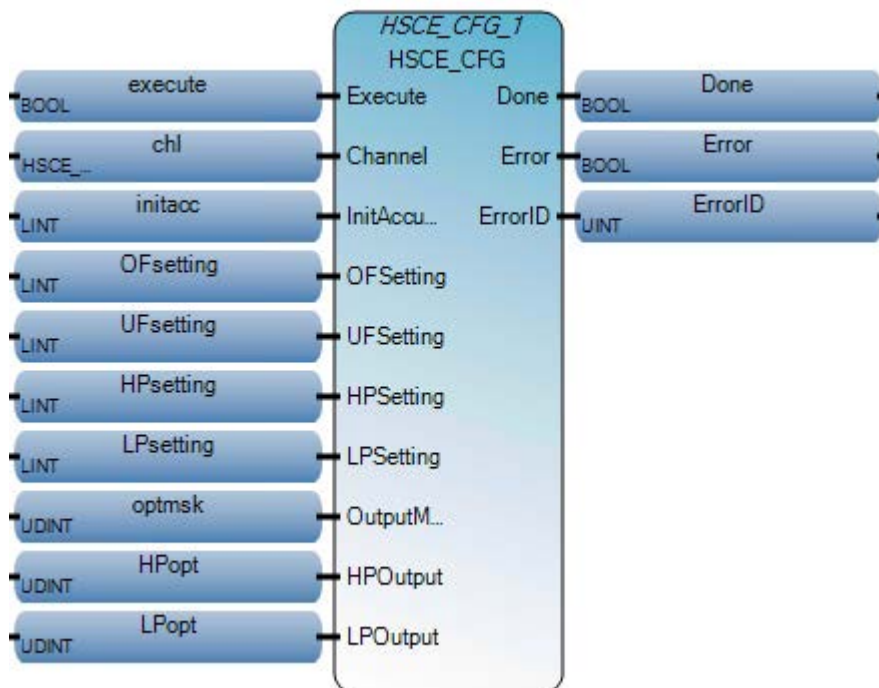


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

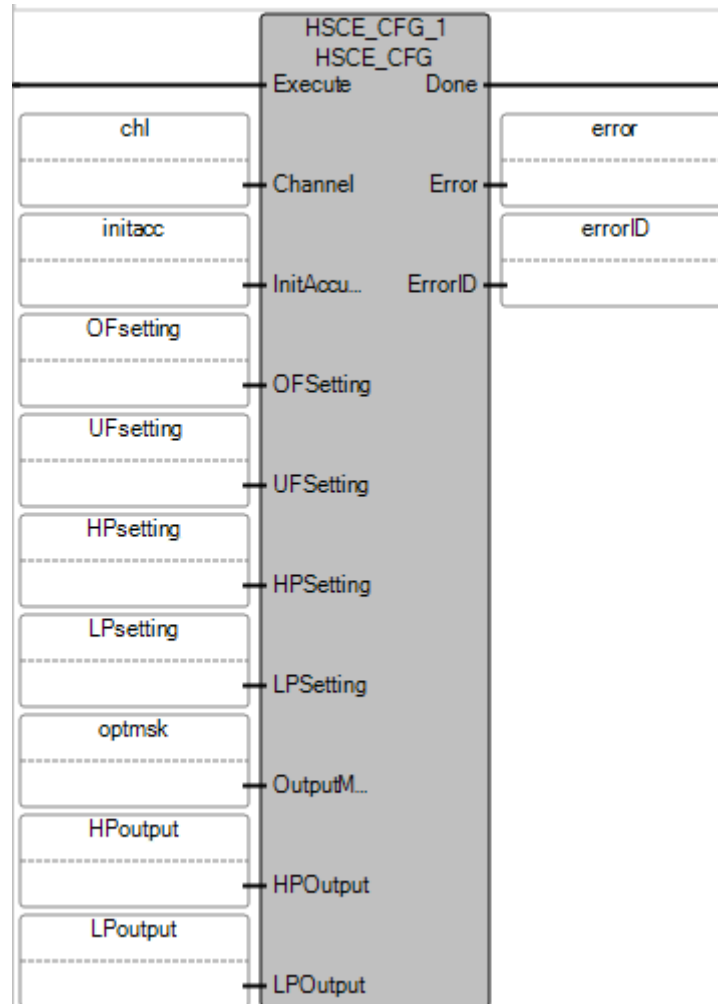
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Il fronte di salita avvia la configurazione dell'HSC. (HSCE Enable deve essere uguale a FALSE) Il fronte di discesa cancellerà tutti i valori di uscita.
Canale	Ingresso	HSCE_CHANNE L	Il canale HSCE.
InitAccumulator	Ingresso	LINT	Valore iniziale accumulatore.
OFSetting	Ingresso	LINT	Valore limite per overflow contatore.
UFSetting	Ingresso	LINT	Valore limite per underflow contatore.
HPSetting	Ingresso	LINT	Valore alto preimpostato (HP) di HSCE.
LPSetting	Ingresso	LINT	Valore basso preimpostato (LP) di HSCE.
PLS_Offset	Ingresso	USINT	Offset per l'avvio nel vettore dati PLS.

OutputMask	Ingresso	USINT	Maschera di uscita per la funzionalità PLS.
HPOutput	Ingresso	UDINT	Stato uscite valore alto preimpostato.
LPOutput	Ingresso	UDINT	Stato uscite valore basso preimpostato.
Done	Uscita	BOOL	Azione di configurazione dell'HSC (avviata da questa istruzione) eseguita correttamente.
Errore	Uscita	BOOL	Indica che si è verificato un errore.
ErrorID	Uscita	UINT	Se si verifica un errore, ErrorID contiene il codice dell'errore.

Esempio di programmazione a blocchi funzionali HSCE_CFG



Esempio di diagramma ladder HSCE_CFG



Esempio di testo strutturato HSCE_CFG

```

HSCE_CFG_1
void HSCE_CFG_1(BOOL Execute, HSCE_CHANNEL Channel, LINT InitAccumulator, LINT OFSetting, LINT UFSetting, LINT HPSetting, LINT LPSetting, UDINT OutputMask, UDINT HPOutput, UDINT LPOutput)
Tipo: HSCE_CFG, Configurazione High Speed Counter

1 | HSCE_CFG_1(Execute, chl, initacc, OFsetting, UFsetting, HPsetting, LPsetting, optmsk, HPOpt, LPOpt) :
2 | Done_HSCE_CFG :=HSCE_CFG_1.Done;
3 | Error_HSCE_CFG :=HSCE_CFG_1.Error;
4 | ErrorID_HSCE_CFG := HSCE_CFG_1.ErrorID;

```

Vedere anche

[Istruzioni HSCE](#) a pagina 337

[HSCE](#) a pagina 337

[HSCE_CFG_PLS](#) a pagina 344

[HSCE_READ_STS](#) a pagina 346

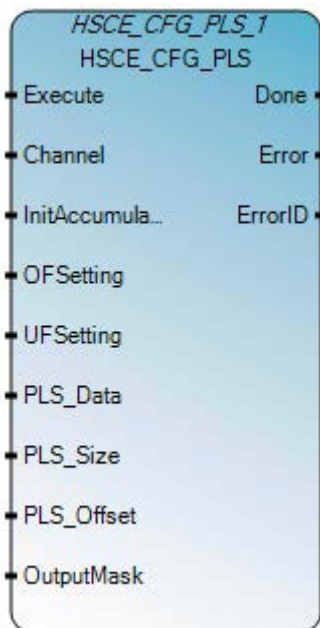
[HSCE_SET_STS](#) a pagina 348

HSCE_CFG_PLS

Utilizzare questa istruzione per la configurazione di HSC con interruttore di fine corsa programmabile (PLS). Questa funzione fornisce un insieme aggiuntivo di modalità operative per l'High Speed Counter. In queste modalità i valori dei dati preimpostati di uscita vengono aggiornati usando i dati forniti dall'utente ogni volta che viene raggiunto uno dei valori preimpostati. Per programmare queste modalità, utilizzare un file PLS con i set di dati da usare.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

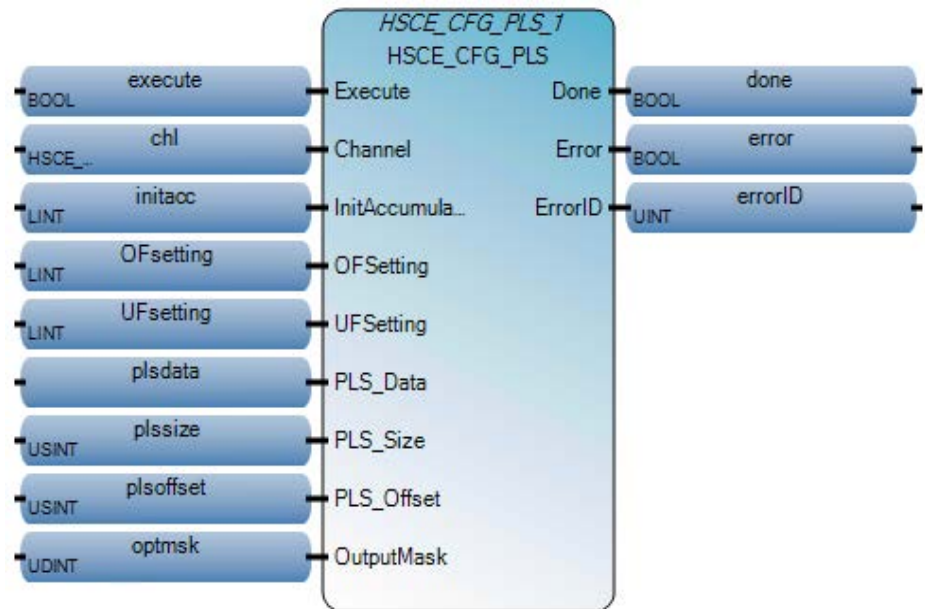
Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.



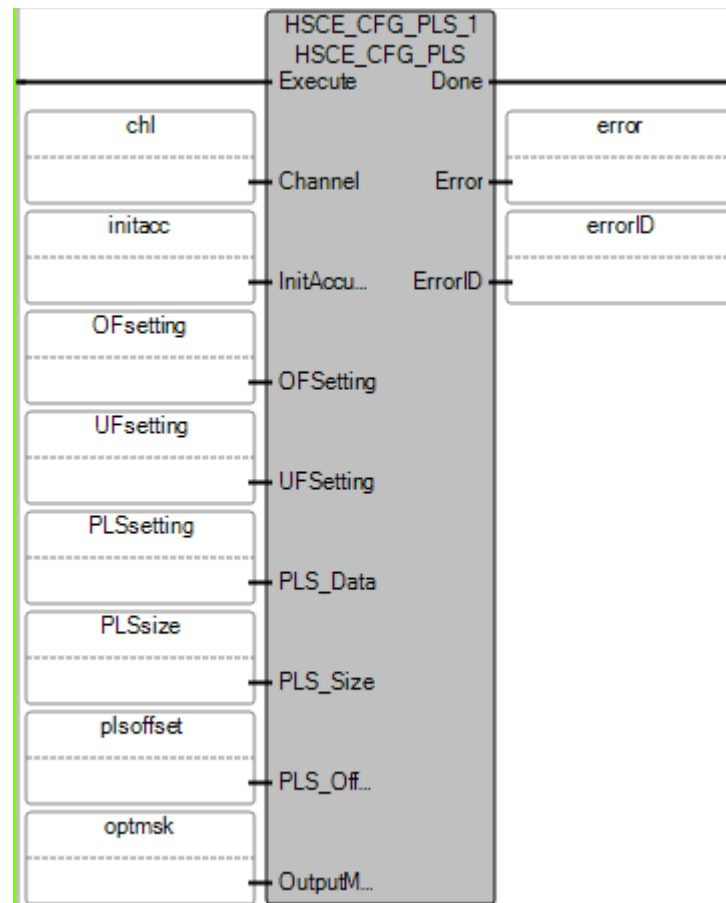
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Il fronte di salita avvia la configurazione dell'HSC. (HSCE Enable deve essere uguale a FALSE) Il fronte di discesa cancellerà tutti i valori di uscita.
Canale	Ingresso	HSCE_CHANNEL	Il canale HSCE.
InitAccumulator	Ingresso	LINT	Valore iniziale accumulatore.
OFSetting	Ingresso	LINT	Valore limite per overflow contatore.
UFSetting	Ingresso	LINT	Valore limite per underflow contatore.
PLS_Data	Ingresso	PLS2	Vettore di PLS (PLS_64)
PLS_Size	Ingresso	USINT	Dimensioni dei dati PLS, con valore massimo di 24 per il plug-in.
PLS_Offset	Ingresso	USINT	Offset per l'avvio nel vettore dati PLS.
OutputMask	Ingresso	USINT	Maschera di uscita per la funzionalità PLS.
Done	Uscita	BOOL	Azione di configurazione dell'HSC (avviata da questa istruzione) eseguita correttamente.
Errore	Uscita	BOOL	Indica che si è verificato un errore.
ErrorID	Uscita	UINT	Se si verifica un errore, ErrorID contiene il codice dell'errore.

Esempio di programmazione a blocchi funzionali HSCE_CFG_PLS



Esempio di diagramma ladder HSCE_CFG_PLS



Esempio di testo strutturato HSCE_CFG_PLS

```

HSCE_CFG_PLS_1
void HSCE_CFG_PLS_1(BOOL Execute, HSCE_CHANNEL Channel, LINT InitAccumulator, LINT OFSetting, LINT UFSetting, PLS_HSCE[1..1] PLS_Data, USINT PLS_Size, USINT PLS_Offset, UDINT OutputMask)
Tip: HSCE_CFG_PLS, Configurazione PLS High Speed Counter

1 HSCE_CFG_PLS_1(Execute, chl, initacc, OFsetting, UFsetting, PLS_data, PLS_Size, PLS_Offset, optmsk);
2 Done_HSCE_CFG_PLS :=HSCE_CFG_PLS_1.Done;
3 Error_HSCE_CFG_PLS :=HSCE_CFG_PLS_1.Error;
4 ErrorID_HSCE_CFG_PLS := HSCE_CFG_PLS_1.ErrorID;
    
```

Vedere anche

[Istruzioni HSCE](#) a pagina 337

[HSCE](#) a pagina 337

[HSCE_CFG](#) a pagina 341

[HSCE_READ_STS](#) a pagina 346

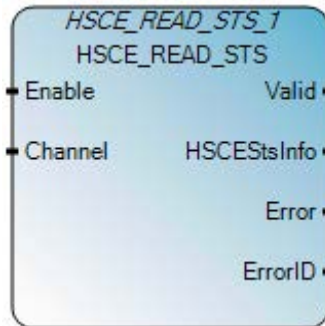
[HSCE_SET_STS](#) a pagina 348

HSCE_READ_STS

L'istruzione viene usata per leggere lo stato corrente dell'HSC.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

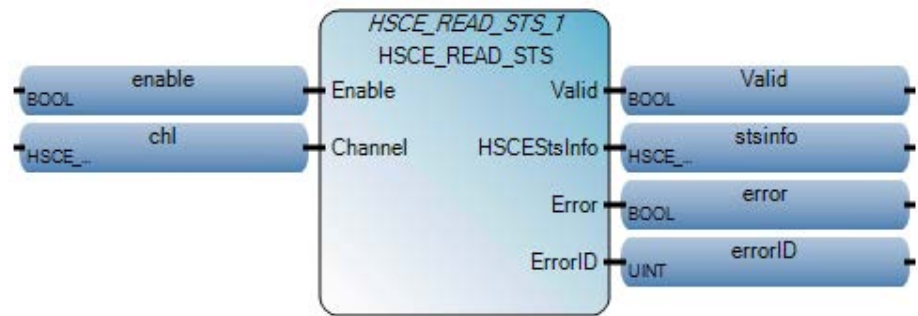
Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.



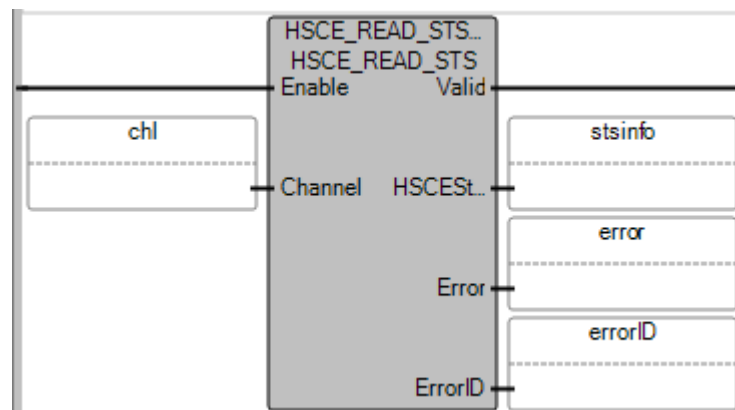
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Se Enable è True, HSC2StsInfo viene aggiornato.
Canale	Ingresso	HSCE_CHANNEL	Il canale HSCE.
Valid	Uscita	BOOL	HSC2StsInfo è Valid se TRUE.
HSC2StsInfo	Ingresso	LINT	Valore limite per overflow contatore.
Errore	Uscita	BOOL	Indica che si è verificato un errore.
ErrorID	Uscita	UINT	Se si verifica un errore, ErrorID contiene il codice dell'errore.

Esempio di programmazione a blocchi funzionali HSCE_READ_STS



Esempio di diagramma ladder HSCE_READ_STS



Esempio di testo strutturato HSCE_READ_STS

```

HSCE_READ_STS_1(
void HSCE_READ_STS_1(BOOL Enable, HSCE_CHANNEL Channel)
Tipo: HSCE_READ_STS, Leggi stato High Speed Counter

1| HSCE_READ_STS_1(Enable,Channel);
2| Valid_HSCE_READ_STS :=HSCE_READ_STS_1.Valid;
3| HSCEstsInfo_HSCE_READ_STS := HSCE_READ_STS_1.HSCEstsInfo;
4| Error_HSCE_READ_STS := HSCE_READ_STS_1.Error;
5| ErrorID_HSCE_READ_STS := HSCE_READ_STS_1.ErrorID;

```

Vedere anche

[Istruzioni HSCE](#) a pagina 337

[HSCE](#) a pagina 337

[HSCE_CFG](#) a pagina 341

[HSCE_CFG_PLS](#) a pagina 344

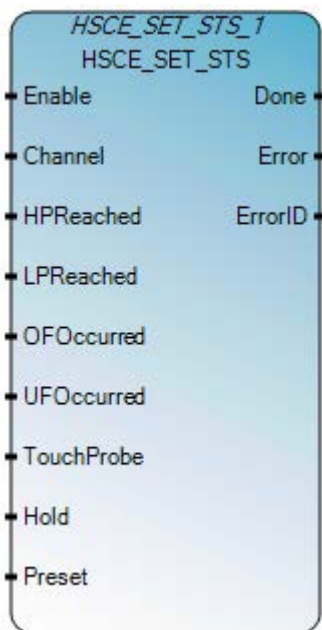
[HSCE_SET_STS](#) a pagina 348

HSCE_SET_STS

L'istruzione consente l'impostazione e il reset manuali dei flag di stato nel conteggio dell'HSC. Il blocco funzione dell'HSC deve essere interrotto per consentire l'impostazione o il reset dello stato HTS da parte del blocco funzione HSCE_SET_STC. Se non si interrompe il blocco funzione dell'HSC o non si configura il canale HSC, il blocco funzione HSC2_SET_STS genererà un errore.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.

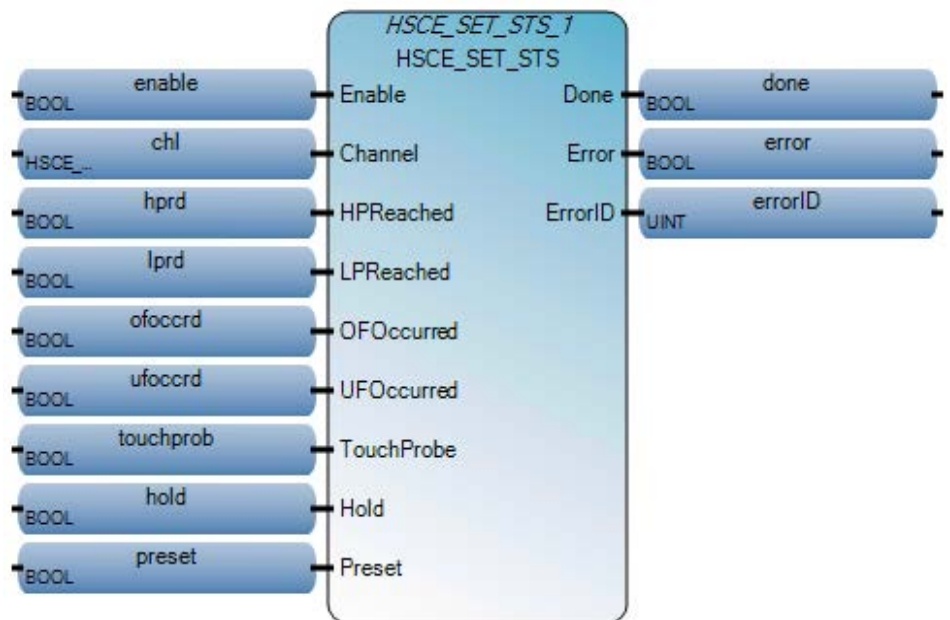


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

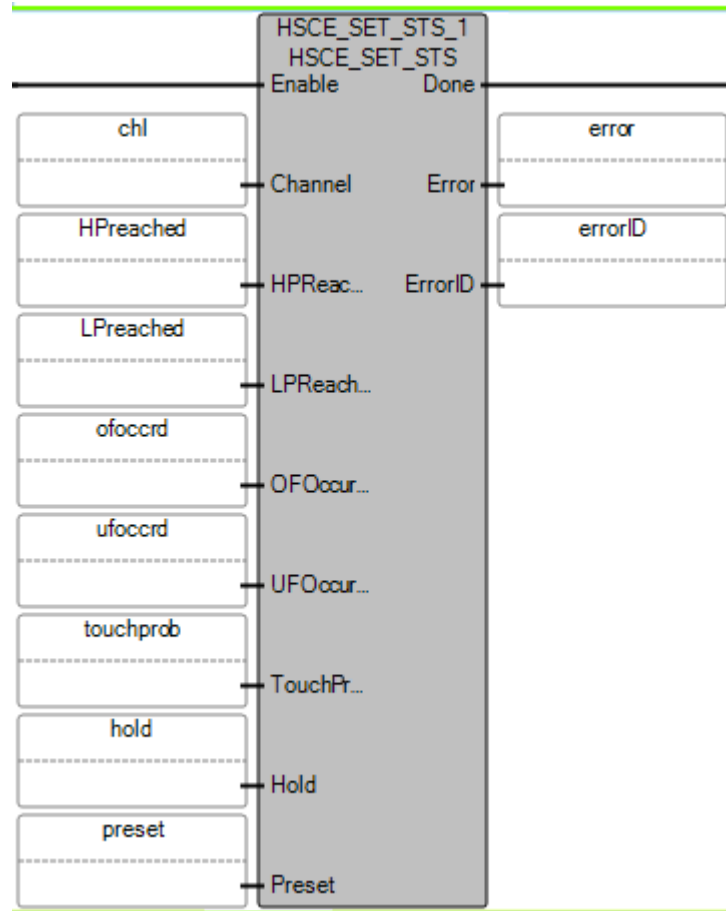
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilita Blocco funzione. TRUE: imposta/segue il reset dello stato dell'HSC. FALSE: nessuna modifica allo stato HSC.
Canale	Ingresso	HSCE_CHANNEL	Il canale HSCE.
HPRReached	Uscita	BOOL	Quando si raggiunge il valore alto preimpostato, questo bit viene impostato su TRUE dal modulo plug-in. Impostare o reimpostare questo bit tramite il blocco funzione HSCE_SET_STS.
LPReached	Ingresso	BOOL	Quando si raggiunge il valore basso preimpostato, questo bit viene impostato su TRUE dal modulo plug-in. Impostare o reimpostare questo bit tramite il blocco funzione HSCE_SET_STS.
OFOccurred	Ingresso	BOOL	Quando si verifica un overflow, questo bit viene impostato su TRUE dal modulo plug-in. Impostare o eseguire il reset di questo bit tramite il blocco funzione HSCE_SET_STS.
UFOccurred	Ingresso	BOOL	Quando si verifica un underflow, questo bit viene impostato su TRUE dal modulo plug-in. Impostare o reimpostare questo bit tramite il blocco funzione HSCE_SET_STS.

TouchProbe	Ingresso	BOOL	Quando si attiva il tastatore, questo bit viene impostato su TRUE dal modulo plug-in.
Conserva	Ingresso	BOOL	Quando si attiva il mantenimento dell'HSC, questo bit viene impostato su TRUE dal modulo plug-in. Impostare o eseguire il ripristino di questo bit tramite il blocco funzione HSCE_SET_STS. Questo parametro di ingresso è valido solo per il contatore 0. Per il contatore 1 è sempre Reset
Preimposta	Ingresso	BOOL	Quando si attiva Z ACC Reset, questo bit viene impostato su TRUE dal modulo plug-in. Impostare o reimpostare questo bit tramite il blocco funzione HSCE_SET_STS. Questo parametro di ingresso è valido solo per il contatore 0. Per il contatore 1 è sempre Reset.
Done	Uscita	BOOL	Azione di configurazione dell'HSC (avviata da questa istruzione) eseguita correttamente.
Errore	Uscita	BOOL	Indica che si è verificato un errore.
ErrorID	Uscita	UINT	Se si verifica un errore, ErrorID contiene il codice dell'errore.

Esempio di programmazione a blocchi funzionali HSCE_SET_STS



Esempio di diagramma ladder HSCE_SET_STS



Esempio di testo strutturato HSCE_SET_STS

```

HSCE_SET_STS_1
void HSCE_SET_STS_1(BOOL Enable, HSCE_CHANNEL Channel, BOOL HPreached, BOOL LPreached, BOOL OFOccurred, BOOL UFOccurred, BOOL TouchProbe, BOOL Hold, BOOL Preset)
Tipo: HSCE_SET_STS, Imposta/ripristina manualmente lo stato High Speed Counter

1 HSCE_SET_STS_1(Enable, chl, hprd, ofoccrd, ufoccrd, touchprob, hold, preset);
2 Done_HSCE_SET_STS := HSCE_SET_STS_1.Done;
3 Error_HSCE_SET_STS := HSCE_SET_STS_1.Error;
4 ErrorID_HSCE_SET_STS := HSCE_SET_STS_1.ErrorID;
    
```

Vedere anche

- [Istruzioni HSCE](#) a pagina 337
- [HSCE](#) a pagina 337
- [HSCE_CFG](#) a pagina 341
- [HSCE_CFG_PLS](#) a pagina 344
- [HSCE_READ_STS](#) a pagina 346

Codici di errore HSCE

Nella tabella seguente sono descritti i codici di errore dello stato per le istruzioni HSCE:

Codice ErrorID	Descrizione errore	Azione correttiva
0	L'istruzione ha completato correttamente l'operazione.	
1	File di configurazione HSC non validi.	Contattare il rappresentante del supporto tecnico di Rockwell Automation. Per le informazioni sui contatti, visitare: http://www.rockwellautomation.com/support
2	Tipo di modulo HSC non valido.	Correggere il tipo di modulo. Ad esempio, selezionare il tipo di modulo come plug-in.
3	ID slot HSC non valido.	Correggere l'ID slot nel canale di ingresso del blocco funzione.
4	ID HSC non valido.	Correggere l'ID HSC. Ad esempio, impostare 0 per contatore 0.
5	Modalità non valida per il canale.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
6	Dimensione PLS non valida.	La dimensione PLS deve essere ≤ 24 e la dimensione del vettore dati PLS deve essere \geq rispetto alla dimensione PLS.
7	Offset PLS non valido.	Offset PLS + Dimensione PLS deve essere inferiore alla dimensione del vettore dati PLS.
8	Valore InitAccumulator non valido.	Correggere il valore InitAccumulator. È possibile che ecceda il limite o sia esterno all'intervallo del limite LP o HP ($LP \leq \text{InitACC} \leq HP$).
9	LP non valido.	Correggere il valore LP. Può eccedere il limite.
10	OF non valido.	Correggere il valore OF.
11	UF non valido.	Correggere il valore UF.
12	HP non valido.	Correggere il valore HP.
13	Nessuna configurazione per HSCE.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
14	Stato HSCE non valido.	Controllare il blocco funzione HSCE correlato per confermare lo stato di questo blocco funzione.
15	Modulo plug-in non valido	Controllare il modulo plug-in per confermare che è un modulo HSC.
16	HSCE è in esecuzione.	Quando HSC è in fase di conteggio, non è consentito alcuno stato di configurazione e impostazione. HSCE_CFG e HSCE_CFG_PLS vengono eseguiti mentre HSCE è in esecuzione.
17	Errore arresto plug-in HSC.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
18	Errore aggiornamento velocità di scrittura.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
19	Errore scrittura del numero di più.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
20	Errore scrittura Applicazione ACC.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
21	Errore scrittura ACC.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
22	Errore scrittura Applicazione.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
23	Errore scrittura numero di PLS.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
24	Errore scrittura under flow.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
25	Errore scrittura over flow.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
26	Errore scrittura valore basso preimpostato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
27	Errore scrittura valore alto preimpostato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
28	Errore scrittura uscita valore basso preimpostato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
29	Errore scrittura uscita valore alto preimpostato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
30	Errore scrittura maschera di uscita.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
31	Errore scrittura uscita valore basso preimpostato PLS.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
32	Errore scrittura uscita valore alto preimpostato PLS.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
33	Errore scrittura valore basso preimpostato PLS.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
34	Errore scrittura uscita valore alto preimpostato PLS.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
35	Errore scrittura offset PLS.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.

Codice ErrorID	Descrizione errore	Azione correttiva
36	Errore scrittura numero PLS.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
37	Errore lettura stato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
38	Errore lettura valore alto preimpostato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
39	Errore lettura valore basso preimpostato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
40	Errore lettura ACC.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
41	Errore lettura larghezza numero di impulsi.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
42	Errore lettura larghezza impulso.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
43	Errore lettura numero di impulsi.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
44	Errore lettura velocità di aggiornamento.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
45	Errore scrittura stato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
46	Errore lettura uscita valore basso preimpostato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
47	Errore lettura uscita valore alto preimpostato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
48	Errore lettura numero PLS.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
49	N.D.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
50	Errore scrittura stato Applica.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
51	Canale di ingresso non valido.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
52	Errore lettura tastatore.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
53	Errore scrittura ripristino ACC.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
54	Errore avvio HSC.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
55	Contatore disabilitato.	Il contatore HSC è disabilitato. Controllare la configurazione canale per verificare se il contatore HSC è abilitato.
56	Valore maschera di uscita non valido.	Controllare se la maschera di uscita è interna all'intervallo valido. Per modulo HSC plug-in, l'intervallo è 0-65535.
57	Uscita valore alto preimpostato non valida.	Controllare se l'uscita HP è interna all'intervallo valido. Per modulo HSC plug-in, l'intervallo è 0-65535.
58	Uscita valore basso preimpostato non valida.	Controllare se l'uscita LP è interna all'intervallo valido. Per modulo HSC plug-in, l'intervallo è 0-65535.
59	Revisione UPM non supportata.	Controllare la configurazione revisione per il modulo plug-in HSC.
60	Nessun modulo HSC aggiunto.	Controllare se il modulo plug-in effettivo è un modulo HSC.
61	Il modulo plug-in HSCE non è configurato.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
62	Un errore di scrittura plug-in UPM si verifica durante la modifica della modalità plug-in HSC in modalità Non in esecuzione.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
63	Un errore di scrittura plug-in UPM si verifica durante la modifica della modalità plug-in HSC in modalità Esecuzione.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.
64	Un errore lettura plug-in UPM si verifica durante la lettura dello stato di uscita.	Contattare il rappresentante del supporto tecnico di Rockwell Automation.

Vedere anche

[HSCE](#) a pagina 337

[HSCE_CFG](#) a pagina 341

[HSCE_CFG_PLS](#) a pagina 344

[HSCE_READ_STS](#) a pagina 346

[HSCE_SET_STS](#) a pagina 348

Istruzioni ingresso/uscita

Utilizzare le istruzioni ingresso/uscita per leggere o scrivere i dati verso o da un controllore oppure un modulo utilizzando i segnali inviati verso un dispositivo fisicamente connesso a un controllore logico programmabile. I relè ingresso trasferiscono i segnali ai relè interni, mentre i relè uscita trasferiscono i segnali ai dispositivi di uscita esterni.

Istruzione	Descrizione
LCD a pagina 356	Solo Micro810. Visualizza una stringa o un numero su una schermata LCD.
LCD_BKLT_REM a pagina 358	Imposta i parametri di retroilluminazione dell'LCD remoto all'interno di un programma utente.
LCD_REM a pagina 360	Visualizza i messaggi definiti dall'utente per l'LCD remoto.
RHC a pagina 364	Legge il valore high speed clock nel controllore Micro800.
RPC a pagina 365	Legge il checksum del programma dell'utente dal controllore o dal modulo di memoria.
DLG a pagina 366	Scrive i valori di variabile dal motore di runtime in un file di registrazione dati su una scheda SD.
IIM a pagina 368	Aggiorna gli ingressi prima della normale scansione in uscita.
IOM a pagina 370	Aggiorna le uscite prima della normale in uscita.
KEY_READ a pagina 372	Solo Micro810. Legge lo stato della chiave su un modulo LCD opzionale quando il display dell'utente è attivo.
KEY_READ_REM a pagina 375	Solo Micro820. Legge lo stato della chiave su un modulo LCD remoto opzionale quando il display dell'utente è attivo.
MM_INFO a pagina 377	Legge le informazioni sull'intestazione del modulo di memoria.
MODULE_INFO a pagina 380	Legge le informazioni di modulo da un modulo plug-in o di espansione ad eccezione del modulo di memoria 2080-MEMBAK-RTC.
PLUGIN_INFO a pagina 392	Legge le informazioni di modulo da un modulo plug-in generico o di espansione ad eccezione del modulo di memoria 2080-MEMBAK-RTC.
PLUGIN_READ a pagina 394	Legge i dati da un modulo plug-in generico ad eccezione del modulo di memoria 2080-MEMBAK-RTC.
PLUGIN_RESET a pagina 397	Esegue il reset dell'hardware per un modulo plug-in generico ad eccezione del modulo di memoria 2080-MEMBAK-RTC.
PLUGIN_WRITE a pagina 398	Scrive i dati su un modulo plug-in generico ad eccezione del modulo di memoria 2080-MEMBAK-RTC.
RCP a pagina 400	Legge e scrive dati Ricetta su e da una scheda di memoria SD.
RTC_READ a pagina 403	Legge le informazioni sul modulo real-time clock (RTC).
RTC_SET a pagina 405	Imposta i dati RTC sulle informazioni del modulo RTC.
SYS_INFO a pagina 407	Legge il blocco dati dello stato per il controllore Micro800.
TRIMPOT_READ a pagina 410	Legge il valore TrimPot da un trimpot specifico.

Vedere anche

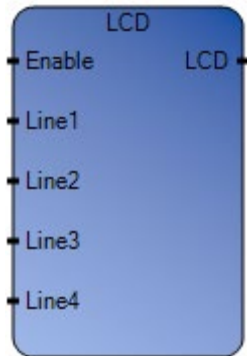
[Istruzione impostate in ordine alfabetico](#) a pagina 18

LCD

Visualizza una stringa o un numero sullo schermo LCD opzionale.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

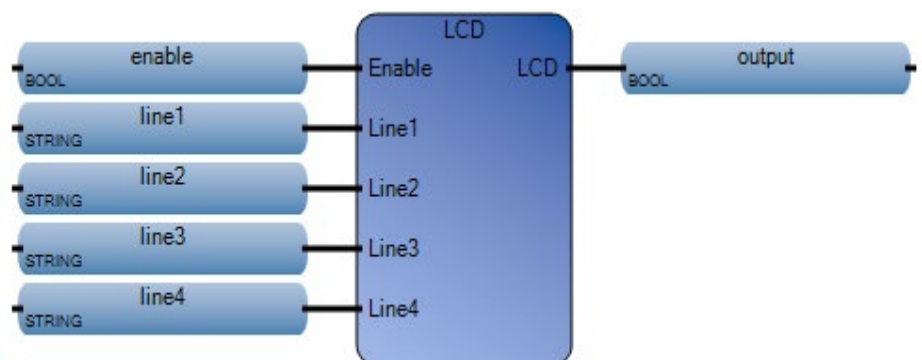
Questa istruzione vale per i controllori Micro810.



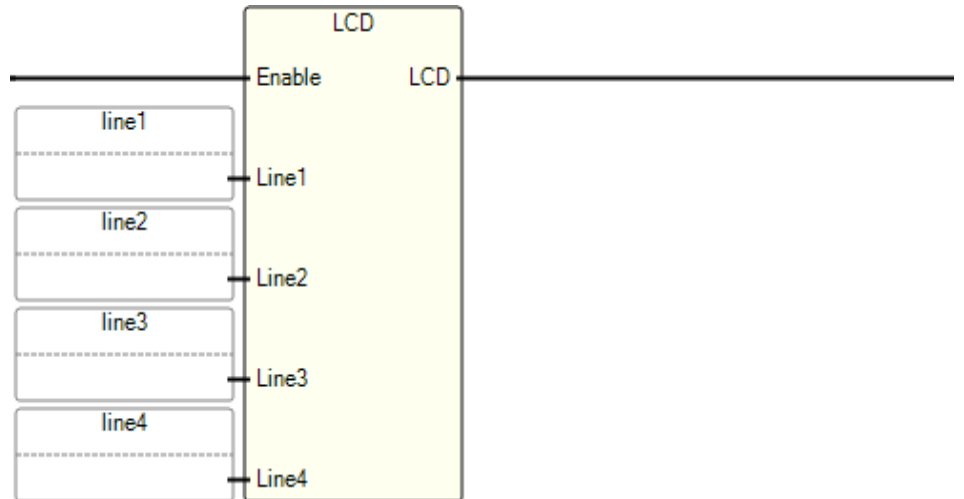
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione istruzioni. TRUE: l'LCD passa allo schermo definito dall'utente (le stringhe vengono visualizzate sullo schermo LCD) invece che alla schermata di stato I/O. FALSE: l'LCD visualizza il contenuto della schermata di stato I/O.
Line1	Ingresso	STRING	Stringa visualizzare alla riga 1 dell'LCD.
Line2	Ingresso	STRING	Stringa visualizzare alla riga 2 dell'LCD.
Line3	Ingresso	STRING	Stringa visualizzare alla riga 3 dell'LCD.
Line4	Ingresso	STRING	Stringa visualizzare alla riga 4 dell'LCD.
LCD	Uscita	BOOL	TRUE: la funzione è abilitata.

Esempio di programmazione a blocchi funzionali LCD



Esempio di Diagramma ladder LCD



Esempio di testo strutturato LCD

```
LCD (
  BOOL LCD(BOOL Enable, STRING Line1, STRING Line2, STRING Line3, STRING Line4)
  Visualizzare la stringa o il numero in base ai requisiti utente se è installato l'LCD opzionale.
```

```
1 enable := TRUE;
2 line1 := 'R';
3 line2 := 'O';
4 line3 := 'C';
5 line4 := 'K';
6 output := LCD(enable, line1, line2, line3, line4);
```

(* Equivalenza ST: *)

```
TESTOUTPUT := LCD(LCDENABLE, LINE1, LINE2, LINE3, LINE4);
```

Risultati

Name	Logical Value	Physical Value	Lock	Data Type
enable	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
line1	R	N/A	<input type="checkbox"/>	STRING
line2	O	N/A	<input type="checkbox"/>	STRING
line3	C	N/A	<input type="checkbox"/>	STRING
line4	K	N/A	<input type="checkbox"/>	STRING
output	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

[KEY_READ](#) a pagina 372

LCD_BKLT_REM (retroilluminazione LCD remoto)

Imposta i parametri di retroilluminazione dell'LCD remoto all'interno del programma utente.

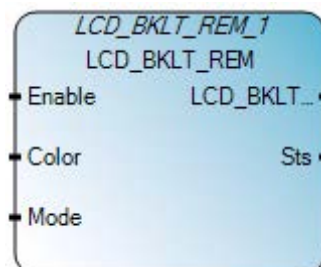
Dettagli operazione:

Le impostazioni di retroilluminazione definite in LCD_BKLT_REM vengono utilizzate quando il display LCD remoto è:

- Una schermata definita dall'utente mediante LCD_REM.
- La schermata predefinita dello stato I/O.
- Per tutte le altre schermate, le impostazioni di retroilluminazione utilizzate sono quelle definite per mezzo dei menu disponibili nell'LCD remoto.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

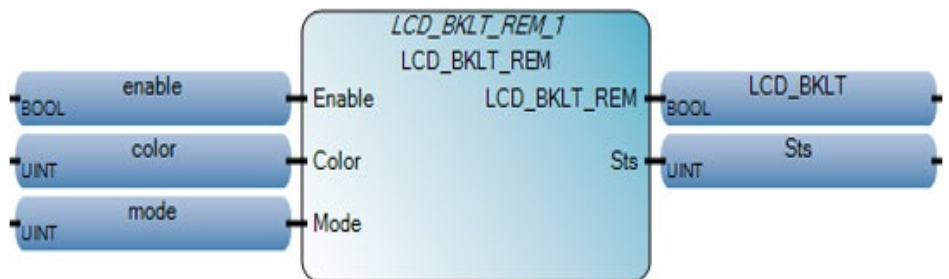
Questa istruzione vale per i controllori Micro820.



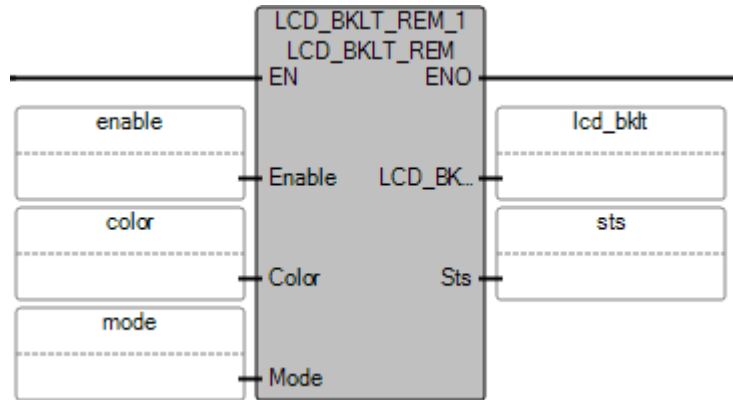
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue REM_LCD_BKLT, le eventuali impostazioni di retroilluminazione correnti vengono sovrascritte. FALSE: REM_LCD_BKLT viene disattivato e le impostazioni di menu LCD remoto diventano effettive.
Colore	Ingresso	UINT	Codice colore retroilluminazione <ul style="list-style-type: none"> • 0: Bianco • 1: Blu • 2: Rosso • 3: Verde • 4-65535: Riservato
Modalità	Ingresso	UINT	<ul style="list-style-type: none"> • 0: Definitivamente OFF • 1: Definitivamente ON • 2: Lampeggiamento (intervallo 1 sec) • 3-65535: Riservato
LCD_BKLT_REM	Uscita	BOOL	TRUE: istruzione eseguita con successo. FALSE: si è verificato un errore durante l'esecuzione dell'istruzione.
Sts	Uscita	UINT	Stato dell'operazione dell'LCD remoto. Codici di stato LCD_BKLT_REM: <ul style="list-style-type: none"> • 0: ingresso abilitazione è False. • 1: riuscito. • 2: LCD remoto non rilevato. Potrebbe verificarsi quando: <ul style="list-style-type: none"> • L'LCD remoto non è connesso fisicamente al controllore oppure il cablaggio non è corretto. • Le impostazioni della porta seriale sono diverse rispetto a quelle richieste per l'LCD remoto. • 3: errore di connessione. Potrebbe verificarsi quando è presente un errore interno dello stato macchina, come un'incompatibilità tra la versione FW del controllore e la versione FW RLCD. • 4: codice colore non valido. • 5: modalità non valida. • 6-65535: riservato.

Esempi di programmazione a blocchi funzionali LCD_BKLT_REM



Esempio di diagramma ladder LCD_BKLT_REM



Esempio di testo strutturato LCD_BKLT_REM

```
LCD_BKLT_REM_4(
void LCD_BKLT_REM_4(BOOL Enable, UINT Color, UINT Mode)
Tipo: LCD_BKLT_REM, LCD remoto: imposta parametri di retroilluminazione
```

```
1 LCD_BKLT_REM_1 (EN, Enable, Color, Mode);
2 output := LCD_BKLT_REM_1.ENO
3 LCD_BKLT_REM_1 := LCD_BKLT_REM_1.LCD_BKLT_REM
4 sts_lcd_rem := LCD_REM_1.Sts
```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

LCD_REM (LCD remoto)

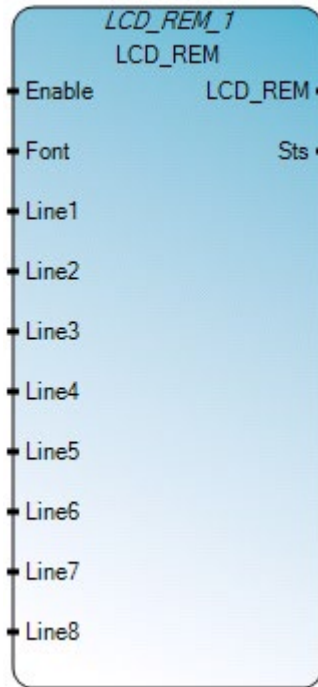
Visualizza i messaggi definiti dall'utente per l'LCD remoto.

Dettagli operazione:

- Da Line1 a Line8 la lunghezza massima della stringa è di 24 caratteri.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820.

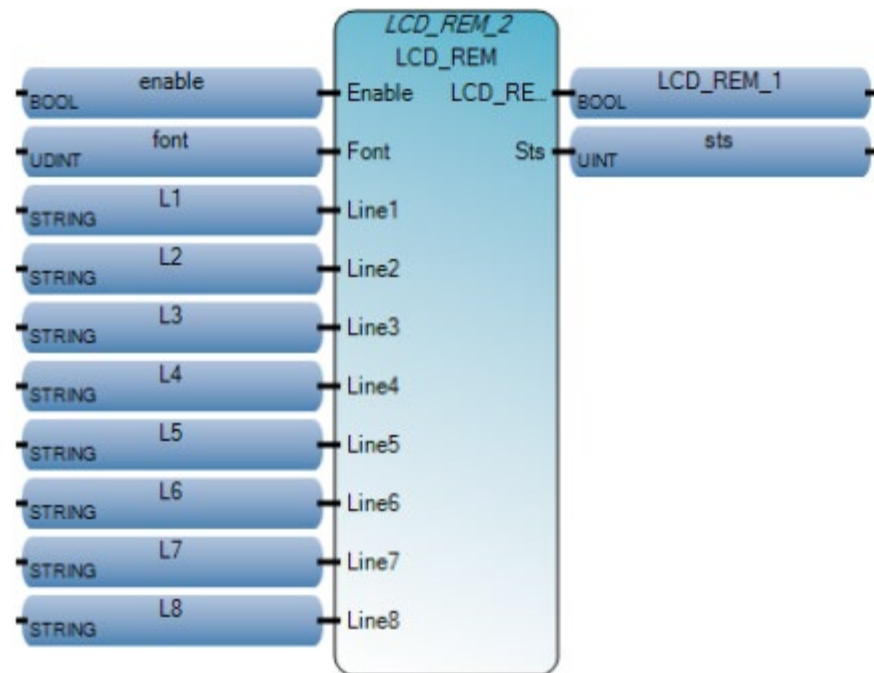


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

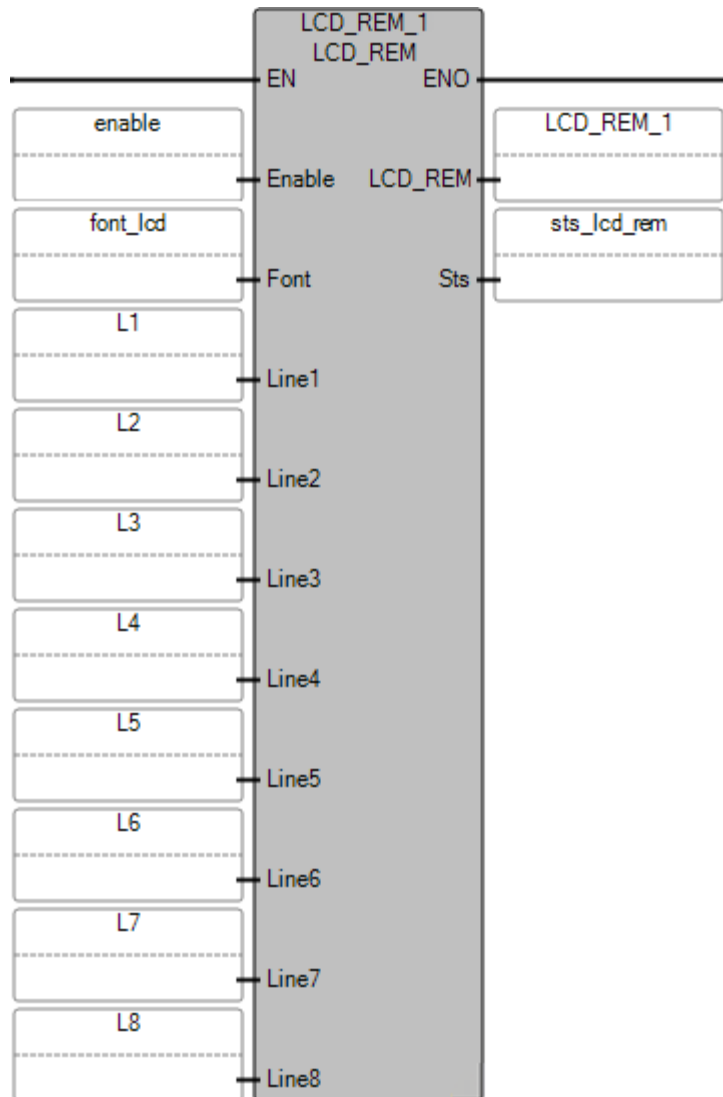
Parametri	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitare il blocco istruzione. TRUE: l'LCD remoto passa dalla schermata di stato I/O alla schermata definita dall'utente. FALSE: l'LCD remoto torna alla schermata di stato I/O.
Carattere	Ingresso	UDINT	Dimensioni carattere per il messaggio di avvio: <ul style="list-style-type: none"> • 0: Predefinito (grande - 8x16) • 1: Piccolo (8x8) • 2: Grande (8x16) • 3: Molto grande (16x16) • Da 4 in poi: Riservato Le dimensioni dell'LCD remoto sono pari a 192x64 pixel. Se l'opzione selezionata per le dimensioni carattere del messaggio di avvio è Piccolo, l'LCD remoto visualizza: <ul style="list-style-type: none"> • Stringhe dalla riga 1 alla riga 8. • Un massimo di 24 caratteri per riga. Se l'opzione selezionata per le dimensioni carattere del messaggio di avvio è Grande o Molto grande, l'LCD remoto visualizza: <ul style="list-style-type: none"> • Stringhe dalla riga 1 alla riga 4. • Ignora le stringhe dalla riga 5 alla riga 8. • Un massimo di 12 caratteri per riga.
Line1	Ingresso	Stringa	Stringa visualizzare alla riga 1 del LCD. Da Line1 a Line8 la lunghezza massima della stringa è di 24 caratteri.
Line2	Ingresso	Stringa	Stringa visualizzare alla riga 2 dell'LCD.
Line3	Ingresso	Stringa	Stringa visualizzare alla riga 3 dell'LCD.
Line4	Ingresso	Stringa	Stringa visualizzare alla riga 4 dell'LCD.
Line5	Ingresso	Stringa	Stringa visualizzare alla riga 5 dell'LCD.
Line6	Ingresso	Stringa	Stringa visualizzare alla riga 6 dell'LCD.

Line7	Ingresso	Stringa	Stringa visualizzare alla riga 7 dell'LCD.
Line8	Ingresso	Stringa	Stringa visualizzare alla riga 8 dell'LCD.
LCD_REM	Uscita	BOOL	Abilita Blocco funzione. Quando Enable = TRUE, lo schermo dell'utente è attivo. Quando Enable = FALSE, lo schermo Stato/Menu IO è attivo.
Sts	Uscita	UINT	Stato dell'operazione dell'LCD remoto. Codici di stato LCD_REM: <ul style="list-style-type: none"> • 0: ingresso abilitazione è False. • 1: messaggio utente visualizzato correttamente. • 2: LCD remoto non rilevato. • 3: errore di connessione. Potrebbe verificarsi quando: <ul style="list-style-type: none"> • L'LCD remoto non è connesso fisicamente al controllore (o il cablaggio non è corretto). • Le impostazioni della porta seriale sono diverse rispetto a quelle richieste per l'LCD remoto. <ul style="list-style-type: none"> • 4: codice font non valido. • 5: 5-65535: riservati.

Esempio di programmazione a blocchi funzionali LCD_REM



Esempio di diagramma ladder LCD_REM



Esempio di testo strutturato LCD_REM

```

LCD_REM_1(
void LCD_REM_1(BOOL Enable, UDINT Font, STRING Line1, STRING Line2, STRING Line3, STRING Line4, STRING Line5, STRING Line6, STRING Line7, STRING Line8)
Tipo: LCD_REM, Visualizzare le stringhe utente sull'LCD remoto, quando è connesso.

1| LCD_REM_1 (EN, Enable, Font, Line1, Line2, Line3, Line4, Line5, Line6, Line7, Line8);
2| output := LCD_REM_1.ENO
3| LCD_REM_1 := LCD_REM_1.LCD_REM
4| sts_lcd_rem := LCD_REM_1.Sts
    
```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

RHC (lettura high speed clock)

Legge il valore high speed clock nel controllore Micro800.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



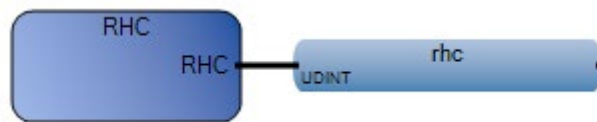
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: lettura high speed clock. FALSE: nessuna operazione. Applicabile alle programmazioni in Ladder Diagram.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.
RHC	Uscita	UDINT	Valore high speed clock.

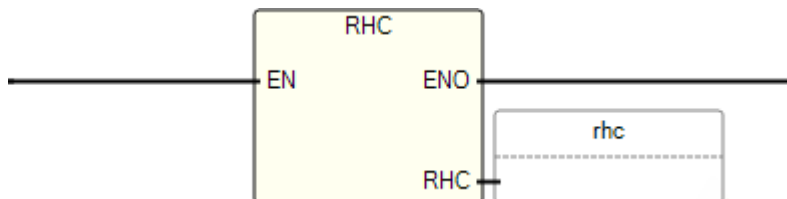
Risoluzione del clock ad alta velocità

Tipo di controllore	Incrementi	Timebase	Risoluzione
Micro810	4 ogni 40 microsecondi	10 microsecondi	40 microsecondi
Micro820 Micro830 Micro850	1 ogni 10 microsecondi	10 microsecondi	10 microsecondi

Esempio di programmazione a blocchi funzionali RHS



Esempio di Diagramma ladder RHS



Esempio di testo strutturato RHS

```

RHC (|
  UDINT RHC()
  Leggere l'orologio ad alta velocità.

1| rhc := RHC ();
(* Equivalenza ST: *)
TESTOUTPUT2 := RHC();
    
```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

RPC (lettura checksum del programma)

Legge il checksum del programma dell'utente dal controllore o dal modulo di memoria.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

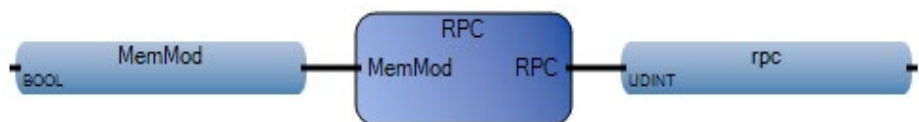
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.



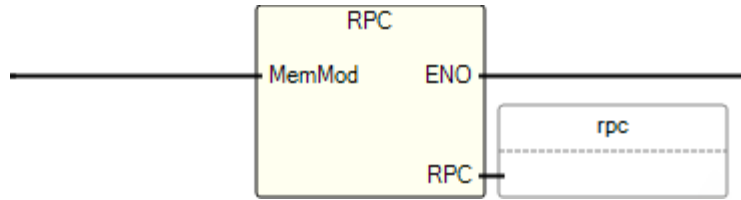
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
MemMod	Ingresso	BOOL	TRUE: il valore viene preso dal modulo di memoria. FALSE: il valore viene preso dal controllore Micro800.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.
RPC	Uscita	UDINT	Valore di checksum di un programma specificato dell'utente.

Esempio di programmazione a blocchi funzionali RPC



Esempio di diagramma ladder RPC



Esempio di testo strutturato RPC

```

RPC {
  UDINT RPC(BOOL MemMod)
  Legge il checksum del programma utente.

1 | MemMod := TRUE;
2 | rpc := RPC(MemMod);

(* Equivalenza ST: *)
TESTOUTPUT2 := RPC(TESTINPUT);

```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

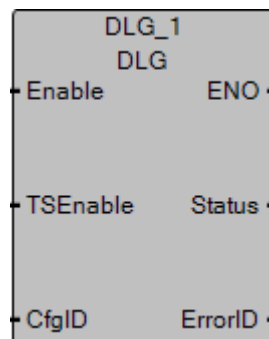
DLG (registro dati)

Scrive i valori di variabile dal motore di runtime in un file di registrazione dati su una scheda SD.

durante la scrittura in un registro dati è consentito un massimo di 50 cartelle di gruppo per giorno. Ogni cartella di gruppo contiene un massimo di 50 file con una dimensione compresa tra 4k e 8k.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820.



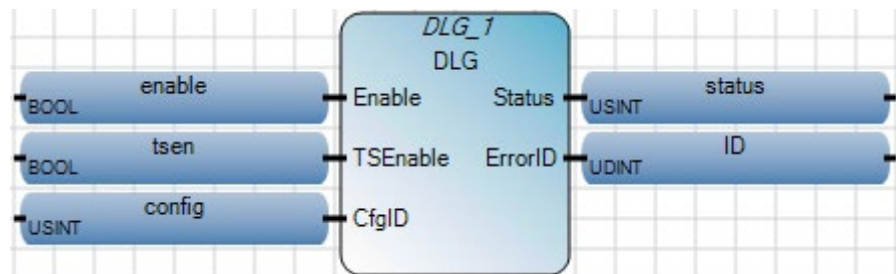
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione di scrittura per la registrazione dati. TRUE: abilitazione del fronte di salita rilevata, avvia l'operazione di registrazione dei dati quando le operazioni dell'istruzione precedente sono state completate. FALSE: fronte di salita non rilevato.
TSEnable	Ingresso	BOOL	TRUE: flag di abilitazione registrazione indicatore data e ora.
CfgID	Ingresso	USINT	Numero ID configurazione registrazione dati VA ID da 1 a 10.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.
Stato	Uscita	USINT	Stato corrente dell'istruzione. Codici di stato registrazione dati: <ul style="list-style-type: none"> • 0: Idle • 1: Doing • 2: Succeed, indica che la registrazione dei dati è stata completata. • 3: Error, indica che la registrazione dei dati è stata completata con errore.
ErrorID	Uscita	UDINT	Un codice di errore numerico univoco per DLG.

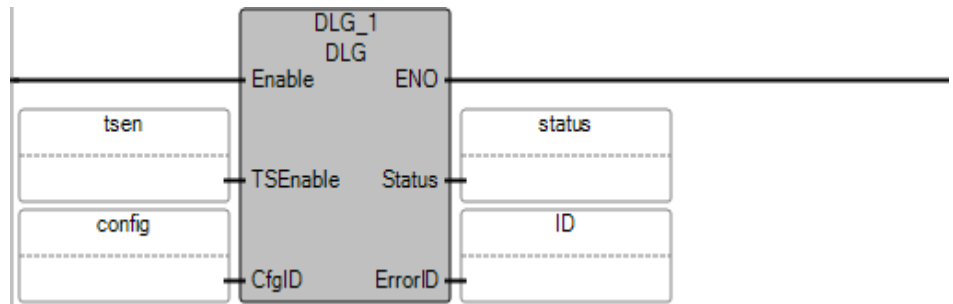
Codici di errore DLG

Codice errore	Nome errore	Commenti
0	DLG_ERR_NONE	Nessun errore.
1	DLG_ERR_NO_SDCARD	Scheda SD assente.
2	DLG_ERR_RESERVED	Riservato.
3	DLG_ERR_DATAFILE_ACCESS	Errore accesso file registrazione dati.
4	DLG_ERR_CFG_ABSENT	Il file di configurazione registrazione dati è assente.
5	DLG_ERR_CFG_ID	L'ID di configurazione nel file di configurazione registrazione dati è assente
6	DLG_ERR_RESOURCE_BUSY	L'operazione di registrazione dati collegata a questo ID di registrazione dati è utilizzata da un'altra operazione FB.
7	DLG_ERR_CFG_FORMAT	Il formato del file di configurazione registrazione dati non è valido.
8	DLG_ERR_RTC	Real-Time Clock non valido.
9	DLG_ERR_UNKNOWN	Si è verificato un errore non specificato.

Esempi di programmazione a blocchi funzionali DLG



Esempio di diagramma ladder DLG



Esempio di testo strutturato DLG

```
DLG_1 (
void DLG_1(BOOL Enable, BOOL TSEnable, USINT CfgID)
Tipo: DLG, Salvare un elenco di istanze di dati in un file di registro dati della scheda SD.

1 DLG_1 (EN, Enable, TSEnable, CfgID)
2 output := DLG_1.ENO
3 status := DLG_1.Status
4 ID := DLG_1.ErrorID
```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

IIM (ingresso immediato)

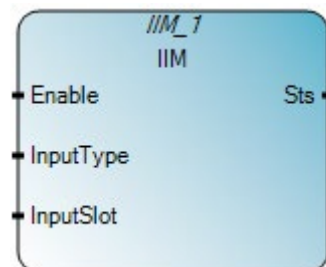
Aggiornamento degli input prima della normale scansione in output.

Dettagli operazione:

- Usate di solito all'inizio di un programma di interrupt per selezionare o mascherare gli ingressi che sono immediatamente scansionati per ottenere gli ingressi correnti.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

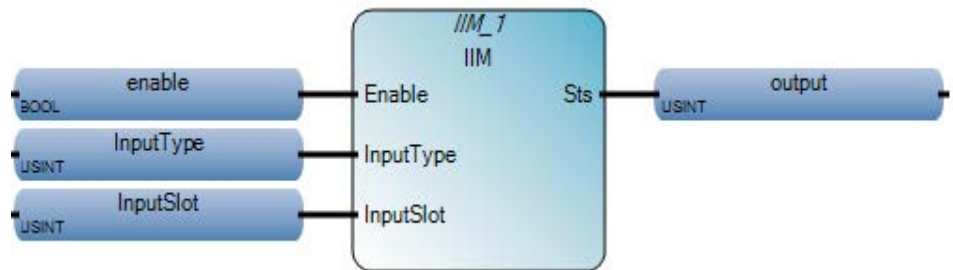
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870.



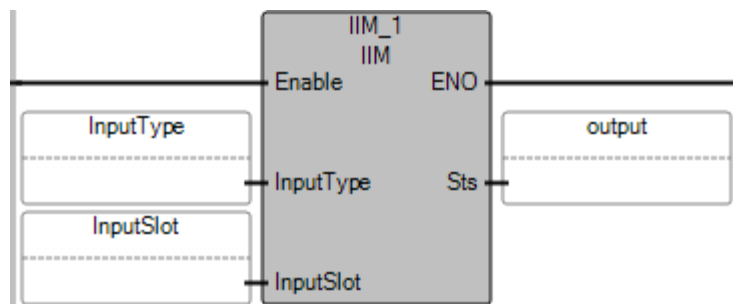
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il blocco istruzione. FALSE: non viene eseguito.
InputType	Ingresso	USINT	Identifica il tipo di ingresso. 0 - Ingresso incorporato. 1 - Ingresso plug-in.
InputSlot	Ingresso	USINT	Identifica lo slot di ingresso. 0: ingresso integrato. 1, 2, 3, 4, 5: numero slot plug-in. (Gli slot sono numerati da sinistra a destra, a partire da numero 1.) Per ingresso incorporato, è sempre 0. Per ingresso plug-in, slot ingresso 1,2,3,4,5,(numero slot plug-in, inizia dal primo slot a sinistra = 1).
Sts	Uscita	USINT	Stato scansione ingresso immediato. Codici di stato (Sts) IIM: <ul style="list-style-type: none"> • 0x00: non abilitato (nessuna azione intrapresa). • 0x01: scansione ingresso/uscita riuscita. • 0x02: tipo ingresso/uscita non valido. • 0x03: slot ingresso/uscita non valido.
ENO	Uscita	BOOL	Abilita uscita. TRUE: ingresso aggiornato. FALSE: ingresso non aggiornato. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali IIM



Esempio di diagramma ladder IIM



Esempio di testo strutturato IIM

```

IIM_1 {
void IIM_1(BOOL Enable, USINT InputType, USINT InputSlot)
Tipo: IIM, Aggiornare gli ingressi prima della scansione degli ingressi normali.

1  enable := TRUE;
2  InputType := 0;
3  InputSlot := 0;
4  IIM_1(enable, InputType, InputSlot);
5  output := IIM_1.Sts;

```

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
IIM_1	<input type="checkbox"/>	IIM
InputSlot	0	N/D		<input type="checkbox"/>	USINT
InputType	0	N/D		<input type="checkbox"/>	USINT
output	1	N/D		<input type="checkbox"/>	USINT

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

IOM (uscita istantanea)

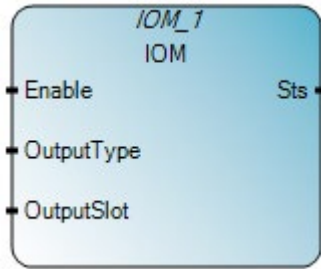
Aggiornare le uscite prima della scansione delle uscite normali.

Dettagli operazione:

- Usate di solito alla fine di un programma di interrupt per selezionare o mascherare le uscite che sono immediatamente scansionate e aggiornate.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

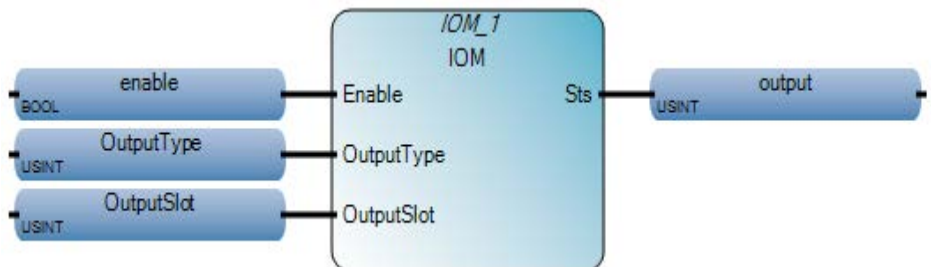
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870.



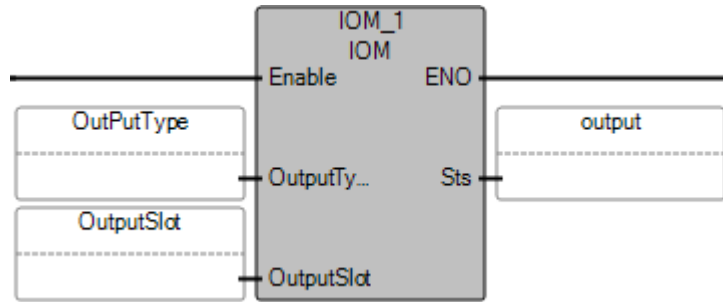
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue l'istruzione. FALSE: non viene eseguito.
OutputType	Ingresso	USINT	Identifica il tipo di uscita: 0 - Uscita integrata. 1 - Uscita plug-in.
OutputSlot	Ingresso	USINT	Identifica lo slot di uscita: 0: uscita integrata 1, 2, 3, 4, 5: numero slot plug-in. (Gli slot sono numerati da sinistra a destra, a partire da numero 1.) Per uscita integrata, è sempre 0. Per uscita plug-in, slot uscita 1,2,3,4,5 (numero slot plug-in, inizia dal primo slot a sinistra = 1).
Sts	Uscita	USINT	Stato scansione uscita immediato. Codici di stato (sts) IOM: <ul style="list-style-type: none"> • 0x00: non abilitato (nessuna azione intrapresa). • 0x01: scansione ingresso/uscita riuscita. • 0x02: tipo ingresso/uscita non valido. • 0x03: slot ingresso/uscita non valido.
ENO	Uscita	BOOL	Abilita uscita. TRUE: uscita aggiornata. FALSE: uscita non aggiornata. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali IOM



Esempio di diagramma ladder IOM



Esempio di testo strutturato IOM

```

IOM_1 (
void IOM_1(BOOL Enable, USINT OutputType, USINT OutputSlot)
Tipo: IOM, Aggiornare le uscite prima della scansione delle uscite normali.

1 enable := TRUE;
2 OutputType := 0;
3 OutputSlot := 0;
4 IOM_1(enable, OutputType, OutputSlot);
5 output := IOM_1.Sts;

```

Risultati

Monitoraggio della variabile							
Variabili globali utente - Micro870		Variabili locali - RA_IOM_LD		Variabili di sistema - Micro870		I/O - Micro870	Parole definite
Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati		
OutputType	0	N/D		<input type="checkbox"/>	USINT		
OutputSlot	0	N/D		<input type="checkbox"/>	USINT		
output	1	N/D		<input type="checkbox"/>	USINT		
IOM_1	<input type="checkbox"/>	IOM		
enable	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL		

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

KEY_READ (lettura tasti su LCD)

Legge lo stato della chiave su un modulo LCD opzionale quando il display dell'utente è attivo.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

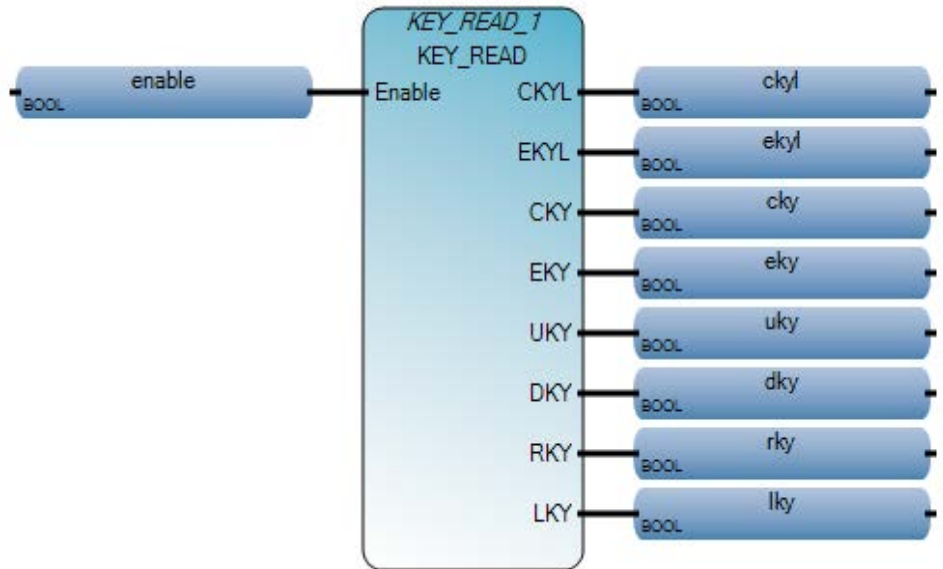
Questa istruzione si applica ai controllori Micro810.



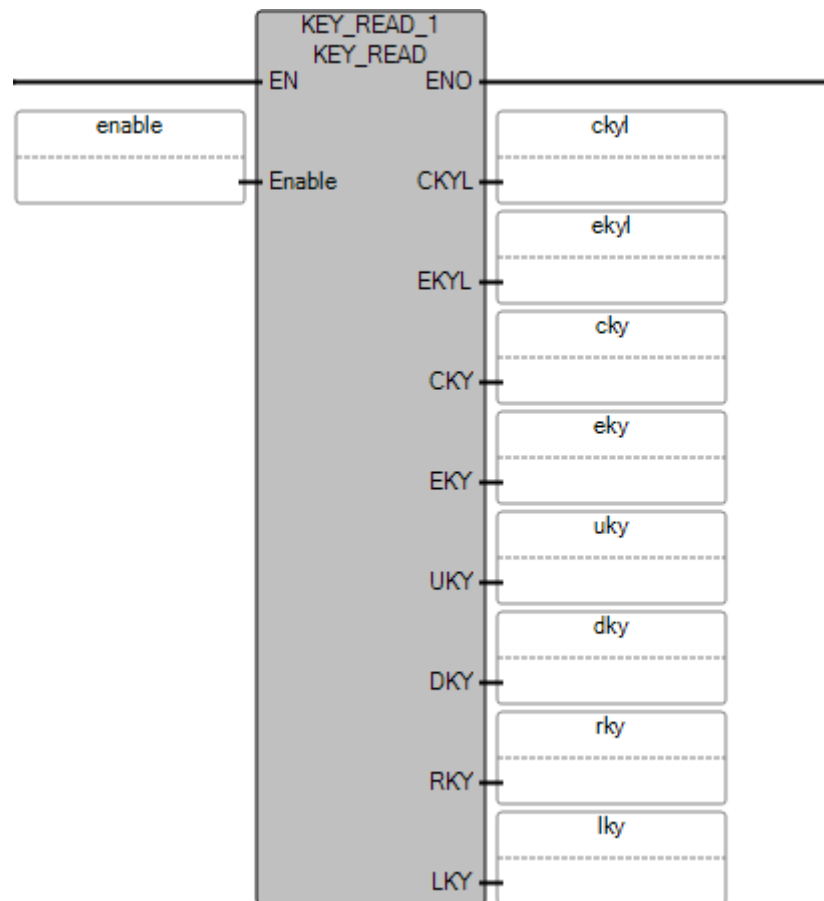
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilita il blocco istruzione. TRUE: abilita i tasti di lettura sulla tastiera LCD remoto. FALSE: disabilita i tasti di lettura sulla tastiera LCD remoto.
CKYL	Uscita	BOOL	TRUE: tasto ESC premuto per più di 2 secondi.
EKYL	Uscita	BOOL	TRUE: tasto OK premuto per più di 2 secondi.
CKY	Uscita	BOOL	TRUE: tasto ESC premuto.
EKY	Uscita	BOOL	TRUE: tasto OK premuto.
UKY	Uscita	BOOL	TRUE: tasto Su premuto.
DKY	Uscita	BOOL	TRUE: tasto Giù premuto.
LKY	Uscita	BOOL	TRUE: tasto Sinistra premuto.
RKY	Uscita	BOOL	TRUE: tasto Destra premuto.

Esempio di programmazione a blocchi funzionali KEY_READ



Esempio di diagramma ladder KEY_READ



Esempio di testo strutturato KEY_READ

KEY_READ_1 (

void KEY_READ_1(BOOL Enable)

Tipo: KEY_READ, Leggere lo stato della chiave sul modulo LCD opzionale.

```

1: KEY_READ_1(enable);
2: cky1 := KEY_READ_1.CKYL;
3: eky1 := KEY_READ_1.EKYL;
4: cky := KEY_READ_1.CKY;
5: eky := KEY_READ_1.EKY;
6: uky := KEY_READ_1.UKY;
7: dky := KEY_READ_1.DKY;
8: lky := KEY_READ_1.LKY;
9: rky := KEY_READ_1.RKY;

```

(* Equivalenza ST: *)

```

KEY_READ_1(KEYENABLE) ;
KEY_EKYL := KEY_READ_1.EKYL ;
KEY_CKY := KEY_READ_1.CKY ;
KEY_EKY := KEY_READ_1.EKY ;
KEY_UKY := KEY_READ_1.UKY ;
KEY_DKY := KEY_READ_1.DKY ;
KEY_RKY := KEY_READ_1.RKY ;
KEY_LKY := KEY_READ_1.LKY ;

```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

[LCD](#) a pagina 356

KEY_READ_REM (lettura chiavi per LCD remoto)

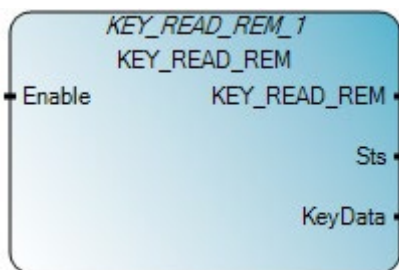
Legge lo stato della chiave su un modulo LCD remoto opzionale quando il display dell'utente è attivo.

Dettagli operazione:

- Utilizzare l'istruzione LCD_REM per attivare il display dell'utente sul modulo LCD remoto. Se il display dell'utente non è attivo, si verifica un errore durante l'esecuzione di KEY_READ_REM.
- La proprietà del TASTO P nel file delle funzioni 'LCD devono essere attivate; altrimenti, tutti gli stati della chiave saranno FALSE.
- Sono supportate solo pressioni di tasti singoli per l'istruzione KEY_READ_REAM; le combinazioni di due tasti non sono supportate.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: abilita FALSE: disabilita
KEY_READ_REM	Uscita	BOOL	TRUE: dati chiave LCD remoto letti con successo. FALSE: l'abilitazione è false, è presente un errore di lettura. Dati chiave LCD remoto o display utente non attivi.
Sts	Uscita	UINT	Stato dell'operazione KEY_READ_REM . Codici di stato KEY_READ_REM: <ul style="list-style-type: none"> • 0: ingresso abilitazione è False. • 1: dati chiave letti correttamente. • 2: LCD remoto non rilevato. Potrebbe verificarsi quando: <ul style="list-style-type: none"> • L'LCD remoto non è connesso fisicamente al controllore (o il cablaggio non è corretto). • Le impostazioni della porta seriale sono diverse rispetto a quelle richieste per l'LCD remoto. • 3: errore di connessione. Potrebbe verificarsi quando è presente un errore interno alla macchina a stati. Probabilmente causato da un'incompatibilità tra la versione FW del controllore e la versione FW RLCD. • 4: display utente non attivo. • 5-65535: riservato.
KeyData	Uscita	UDINT	Dati del tastierino LCD remoto. Le definizioni KeyData sono specificate nella tabella dei campi di bit KeyData.

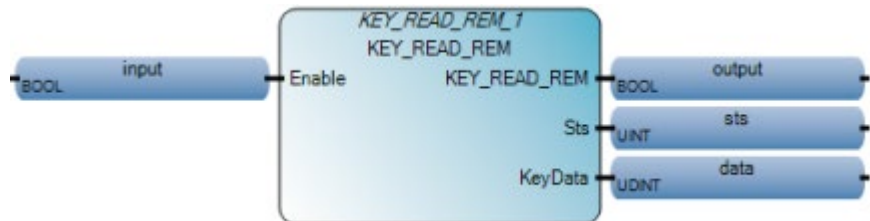
Tabella dei campi di bit KeyData

Utilizzare questa tabella per determinare i campi di bit per KEY_READ_REM.

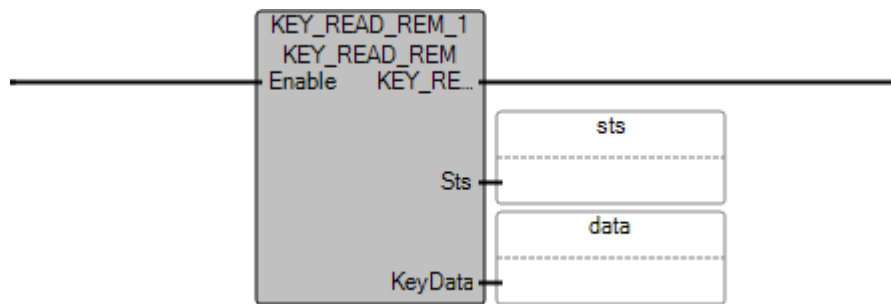
N. bit in KeyData	Nome	Descrizione parametro
0	UKY	TRUE = tasto su premuto.
1	DKY	TRUE = tasto giù premuto.
2	LKY	TRUE = tasto sinistra premuto.
3	RKY	TRUE = tasto destra premuto.
4	F1KY	TRUE = tasto F1 premuto.
5	F2KY	TRUE = tasto F2 premuto.
6	F3KY	TRUE = tasto F3 premuto.
7	F4KY	TRUE = tasto F4 premuto.
8	F5KY	TRUE = tasto F5 premuto.
9	F6KY	TRUE = tasto F6 premuto.

N. bit in KeyData	Nome	Descrizione parametro
10	EKY	TRUE = tasto INVIO premuto.
11	CKY	TRUE = tasto CANC premuto.
12	EKYL	TRUE = tasto INVIO premuto per più di 2 secondi.
13	CKYL	TRUE = tasto CANC premuto per più di 2 secondi.
14-31	--	Riservato.

Esempio di programmazione a blocchi funzionali KEY_READ_REM



Esempio di diagramma ladder KEY_READ_REM



Esempio di testo strutturato KEY_READ_REM

```
KEY_READ_REM_1(
void KEY_READ_REM_1(BOOL Enable)
Tipo: KEY_READ_REM, Controllare lo stato della chiave sull'LCD remoto.
```

```
1 KEY_READ_REM (Enable);
2 output := KEY_READ_REM.ENO
3 sts := KEY_READ_REM.Sts
4 data := KEY_READ_REM.KeyData
```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

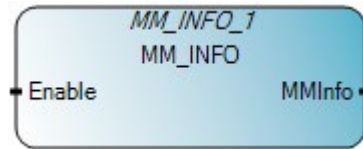
MM_INFO (informazioni sul modulo di memoria)

Legge le informazioni dell'intestazione del modulo di memoria. Se non è presente un modulo di memoria, tutti i valori restituiscono zero (0).

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

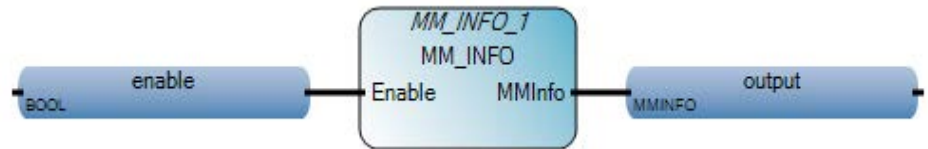
Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.



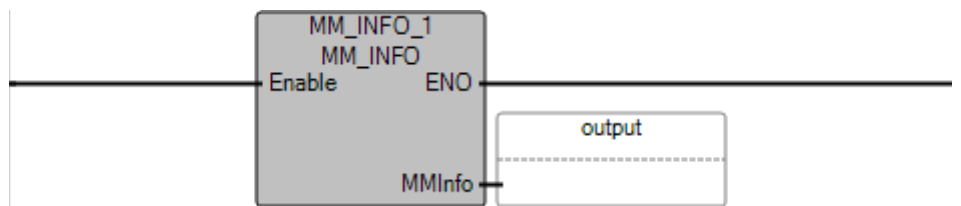
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: legge le informazioni sull'intestazione del modulo di memoria. FALSE: non c'è nessuna operazione di lettura e l'informazione sul modulo di memoria dell'uscita non è valida.
MMInfo	Uscita	MMINFO	Le informazioni sul modulo di memoria sono definite nel tipo di dati MMINFO.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Programmazione a blocchi funzionali MSG_INFO



Esempio di Diagramma ladder MSG_INFO



Esempio di Testo strutturato MSG_INFO

```

MM_INFO_1(
void MM_INFO_1(BOOL Enable)
Tipo: MM_INFO, Leggere le informazioni di intestazione del modulo di memoria.

1: MM_INFO_1(enable);
2: output := MM_INFO_1.MMInfo;

```

Risultati

Per i controllori che utilizzano 2080-MEMBAK-RTC:

Monitoraggio della variabile

Nome	Alias	Valore logico	Valore fisico	Valore iniziale
MM_INFO_2	
MM_INFO_2.Enable		<input checked="" type="checkbox"/>		
MM_INFO_2.MMInfo	
enable		<input type="checkbox"/>	N/D	
output	
output.MMCatalog	
output.MMInfo		2080-MEMBAK-RTC	N/D	
output.Series		1	N/D	
output.Revision		257	N/D	
output.UPValid		<input checked="" type="checkbox"/>	N/D	
output.ModeBehavior		<input type="checkbox"/>	N/D	
output.LoadAlways		<input type="checkbox"/>	N/D	
output.LoadOnError		<input type="checkbox"/>	N/D	
output.FaultOverride		<input type="checkbox"/>	N/D	

Chiudi

Per i controllori che utilizzano una scheda SD:

Variable Monitoring

Name	Alias	Logical Value	Physical Value	Initial Value
TEST_EN		<input checked="" type="checkbox"/>	N/A	
PLUGIN_READ_1	
slotid		1	N/A	
addrOffset		0	N/A	
dataLen		0	N/A	
dataAddr	
sts_read		3	N/A	
MM_INFO_1	
MM_INFO_1.Enable		<input checked="" type="checkbox"/>		
MM_INFO_1.MMInfo	
MM_INFO_1.MMInfo.MMCatalog	
MM_INFO_1.MMInfo.MMInfo		2080-SD-RTC-SC	N/A	
MM_INFO_1.MMInfo.Series		1	N/A	
MM_INFO_1.MMInfo.Revision		257	N/A	
MM_INFO_1.MMInfo.UPValid		<input type="checkbox"/>	N/A	
MM_INFO_1.MMInfo.ModeBehav		<input type="checkbox"/>	N/A	
MM_INFO_1.MMInfo.LoadAlways		<input type="checkbox"/>	N/A	
MM_INFO_1.MMInfo.LoadOnError		<input type="checkbox"/>	N/A	

Close

Vedere anche

[Tipo di dati MMINFO](#) a pagina 380

[Istruzioni ingresso/uscita](#) a pagina 355

Tipo di dati MMINFO

Nella tabella seguente sono descritti i parametri del tipo di dati MMINFO.

Parametro	Tipo di dati	Descrizione
MMCatalog	MMCATNUM	Il codice del modulo di memoria. quando si utilizza l'istruzione MM_INFO su controllori con scheda SD, MMCatalog corrisponde a "SCHEDA SD".
Serie	UINT	La serie del modulo di memoria. Quando si utilizza l'istruzione MM_INFO su controllori con scheda SD, la serie è 1.
Revisione	UINT	La revisione del modulo di memoria. Quando si utilizza MM_INFO su controllori con scheda SD, la revisione è 257.
UPValid	BOOL	Programma utente presente (TRUE: trovato progetto potenzialmente valido). Nota: anche se TRUE, se in fase di download o ripristino risultano file mancanti o danneggiati, è comunque possibile che il progetto venga rilevato come non valido.
ModeBehavior	BOOL	Modalità di comportamento (TRUE: va a RUN all'accensione).
LoadAlways	BOOL	Ripristina sempre il modulo di memoria al controllore all'accensione.
LoadOnError	BOOL	Ripristina sempre il modulo di memoria al controllore in caso di errore all'accensione.
FaultOverride	BOOL	Errore di sostituzione all'accensione.
MMPresent	BOOL	Il modulo di memoria è presente.

Vedere anche

[MM_INFO](#) a pagina 377

MODULE_INFO

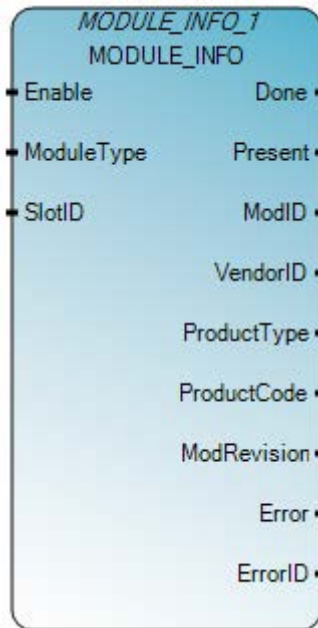
Legge le informazioni di modulo da un modulo plug-in o di espansione.

Dettagli operazione:

- Le informazioni sul modulo plug-in vengono lette durante la fase di esecuzione.
- Il modulo plug-in di memoria 2080-MEMBAK-RTC non è supportato.
- Le informazioni del Modulo di espansione vengono lette quando il modulo è alimentato.
- Quando un modulo plug-in o di espansione non viene definito con un ModuleID, ProductType o ProductCode, l'operazione di MODULE_INFO restituisce 0 per il parametro di uscita interessato.
- Le informazioni di identificazione modulo plug-in e di espansione sono definite da Allen-Bradley e vengono fornite sotto come parte della descrizione di MODULE_INFO.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870. I moduli di espansione sono supportati solo dai controllori Micro850 e Micro870.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

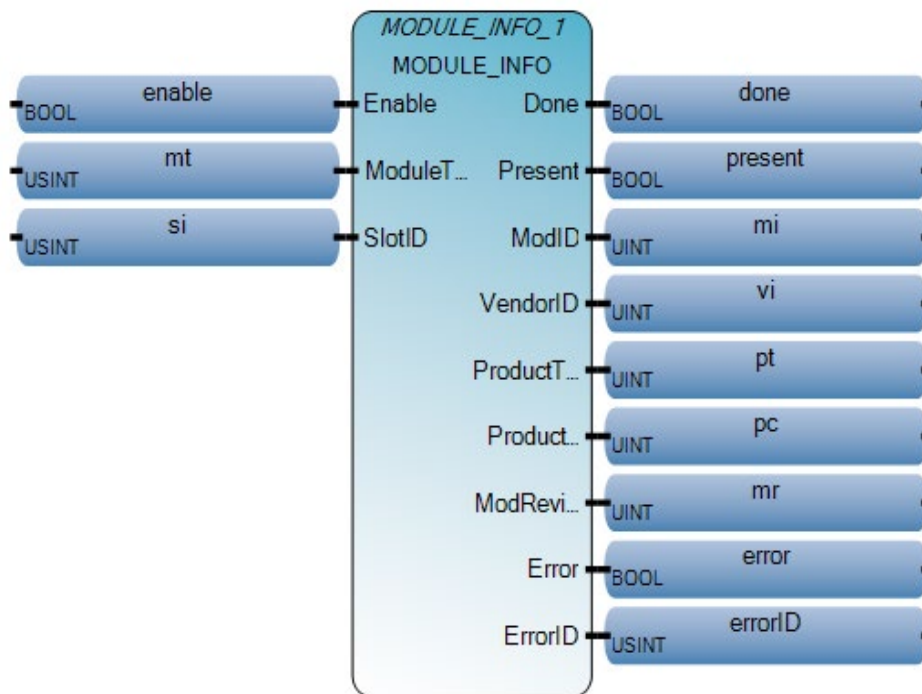
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue l'operazione di lettura MODULE_INFO. FALSE: non esegue l'operazione di lettura. Tutti i valori dei dati di uscita sono ripristinati a 0.
ModuleType	Ingresso	USINT	Identifica il tipo di modulo: <ul style="list-style-type: none"> • 1: modulo di espansione 2085. • 2: modulo plug-in 2080.
SlotID	Ingresso	USINT	Il numero degli slot dove si trovano i moduli plug-in o di espansione. Gli ID slot sono: 1, 2, 3, 4, 5 Lo slot 1 è all'estrema sinistra.
Done	Uscita	BOOL	TRUE: operazione completata correttamente. FALSE: l'operazione non è in esecuzione o si è verificato una condizione di errore.
Presente	Uscita	BOOL	Rileva il modulo plug-in o di espansione nello slot del controllore. TRUE: il modulo è fisicamente presente. FALSE: il modulo non è fisicamente presente.
ModID	Uscita	UINT	Identificazione per il modulo nello slot del controllore. <ul style="list-style-type: none"> • I moduli plug-in sono definiti con un identificatore modulo univoco. • I moduli di espansione non sono definiti con un identificatore modulo univoco, ModID restituisce 0.
VendorID	Uscita	UINT	ID fornitore del modulo plug-in o di espansione. Per i prodotti Allen-Bradley, l'ID fornitore è 1.
Tipo prodotto:	Uscita	UINT	Tipo prodotto del modulo plug-in o di espansione.
Codice Prodotto	Uscita	UINT	Codice prodotto del modulo plug-in o di espansione.
ModRevision	Uscita	UINT	Informazioni sulla revisione del modulo plug-in o di espansione.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	USINT	Numero univoco che identifica l'errore. Gli errori sono definiti nei codici di errore MODULE_INFO.

Codici di errore MODULE_INFO

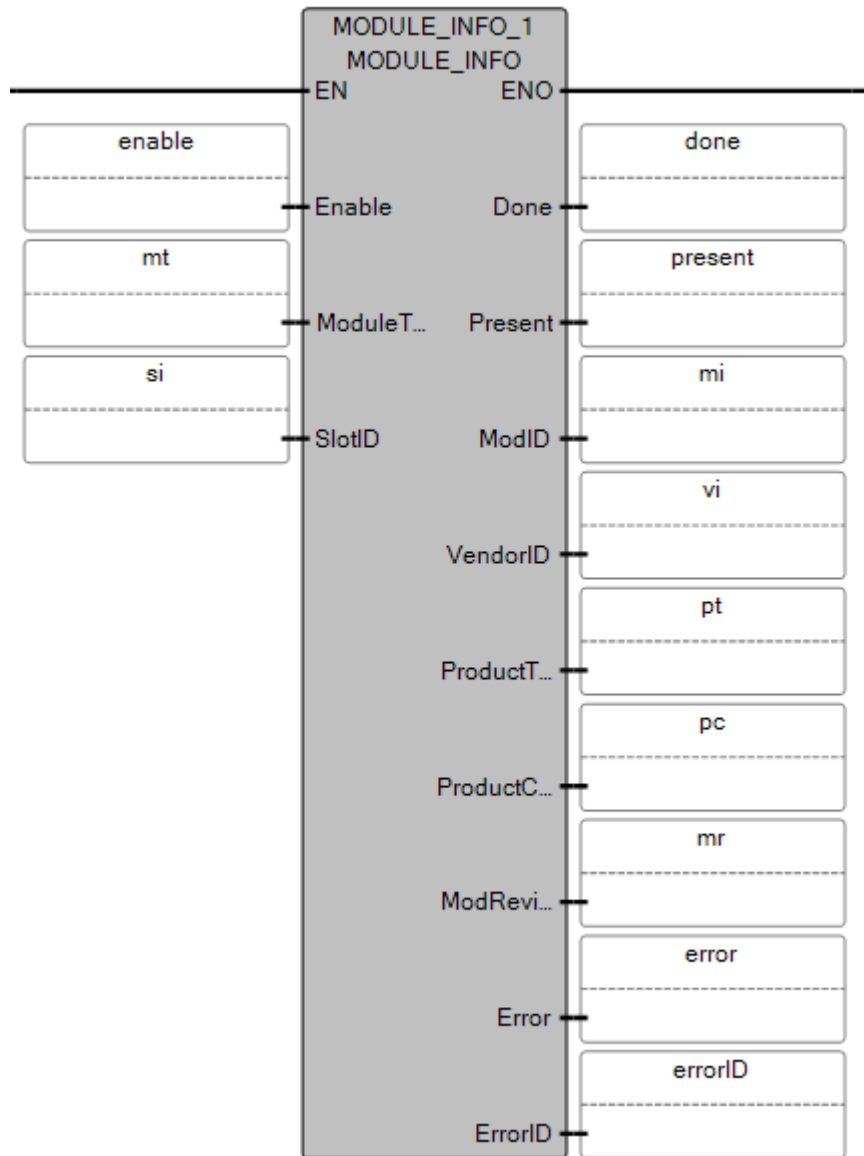
Utilizzare questa tabella per determinare i codici di errore MODULE_INFO e le descrizioni.

Codice errore	Descrizione errore
1	Tipo di modulo non valido. Sostituire con un tipo di modulo valido. I tipi di modulo validi sono: <ul style="list-style-type: none"> • 1: modulo di espansione 2085. Supportato solo dai controllori Micro850. • 2: modulo plug-in 2080. Supportato dai controllori Micro820, Micro830 e Micro850.
2	Numero slot non valido. Sostituire con un numero slot valido.
3	Tipo di modulo di espansione non valido.
4	Errore irreversibile del modulo di espansione.
5	Le informazioni sulla lettura del modulo plug-in non sono supportate.
6	Si è verificato un errore nella lettura del modulo plug-in durante la lettura dell'ID modulo.
7	Si è verificato un errore nella lettura del modulo plug-in durante la lettura dell'ID fornitore.
8	Errore nella lettura del modulo plug-in durante la lettura del Tipo prodotto.
9	Si è verificato un errore nella lettura del modulo plug-in durante la lettura del Codice prodotto.
10	Si è verificato un errore nella lettura del modulo plug-in durante la lettura della Revisione modulo.

Esempio di Programmazione a blocchi funzionali MODULE_INFO



Esempio di Diagramma ladder MODULE_INFO



Esempio di Testo strutturato MODULE_INFO

```

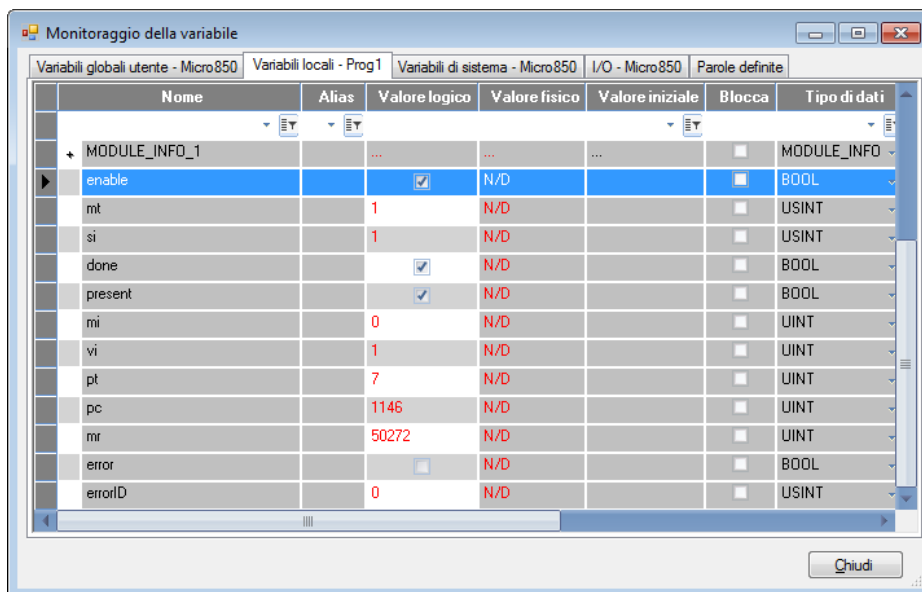
1  MODULE_INFO_1(enable, mt, si);
2  done := MODULE_INFO_1.Done;
3  present :=MODULE_INFO_1.Present;
4  mi :=MODULE_INFO_1.ModID;
5  vi :=MODULE_INFO_1.VendorID;
6  pt :=MODULE_INFO_1.ProductType;
7  pc :=MODULE_INFO_1.ProductCode;
8  mr :=MODULE_INFO_1.ModRevision;
9  error :=MODULE_INFO_1.Error;
10 errorID :=MODULE_INFO_1.ErrorID;

```

MODULE_INFO_1 |

void **MODULE_INFO_1**(BOOL Enable, USINT ModuleType, USINT SlotID)
 Tipo: MODULE_INFO, Ottenere informazioni sul modulo dall'IO di espansione e del plug-in

Risultati



Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

[Diagrammi di temporizzazione dell'istruzione MODULE_INFO](#) a pagina 386

[MODULE_INFO - Informazioni sui moduli plug-in e di espansione](#) a pagina 384

MODULE_INFO - Informazioni sui moduli plug-in e di espansione

Le seguenti informazioni forniscono il tipo, l'ID modulo, l'ID fornitore e i Codici prodotto per i moduli plug-in e di espansione, oltre alle descrizioni delle parole per la revisione dei moduli di espansione definite da Allen-Bradley.

Informazioni sul modulo a innesto

Utilizzare questa tabella per determinare le informazioni sul modulo plug-in definite da Allen-Bradley.

Modulo Plug-in	Tipo plug-in	ID modulo	ID fornitore	Tipo prodotto	Codice prodotto
2080-IF2	Analogico	96	1	10	32
2080-IF4	Analogico	98	1	10	33
2080-OF2	Analogico	100	1	10	34
2080-TC2	Analogico	102	1	10	35
2080-RTD2	Analogico	104	1	10	36
2080-DNET20	Comunicazione	34	1	12	249
2080-SERIALISOL	Comunicazione	32	1	-	-
2080-IQ4	Digitale	192	1	7	192
2080-OB4	Digitale	193	1	7	193

2080-OV4	Digitale	194	1	7	194
2080-IQ4OB4	Digitale	195	1	7	195
2080-IQ4OV4	Digitale	196	1	7	196
2080-OW4I	Digitale	197	1	7	197
2080-MOT-HSC	Speciale	48	1	43	48
2080-TRIMPOT6	Speciale	72	1	-	-

Informazioni sul modulo di espansione

Utilizzare questa tabella per determinare le informazioni sul modulo di espansione definite da Allen-Bradley.

Codice modello	Tipo di espansione	ID modulo	ID fornitore	Tipo prodotto	Codice prodotto
2085-IF4	Analogico	-	1	10	208
2085-IF8	Analogico	-	1	10	206
2085-IRT4	Analogico	-	1	10	213
2085-OF4	Analogico	-	1	10	214
2085-IA8	Digitale	-	1	7	1148
2085-IM8	Digitale	-	1	7	1152
2085-IQ16	Digitale	-	1	7	1144
2085-IQ32T	Digitale	-	1	7	1145
2085-OA8	Digitale	-	1	7	1149
2085-OB16	Digitale	-	1	7	1146
2085-OV16	Digitale	-	1	7	1147
2085-OW16	Digitale	-	1	7	1151
2085-OW8	Digitale	-	1	7	1150

Descrizioni delle parole per la revisione del modulo di espansione

Utilizzare questa tabella per determinare le informazioni sulle parole per la revisione dei moduli di espansione Allen-Bradley.

Bit	Nome	Descrizione																																																																								
15 14	Baud rate massimo (1:0)	I bit identificano la frequenza massima per il Baud Rate massimo (1:0). <ul style="list-style-type: none"> • 00 (bin) è 2 MB/s • 01 (bin) è 4 MB/s • 10 (bin) è 8 MB/s • 11 (bin) è 16 MB/s 																																																																								
13 12 11 10	Revisione secondaria (3:0)	La Revisione secondaria (3:0) è la denominazione della revisione secondaria del prodotto. Questo campo indica la revisione minore del numero del catalogo designato da ID fornitore, Tipo prodotto, Codice prodotto, Serie e Revisione principale. La Revisione secondaria è compresa tra 0 e 15.																																																																								
9 8 7 6 5	Revisione principale (4:0)	La Revisione principale (4:0) è la denominazione della revisione principale del prodotto. Questo campo indica la revisione principale del numero del catalogo designato da ID fornitore, Tipo prodotto, Codice prodotto e Serie. La Revisione principale è compresa tra 0 e 31.																																																																								
4	Serie (4:0)	SERIE (4:0) è la denominazione delle serie del prodotto. Questo campo indica la lettera di serie del numero del catalogo designato da ID fornitore, Tipo prodotto e Codice prodotto. <table border="1" data-bbox="662 697 1429 1031"> <thead> <tr> <th>Serie (4:0)</th> <th>Serie</th> <th>Serie (4:0)</th> <th>Serie</th> <th>Serie (4:0)</th> <th>Serie</th> <th>Serie (4:0)</th> <th>Serie</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>A</td> <td>8</td> <td>I</td> <td>16</td> <td>Q</td> <td>24</td> <td>Y</td> </tr> <tr> <td>1</td> <td>B</td> <td>9</td> <td>J</td> <td>17</td> <td>R</td> <td>25</td> <td>Z</td> </tr> <tr> <td>2</td> <td>C</td> <td>10</td> <td>K</td> <td>18</td> <td>S</td> <td>26</td> <td>AA</td> </tr> <tr> <td>3</td> <td>D</td> <td>11</td> <td>L</td> <td>19</td> <td>T</td> <td>27</td> <td>AB</td> </tr> <tr> <td>4</td> <td>E</td> <td>12</td> <td>M</td> <td>20</td> <td>U</td> <td>28</td> <td>AC</td> </tr> <tr> <td>5</td> <td>F</td> <td>13</td> <td>N</td> <td>21</td> <td>V</td> <td>29</td> <td>AD</td> </tr> <tr> <td>6</td> <td>G</td> <td>14</td> <td>O</td> <td>22</td> <td>W</td> <td>30</td> <td>AE</td> </tr> <tr> <td>7</td> <td>H</td> <td>15</td> <td>P</td> <td>23</td> <td>X</td> <td>31</td> <td>>AE</td> </tr> </tbody> </table>	Serie (4:0)	Serie	Serie (4:0)	Serie	Serie (4:0)	Serie	Serie (4:0)	Serie	0	A	8	I	16	Q	24	Y	1	B	9	J	17	R	25	Z	2	C	10	K	18	S	26	AA	3	D	11	L	19	T	27	AB	4	E	12	M	20	U	28	AC	5	F	13	N	21	V	29	AD	6	G	14	O	22	W	30	AE	7	H	15	P	23	X	31	>AE
Serie (4:0)	Serie	Serie (4:0)	Serie	Serie (4:0)	Serie	Serie (4:0)	Serie																																																																			
0	A	8	I	16	Q	24	Y																																																																			
1	B	9	J	17	R	25	Z																																																																			
2	C	10	K	18	S	26	AA																																																																			
3	D	11	L	19	T	27	AB																																																																			
4	E	12	M	20	U	28	AC																																																																			
5	F	13	N	21	V	29	AD																																																																			
6	G	14	O	22	W	30	AE																																																																			
7	H	15	P	23	X	31	>AE																																																																			

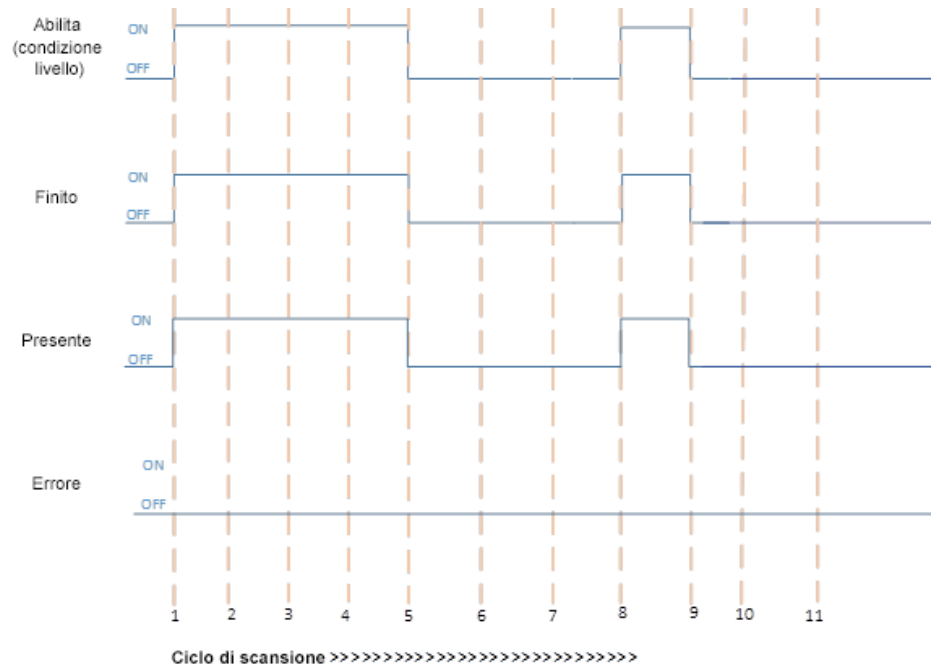
Vedere anche

[MODULE_INFO](#) a pagina 380

Diagrammi di temporizzazione dell'istruzione MODULE_INFO

Gli esempi di diagramma di temporizzazione seguenti descrivono scenari di esecuzione per l'istruzione MODULE_INFO.

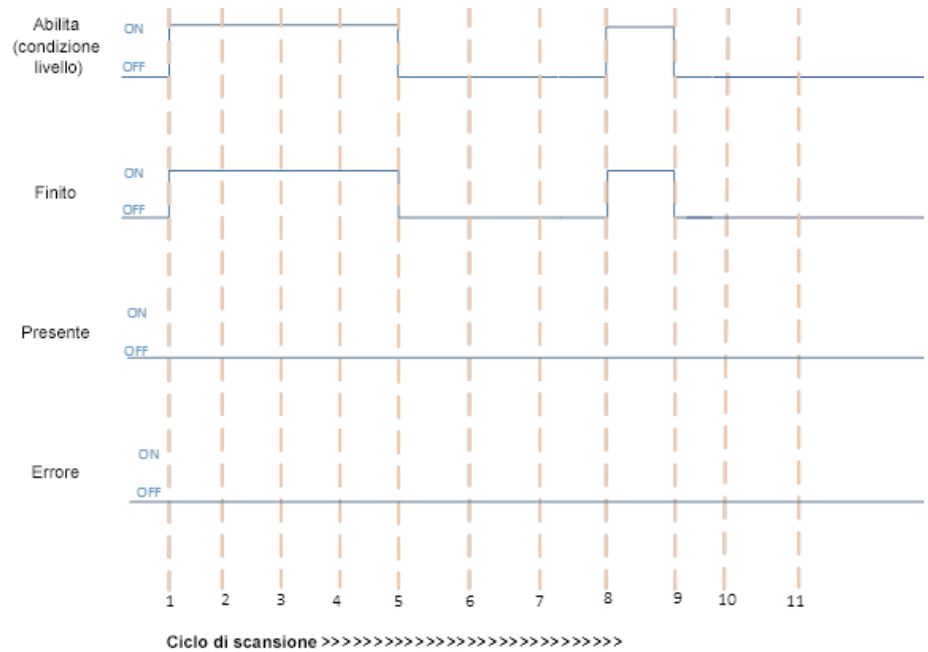
Completamento dell'esecuzione quando un modulo è fisicamente presente



Utilizzare questa tabella per determinare i valori del parametro MODULE_INFO per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1, 8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • ModuleType e SlotID sono validi. Un modulo fisico è presente. • I bit di uscita Completato e Presente sono TRUE. • Il bit di uscita Errore è FALSE. • Aggiornare di conseguenza le informazioni sul modulo per ID modulo, ID fornitore, Tipo prodotto, codice prodotto e Revisione modulo.
2, 3, 4	Nessuna modifica nella condizione ramo.
5, 9	La condizione Ramo diventa FALSE quando il bit Abilita è FALSE. Tutti i parametri di uscita vengono cancellati.
6, 7, 10, 11	Nessuna modifica nella condizione ramo. Il bit Abilita è FALSE. Tutti i parametri di uscita vengono cancellati.

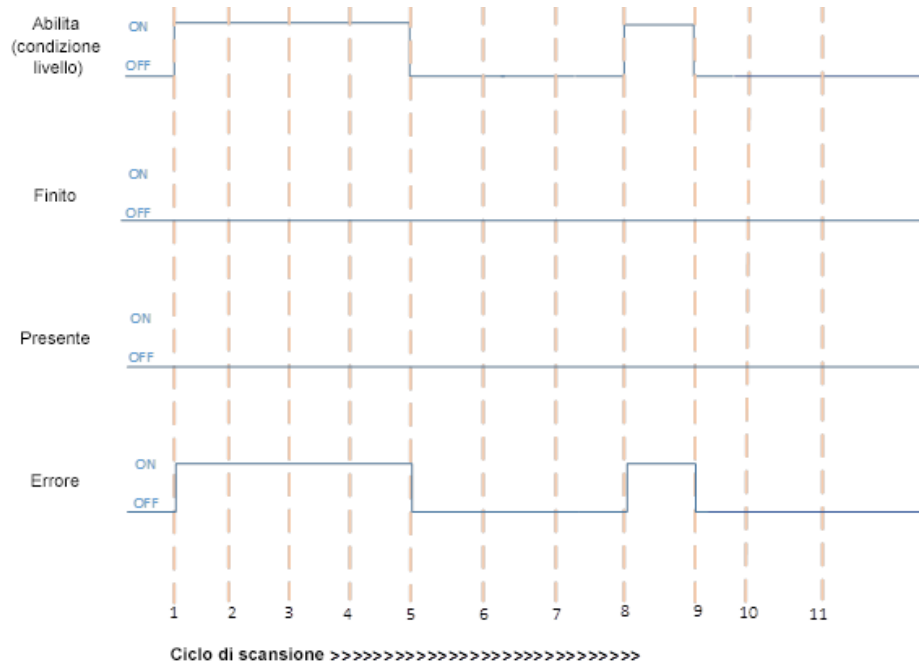
Completamento dell'esecuzione quando il modulo non è fisicamente presente



Utilizzare questa tabella per determinare i valori del parametro MODULE_INFO per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1, 8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • ModuleType e SlotID sono validi. Non è presente un modulo fisico. • Il bit di uscita Completato è TRUE. • I bit di uscita Errore e Presente sono FALSE. • Aggiornare di conseguenza le informazioni sul modulo per ID modulo, ID fornitore, Tipo prodotto, codice prodotto e Revisione modulo.
2, 3, 4	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso sono validi e non è presente un modulo fisico. • Aggiornare di conseguenza i parametri di uscita.
5, 9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Tutti i parametri di uscita vengono cancellati.
6, 7, 10, 11	Nessuna modifica nella condizione ramo. Il bit Abilita è FALSE. Tutti i parametri di uscita vengono cancellati.

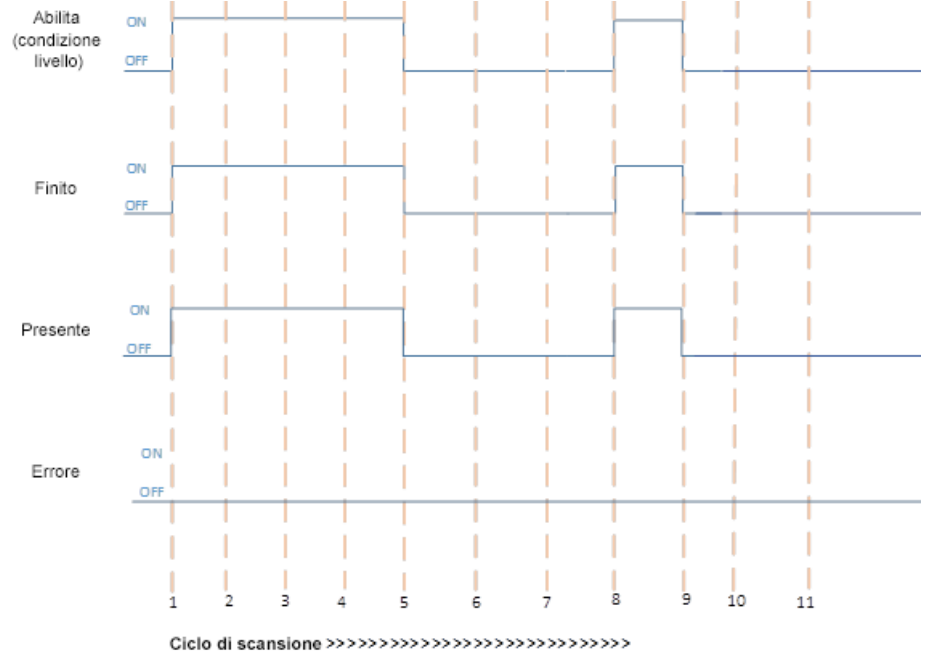
Esecuzione di MODULE_INFO con errore



Utilizzare questa tabella per determinare i valori del parametro MODULE_INFO per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1, 8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • ModuleType e SlotID sono validi. Non è presente un modulo fisico. • I bit di uscita Completato e Presente sono TRUE. • Il bit di uscita Errore è TRUE. • Le informazioni sul modulo per ID modulo, ID fornitore, Tipo prodotto, codice prodotto e Revisione modulo sono deselezionate.
2, 3, 4	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • ModuleType, SlotID o entrambi non sono validi. • Aggiornare di conseguenza i parametri di uscita.
5, 9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Tutti i parametri di uscita vengono cancellati.
6, 7, 10, 11	Nessuna modifica nella condizione ramo. Il bit Abilita è FALSE. Tutti i parametri di uscita vengono cancellati.

Completamento dell'esecuzione di MODULE_INFO con errore quando è presente un modulo fisico



Utilizzare questa tabella per determinare i valori del parametro MODULE_INFO per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1, 8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • ModuleType o SlotID o entrambi sono validi. Il modulo è fisicamente presente. • I bit di uscita Completato e Presente sono TRUE. • Il bit di uscita Errore è FALSE. • Aggiornare di conseguenza le informazioni sul modulo per ID modulo, ID fornitore, Tipo prodotto, codice prodotto e Revisione modulo.
2	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso del modulo sono validi e il modulo è fisicamente presente. • Aggiornare di conseguenza i parametri di uscita.
3, 4	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • ModuleType e SlotID sono entrambi non validi. Il modulo è fisicamente presente. • I bit di uscita Completato e Presente sono FALSE. • Il bit di uscita Errore è TRUE e deselezionato. • Aggiornare di conseguenza le informazioni sul modulo per ID modulo, ID fornitore, Tipo prodotto, codice prodotto e Revisione modulo.
5, 9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Tutti i parametri di uscita vengono cancellati.
6, 7, 10, 11	Nessuna modifica nella condizione ramo. Il bit Abilita è FALSE. Tutti i parametri di uscita vengono cancellati.

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

[MODULE_INFO](#) a pagina 380

PLUGIN_INFO (informazioni sui plug-in)

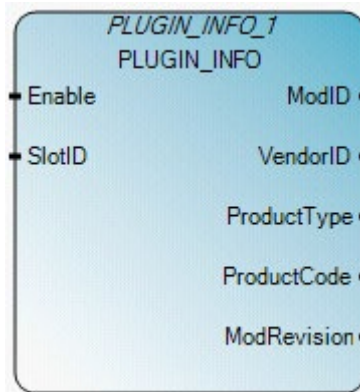
Leggere le informazioni del modulo da un modulo plug-in generico o di espansione.

Dettagli operazione:

- In Connected Components Workbench 10 o versione successive, l'istruzione PLUGIN_INFO consente di leggere qualsiasi informazione del modulo plug-in generico o di espansione, ad eccezione del modulo 2080-MEMBAK-RTC.
- Se non è presente un modulo plug-in generico o di espansione, tutti i valori sono reimpostati su zero (0).

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.

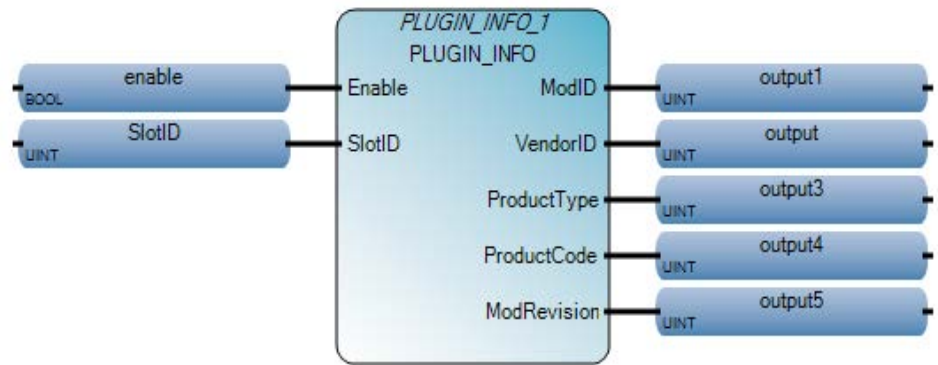


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

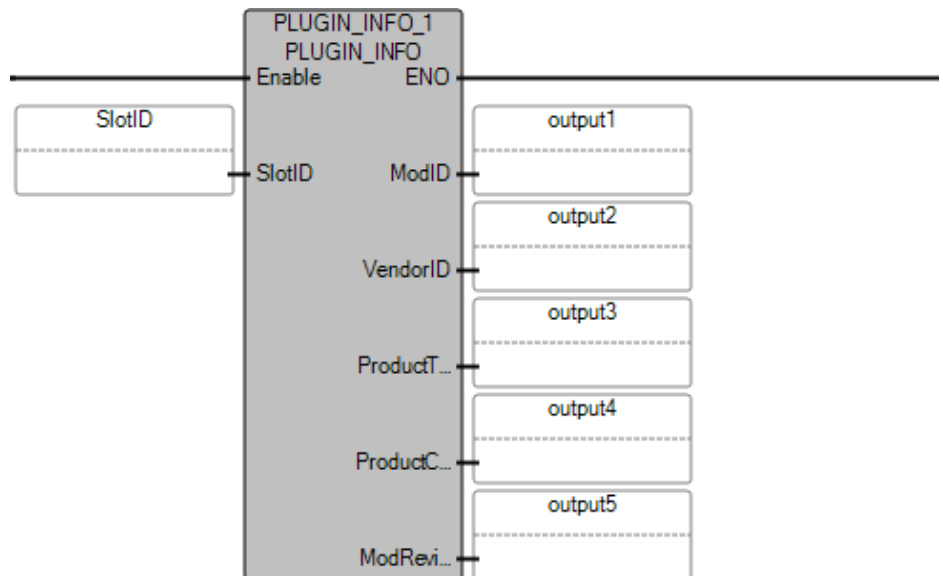
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue la lettura delle informazioni sui moduli plug-in o di espansione. FALSE: il blocco istruzione non viene eseguito. Tutti i valori dei dati di uscita sono ripristinati a 0.
SlotID	Ingresso	UINT	Numero di slot del plug-in: ID slot = 1,2,3,4,5 (Inizia dal primo Slot a sinistra = 1). Numero di slot di espansione: ID slot = 101, 102, 103, 104 (Inizia dal primo Slot a sinistra = 101).
ModID	Uscita	UINT	ID fisico del modulo generico plug-in. <ul style="list-style-type: none"> • Se il modulo di espansione non è supportato ModID = 0xFF • Se un modulo plug-in o di espansione non è presente ModID = 0x0000

VendorID	Uscita	UINT	L'ID fornitore del modulo plug-in o di espansione generico. Per i prodotti Allen Bradley, ID fornitore = 1. Se un modulo plug-in o di espansione non è presente VendorID = 0x0000
Tipo prodotto:	Uscita	UINT	Tipo di prodotto del modulo plug-in o di espansione generico. Se un modulo plug-in o di espansione non è presente ProductType = 0x0000
Codice Prodotto	Uscita	UINT	Codice prodotto del modulo plug-in o di espansione generico. Se un modulo plug-in o di espansione non è presente ProductCode = 0x0000
ModRevision	Uscita	UINT	Informazioni sulla revisione del modulo plug-in generico o di espansione. Se un modulo plug-in o di espansione non è presente ModRevision = 0x0000
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram. Diagramma ladder aggiunge automaticamente l'uscita ENO.

Esempio di Programmazione a blocchi funzionali PLUGIN_INFO



Esempio di Diagramma ladder PLUGIN_INFO



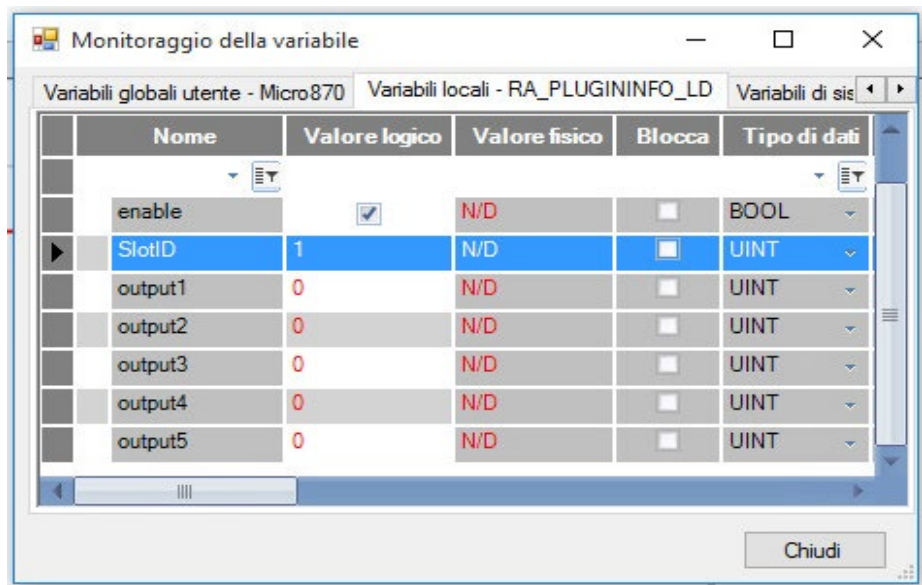
Esempio di testo strutturato PLUGIN_INFO

```

PLUGIN_INFO_1(
void PLUGIN_INFO_1(BOOL Enable, UINT SlotID)
Tipo: PLUGIN_INFO, Acquisire informazioni sul modulo da un modulo plug-in generico.

1: SlotID := 1;
2: PLUGIN_INFO_1(enable, SlotID);
3: output1 := PLUGIN_INFO_1.ModID;
4: output2 := PLUGIN_INFO_1.VendorID;
5: output3 := PLUGIN_INFO_1.ProductType;
6: output4 := PLUGIN_INFO_1.ProductCode;
7: output5 := PLUGIN_INFO_1.ModRevision;
    
```

Risultati



Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

PLUGIN_READ (lettura plug-in)

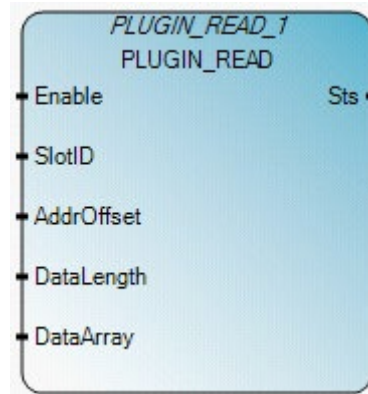
Legge i dati da un modulo plug-in generico.

Dettagli operazione:

- Qualsiasi modulo plug-in, eccetto i moduli 2080-MEMBAK-RTC.
- Quando non è presente un modulo plug-in generico, tutti i valori tornano a zero (0).

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

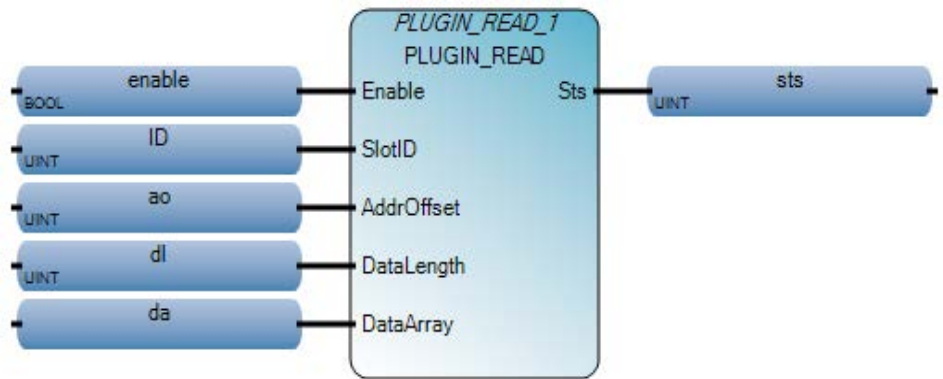
Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.



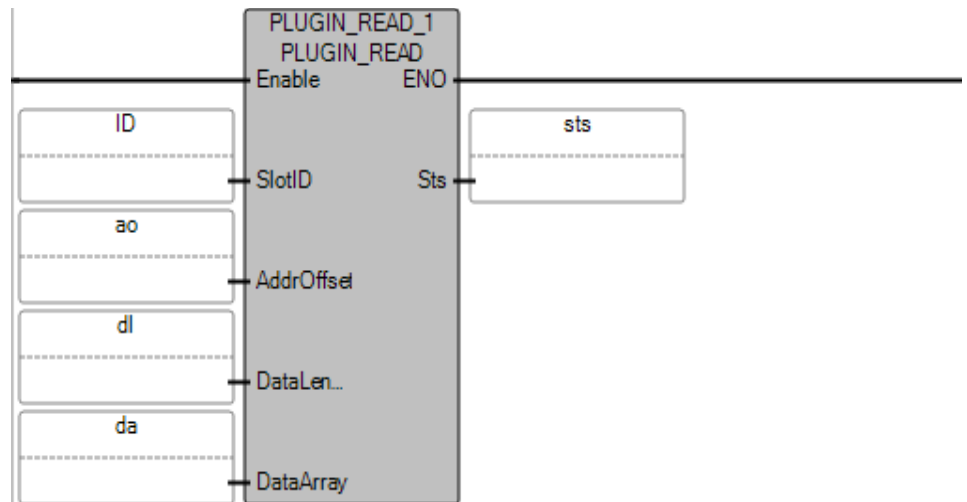
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue la lettura UPM. FALSE: nessuna lettura e i dati all'interno del vettore dati non sono validi.
SlotID	Ingresso	UINT	Numero dello slot plug-in. ID slot = 1,2,3,4,5 (iniziando dal primo slot a sinistra = 1).
Offset	Ingresso	UINT	Offset indirizzo dei primi dati da leggere, calcolando dal primo byte del modulo generico plug-in.
DataLength	Ingresso	UINT	Il numero di byte da leggere.
DataArray	Ingresso	USINT	Un array usato per archiviare i dati letti dal modulo a innesto generico.
Sts	Uscita	UINT	Codici di stato per PLUGIN_READ. Codici di stato (Sts): <ul style="list-style-type: none"> • 0x00 - Blocco funzione non abilitato (nessuna operazione). • 0x01 - Operazione plug-in riuscita. • 0x02 - Operazione plug-in non riuscita a causa di un ID slot non valido. • 0x03 - Operazione plug-in non riuscita perché non si tratta di un modulo plug-in generico valido. • 0x04 : - operazione plug-in non riuscita a causa di dati utilizzati fuori intervallo. • 0x05 - Operazione plug-in non riuscita a causa di un errore di parità in accesso dati.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Programmazione a blocchi funzionali PLUGIN_READ



Esempio di Diagramma ladder PLUGIN_READ



Esempio di Testo strutturato PLUGIN_READ

```

PLUGIN_READ_1
void PLUGIN_READ_1(BOOL Enable, UINT SlotID, UINT AddrOffset, UINT DataLength, USINT[1..1] DataArray)
Tipo: PLUGIN_READ, Leggere i dati da un modulo plug-in generico.

1 | PLUGIN_READ_1(enable, ID, ao, dl, da);
2 | sts := PLUGIN_READ_1.Sts;
    
```

Vedere anche

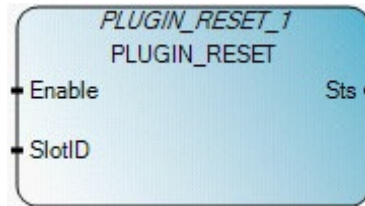
[Istruzioni ingresso/uscita](#) a pagina 355

PLUGIN_RESET (reset plug-in)

Ripristina qualsiasi hardware di modulo plug-in generico, ad eccezione dei moduli 2080-MEMBAK-RTC. Dopo il reset dell'hardware, il modulo plug-in generico è pronto per la configurazione e il funzionamento.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

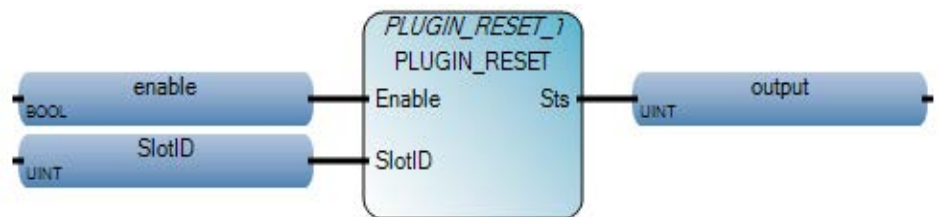
Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.



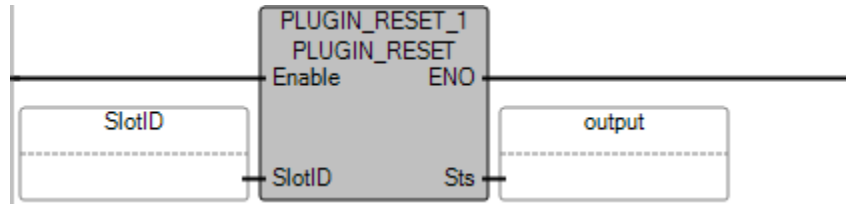
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il ripristino plug-in. FALSE: nessuna operazione di ripristino.
SlotID	Ingresso	UINT	Numero dello slot plug-in. ID slot = 1,2,3,4,5 (iniziando dal primo slot a sinistra = 1).
Sts	Uscita	UINT	Codici di stato PLUGIN_RESET. Codici di stato (Sts): <ul style="list-style-type: none"> • 0x00 - Blocco funzione non abilitato (nessuna operazione). • 0x01 - Operazione plug-in riuscita. • 0x02 - Operazione plug-in non riuscita a causa di un ID slot non valido. • 0x03 - Operazione plug-in non riuscita perché non si tratta di un modulo plug-in generico valido. La configurazione del modulo -2080-MOT-HSC è in modalità Istruzioni High Speed Counter. <ul style="list-style-type: none"> • 0x04 - Operazione plug-in non riuscita a causa di dati utilizzati fuori intervallo. • 0x05 - Operazione plug-in non riuscita a causa di un errore di parità in accesso dati.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Programmazione a blocchi funzionali PLUGIN_RESET



Esempio di Diagramma ladder PLUGIN_RESET



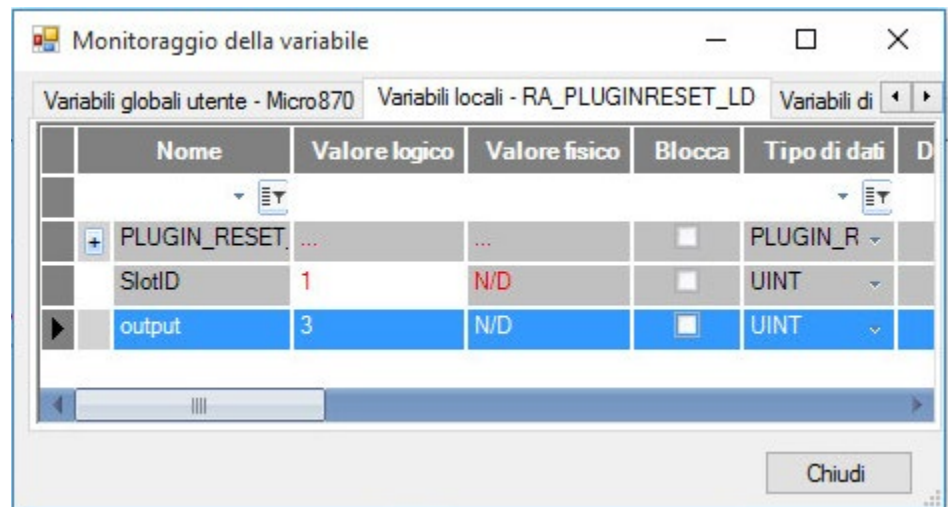
Esempio di Testo strutturato PLUGIN_RESET

```

PLUGIN_RESET_1 {
    void PLUGIN_RESET_1(BOOL Enable, UINT SlotID)
    Tipo: PLUGIN_RESET, Reimpostare un modulo plug-in generico (reset dell'hardware).

1 SlotID := 1;
2 PLUGIN_RESET_1(enable, SlotID);
3 output := PLUGIN_RESET_1.Sts;
    
```

Risultati



Vedere anche

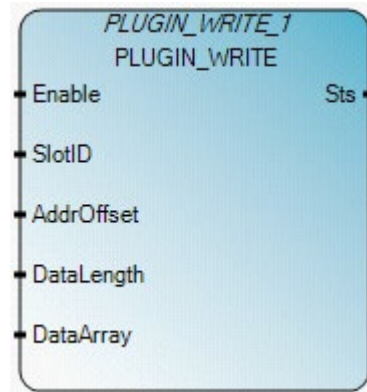
[Istruzioni ingresso/uscita](#) a pagina 355

PLUGIN_WRITE (plug-in di scrittura)

Scrive un blocco di dati su qualsiasi hardware di modulo plug-in generico, ad eccezione dei moduli 2080-MEMBAK-RTC.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

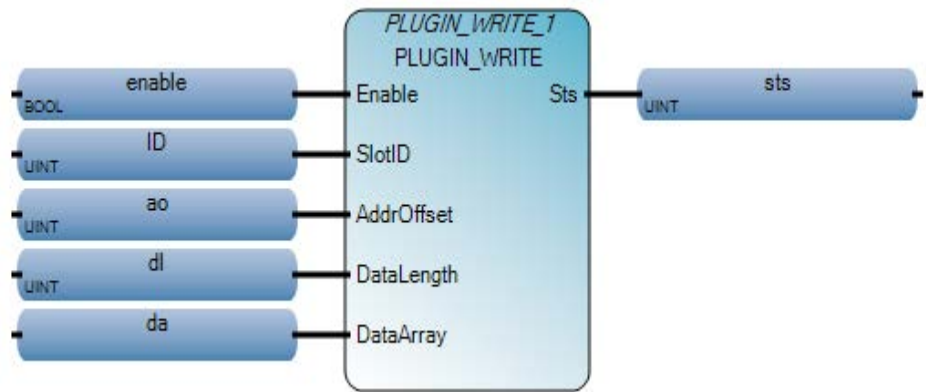
Questa istruzione vale per i controllori Micro820, Micro830, Micro850 e Micro870.



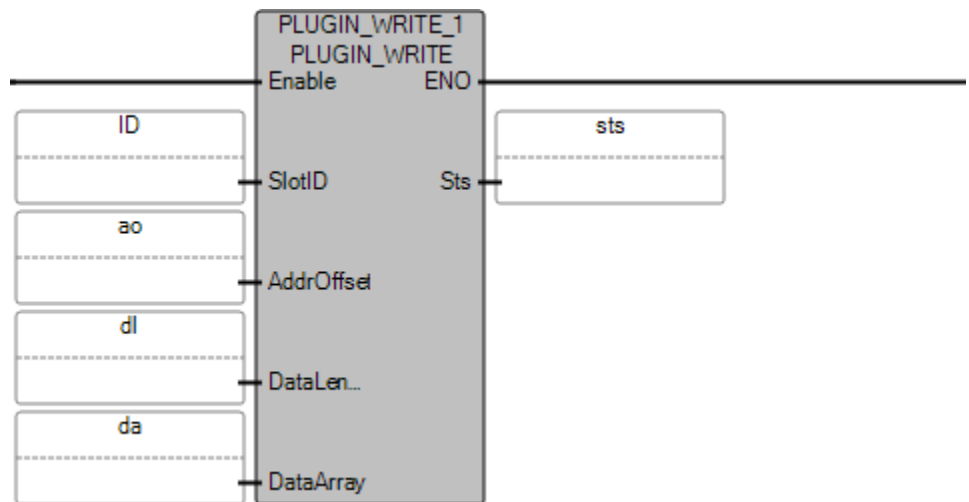
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il plug-in di scrittura. FALSE: nessuna operazione di scrittura dati.
SlotID	Ingresso	UINT	Numero dello slot plug-in. ID slot = 1,2,3,4,5 (iniziando dal primo slot a sinistra = 1).
AddrOffset	Ingresso	UINT	Offset indirizzo dei primi dati da scrivere, calcolando dal primo byte del modulo generico plug-in.
DataLength	Ingresso	UINT	Il numero di byte da scrivere.
DataArray	Ingresso	USINT	Dati da scrivere nel modulo generico plug-in.
Sts	Uscita	UINT	Codici di stato PLUGIN_WRITE. Codici di stato (Sts): <ul style="list-style-type: none"> • 0x00 - Blocco funzione non abilitato (nessuna operazione). • 0x01- Operazione plug-in riuscita. • 0x02 - Operazione plug-in non riuscita a causa di un ID slot non valido. • 0x03 - Operazione plug-in non riuscita perché non si tratta di un modulo plug-in generico valido. • 0x04 : - operazione plug-in non riuscita a causa di dati utilizzati fuori intervallo. • 0x05 - Operazione plug-in non riuscita a causa di un errore di parità in accesso dati.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Programmazione a blocchi funzionali PLUGIN_WRITE



Esempio di Diagramma ladder PLUGIN_WRITE



Esempio di Testo strutturato PLUGIN_WRITE

```

PLUGIN_WRITE_1 {
void PLUGIN_WRITE_1(BOOL Enable, UINT SlotID, UINT AddrOffset, UINT DataLength, USINT[1..1] DataArray)
Tipo: PLUGIN_WRITE, Scrivere i dati in un modulo plug-in generico.

1  PLUGIN_WRITE_1(enable, ID, ao, dl, da);
2  sts := PLUGIN_WRITE_1.Sts;
    
```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

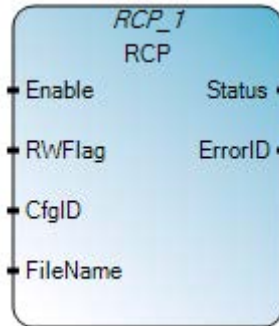
RCP (ricetta)

Legge i valori dei dati di una variabile dal file di dati ricetta che si trova nella cartella file di dati ricetta nella scheda SD e aggiorna il valore sul

motore di runtime. Scrive il valore della variabile con il motore di runtime in un file di dati ricetta nella scheda SD.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

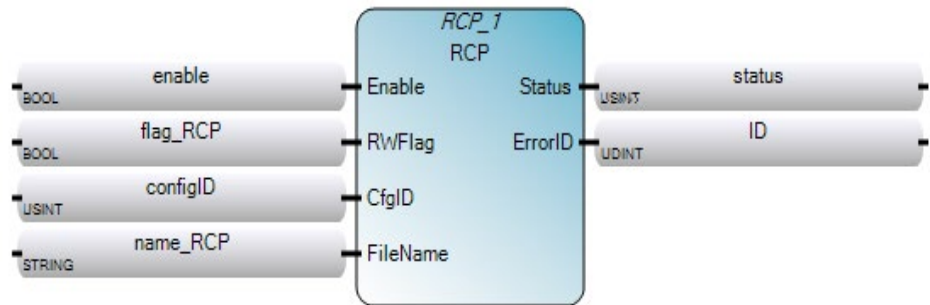
Parametri	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitare il blocco istruzione di lettura/scrittura della ricetta. TRUE: fronte di salita rilevato, esegue l'istruzione ricetta, se la precedente operazione è stata completata. FALSE: fronte di salita non rilevato, non esegue l'istruzione ricetta.
RWFlag	Ingresso	BOOL	TRUE: RWFlag (operazione di scrittura). La ricetta scrive i valori della variabile con il motore di runtime all'interno di un file di dati ricetta nella scheda SD. FALSE: RWFlag (operazione di lettura). La ricetta legge i valori della variabile dalla scheda SD e aggiorna il corrispondente valore della variabile sul motore di runtime.
CfgID	Ingresso	USINT	Numero configurazione della ricetta ID VA 1-10.
FileName	Ingresso	STRING	Nome file di dati ricetta (lunghezza massima 30 caratteri).
Stato	Uscita	USINT	Stato corrente del blocco istruzione Ricetta. Codici di stato RCP: <ul style="list-style-type: none"> • 0 Idle • 1 Doing • 2 Succeed, completato senza errori. • 3 Error, completato senza errori.
ErrorID	Uscita	UDINT	Codice di errore RCP numerico. Le definizioni sono definite nei codici di errore RCP.

Codici di errore RCP

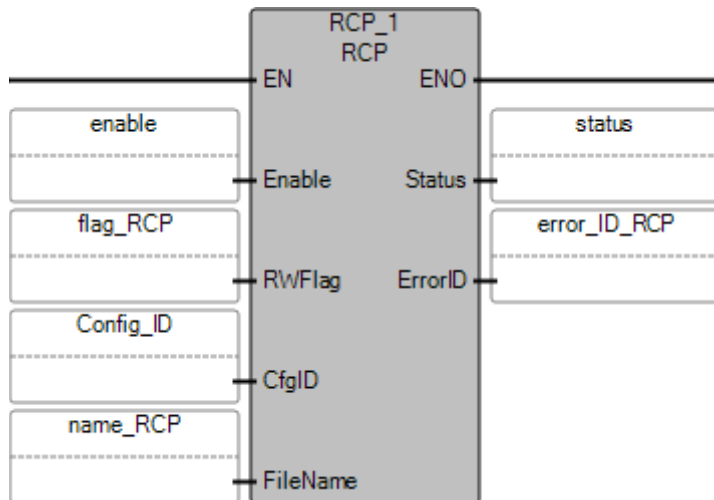
Codice errore	Nome errore
0	RCP_ERR_NONE
1	RCP_ERR_NO_SDCARD
2	RCP_ERR_DATAFILE_FULL
3	RCP_ERR_DATAFILE_ACCESS Una scheda SD viene identificata come: <ul style="list-style-type: none"> • Interrotta. • Piena. • Sola lettura.
4	RCP_ERR_CFG_ABSENT
5	RCP_ERR_CFG_ID
6	RCP_ERR_RESOURCE_BUSY
7	RCP_ERR_CFG_FORMAT

Codice errore	Nome errore
8	RCP_ERR_RESERVED Riservata per una possibile espansione futura.
9	RCP_ERR_UNKNOWN
10	RCP_ERR_DATAFILE_NAME
11	RCP_ERR_DATAFOLDER_INVALID
12	RCP_ERR_DATAFILE_ABSENT
13	RCP_ERR_DATAFILE_FORMAT
14	RCP_ERR_DATAFILE_SIZE La dimensione del file di dati ricetta è troppo grande (>4 kb).

Esempio di programmazione a blocchi funzionali RCP



Esempio di diagramma ladder RCP



Esempio di testo strutturato RCP

```

RCP_1 (
    void RCP_1(BOOL Enable, BOOL RWFlag, USINT CfgID, STRING FileName)
    Tipo: RCP, Salvare/Ripristinare un elenco di dati dal/al file ricetta della scheda SD.

1 | RCP_1 (EN, Enable, RWFlag, CfgID, FileName);
2 | output := RCP_1.ENO
3 | status := RCP_1.Status
4 | error_ID_RCP := RCP_1.ErrorID
    
```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

RTC_READ (lettura Real-Time Clock)

Legge le informazioni sul modulo real-time clock (RTC).

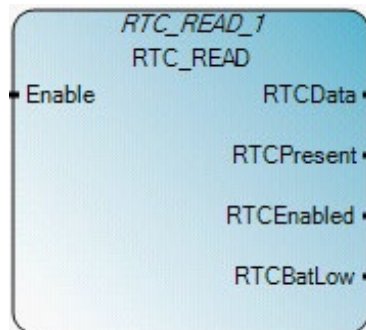
Dettagli operazione:

- Controllori Micro810 o Micro820 con RTC integrato:
 - RTCBatLow è sempre impostato su zero (0).
 - RTCEnabled è sempre impostato su (1).
- Quando l'RTC integrato perde la carica/memoria a causa di una interruzione dell'alimentazione:
 - RTCData è impostato su 2000/1/1/0/0/0.
 - RTCEnabled è impostato su (1).

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

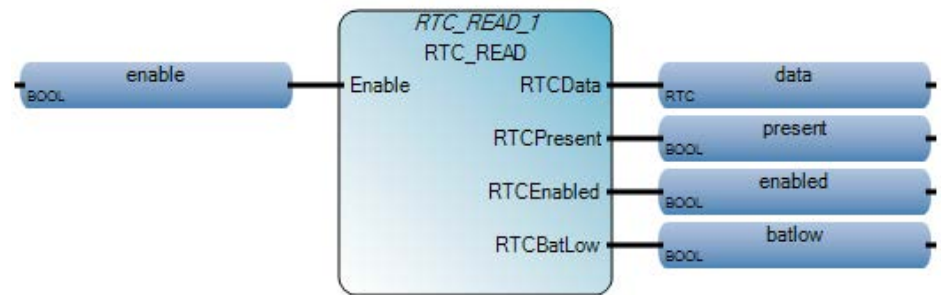
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue la lettura delle informazioni RTC. FALSE: non c'è nessuna operazione di lettura e i dati in uscita RTC non sono validi.
RTCData	Uscita	RTC	Informazioni sui dati RTC: aa/mm/gg, hh/mm/ss, settimana. L'uscita RTCData è definita con il tipo di dati RTC.
RTCPresent	Uscita	BOOL	TRUE: Free Running clock in uso oppure hardware RTC collegato. FALSE: Free Running clock non in uso oppure hardware RTC non collegato.
RTCEnabled	Uscita	BOOL	TRUE: Free Running clock in uso oppure hardware RTC abilitato (temporizzazione). FALSE: Free Running clock non in uso, hardware RTC disabilitato (nessuna temporizzazione).
RTCBatLow	Uscita	BOOL	TRUE - livello batteria RTC basso. FALSE - livello batteria RTC non è basso.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Tipo di dati RTC

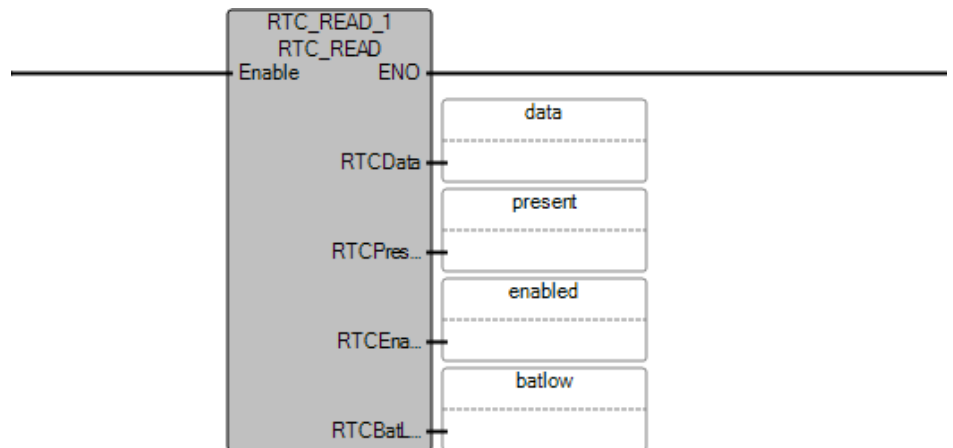
Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati RTC.

Parametri	Tipo di dati	Descrizione
Year	UINT	L'impostazione anno per RTC. Valore a 16 bit, intervallo valido da 2000 (01 gen, 00:00:00) a 2098 (31 dic, 23:59:59)
Month	UINT	L'impostazione mese per RTC.
Day	UINT	L'impostazione giorno per RTC.
Hour	UINT	L'impostazione ora per RTC.
Minute	UINT	L'impostazione minuto per RTC.
Second	UINT	L'impostazione secondi per RTC.
DayOfWeek	UINT	L'impostazione giorno della settimana per RTC. Questo parametro è ignorato per RTC.SET.

Esempio di Programmazione a blocchi funzionali RTC_READ



Esempio di Diagramma ladder RTC_READ



Esempio di Testo strutturato RTC_READ

```

RTC_READ_1 (
void RTC_READ_1(BOOL Enable)
Tipo: RTC_READ, Leggere le informazioni del modulo RTC.

1: RTC_READ_1(enable);
2: data := RTC_READ_1.RTCData;
3: present := RTC_READ_1.RTCPresent;
4: enabled := RTC_READ_1.RTCEnabled;
5: batlow := RTC_READ_1.RTCBatLow;
    
```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

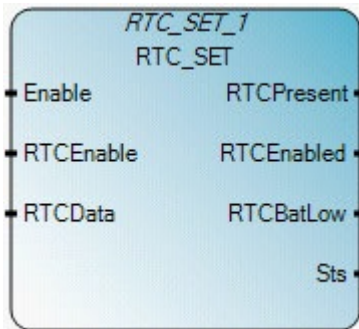
RTC_SET (impostare Real-Time Clock)

Impostare i dati RTC (Real-Time Clock) sulle informazioni del modulo RTC.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue RTC_SET con le informazioni su RTC dall'ingresso. Eseguito in genere per 1 scansione del programma durante l'aggiornamento di RTC. FALSE - non esegue RTC_SET. Impostare su FALSE per utilizzare RTC normalmente.
RTCEnable	Ingresso	BOOL	TRUE - Per abilitare RTC i dati RTC specificati. FALSE - Per disabilitare RTC.
RTCData	Ingresso	RTC	Informazioni sui dati RTC: aa/mm/gg, hh/mm/ss, settimana definita nel tipo di dati RTC. RTCData è ignorato quando RTCEnable = 0.
RTCPresent	Uscita	BOOL	TRUE: Free Running clock in uso oppure hardware RTC collegato. FALSE: Free Running clock non in uso oppure hardware RTC non collegato.

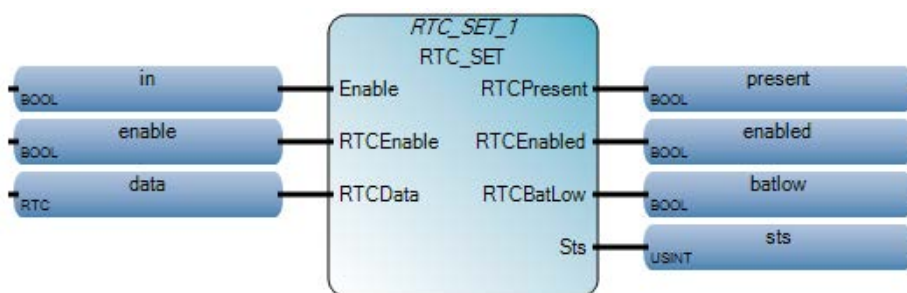
RTCEnabled	Uscita	BOOL	TRUE: Free Running clock in uso oppure hardware RTC abilitato (temporizzazione). FALSE: Free Running clock non in uso oppure hardware RTC disabilitato (nessuna temporizzazione).
RTCBatLow	Uscita	BOOL	TRUE - livello batteria RTC basso. FALSE - livello batteria RTC non è basso.
Sts	Uscita	USINT	Stato dell'operazione di lettura. Valori di stato (Sts) RTC_Set: <ul style="list-style-type: none"> • 0x00: blocco funzione non abilitato (nessuna operazione). • 0x01: operazione di impostazione RTC riuscita. • 0x02: operazione di impostazione RTC non riuscita.

Tipo di dati RTC

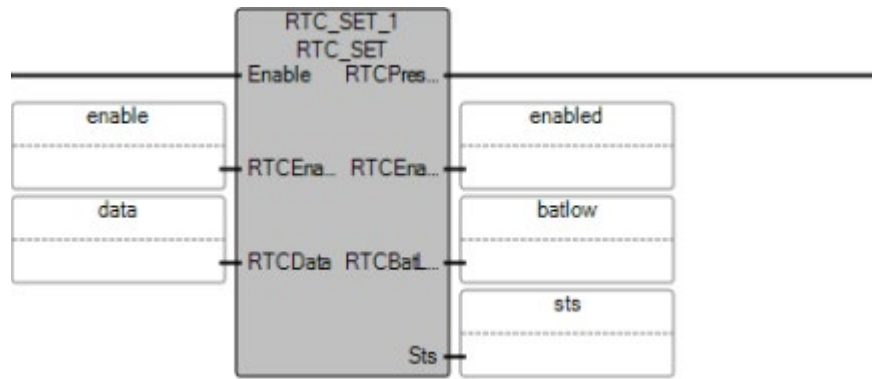
Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati RTC.

Parametri	Tipo di dati	Descrizione
Year	UINT	L'impostazione anno per RTC. Valore a 16 bit, intervallo valido da 2000 (01 gen, 00:00:00) a 2098 (31 dic, 23:59:59)
Month	UINT	L'impostazione mese per RTC.
Day	UINT	L'impostazione giorno per RTC.
Hour	UINT	L'impostazione ora per RTC.
Minute	UINT	L'impostazione minuto per RTC.
Second	UINT	L'impostazione secondi per RTC.
DayOfWeek	UINT	L'impostazione giorno della settimana per RTC. Questo parametro è ignorato per RTC_SET.

Esempio di Programmazione a blocchi funzionali RTC_SET



Esempio di Diagramma ladder RTC_SET



Esempio di Testo strutturato RTC_SET

```

RTC_SET_1(
void RTC_SET_1(BOOL Enable, BOOL RTCEnable, RTC RTCData)
Tipo: RTC_SET, Impostare i dati RTC sul modulo RTC.

1 RTC_SET_1(in, enable, data);
2 present := RTC_SET_1.RTCPresent;
3 enabled := RTC_SET_1.RTCEnable;
4 batlow := RTC_SET_1.RTCBatLow;
5 sts := RTC_SET_1.Sts;
    
```

Vedere anche

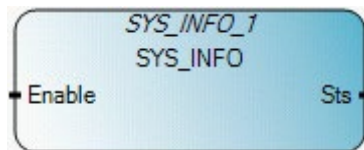
[Istruzioni ingresso/uscita](#) a pagina 355

SYS_INFO

Legge il blocco dati dello stato per il controllore Micro800.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

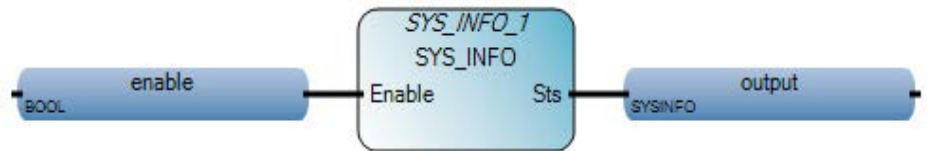
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



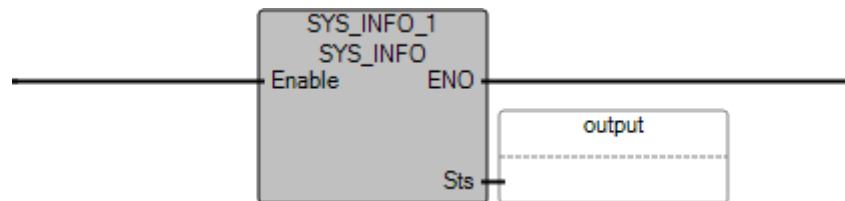
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue l'operazione di lettura. FALSE: non esegue la funzione.
Sts	Uscita	SYSINFO	Blocco dati dello stato del sistema. L'uscita Sts è definita nel tipo di dati SYS_INFO.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Programmazione a blocchi funzionali SYS_INFO



Esempio di Diagramma ladder SYS_INFO



Esempio di Testo strutturato SYS_INFO

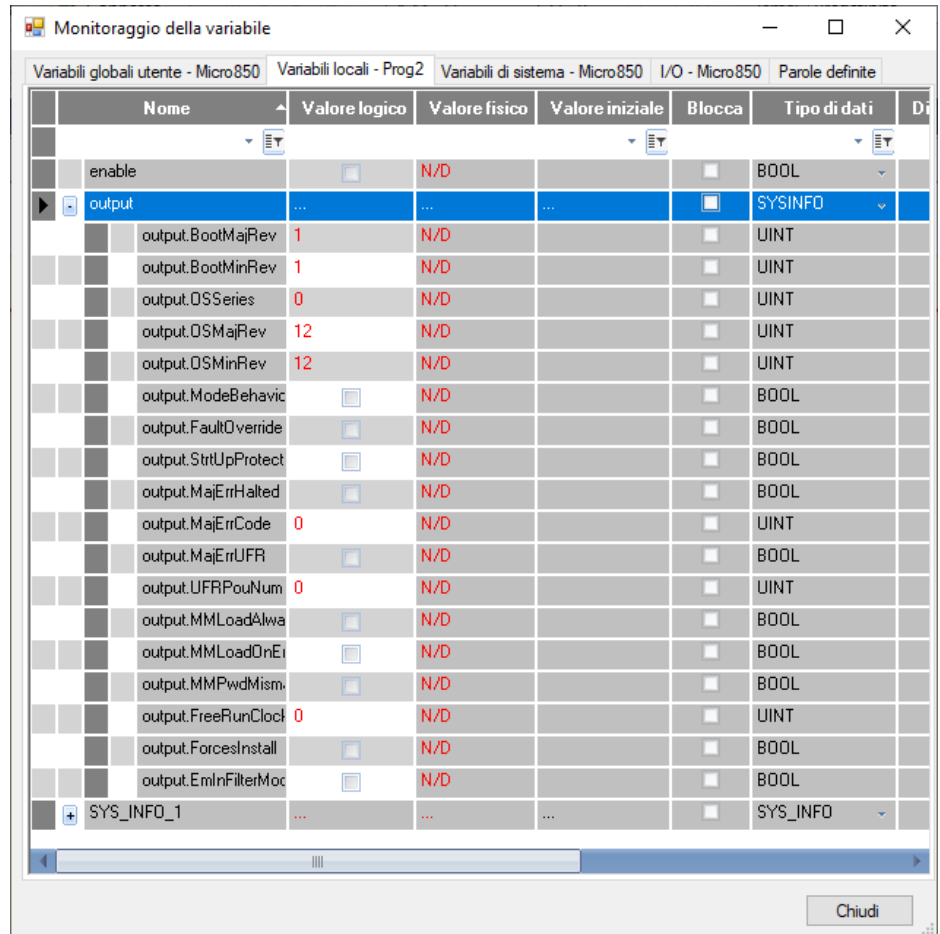
```

SYS_INFO_1 {
void SYS_INFO_1(BOOL Enable)
Tipo: SYS_INFO, Leggere lo stato del sistema Micro800.

1 SYS_INFO_1(enable);
2 output := SYS_INFO_1.Sts;

```

Risultati



Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

[Tipo di dati SYS_INFO](#) a pagina 409

Tipo di dati SYS_INFO

Nella tabella seguente è descritto il tipo di dati SYSINFO.

Parametri	Tipo di dati	Descrizione
BootMajRev	UINT	Revisione maggiore boot.
BootMinRev	UINT	Revisione minore boot.
Serie del sistema operativo:	UINT	Serie del sistema operativo: 0 indica un dispositivo di serie A 1 indica un dispositivo di serie B
OSMajRev	UINT	Revisione maggiore OS.
OSMinRev	UINT	Revisione minore OS.
ModeBehaviour	BOOL	Modalità di comportamento (TRUE: va a RUN all'accensione).
FaultOverride	BOOL	Fault Override (TRUE: errore di sostituzione all'accensione).

Parametri	Tipo di dati	Descrizione
StrtUpProtect	BOOL	Protezione all'avvio (TRUE: all'accensione, esegue programma di protezione in avvio). Per versioni future.
MajErrHalted	BOOL	Errore grave arrestato (TRUE: Errore grave arrestato).
MajErrCode	UINT	Codice di errore grave.
MajErrUFR	BOOL	Errore grave durante routine errore utente. Per versioni future.
UFRPouNum	UINT	Numero programma routine errore utente.
MMLoadAlways	BOOL	Ripristina sempre il modulo di memoria al controllore all'accensione (TRUE: Ripristina).
MMLoadOnError	BOOL	Ripristina sempre il modulo di memoria al controllore in caso di errore all'accensione (TRUE: Ripristina).
MMPwdMismatch	BOOL	Mancata corrispondenza password del modulo memoria (TRUE: Mancata corrispondenza password del controllore e del modulo memoria).
FreeRunClock	UINT	Free running clock con incrementi ogni 100 microsecondi da 0 a 65535 e successivo ritorno a 0. Il clock, accessibile a livello globale, può essere utilizzato in caso sia necessaria una risoluzione maggiore rispetto al timer standard da 1 millisecondo. Supportato solo dai controllori Micro830 e Micro850. Il valore per i controllori Micro810 resta 0.
ForcesInstall	BOOL	Forces installato (TRUE: Attivato).
EMINFilterMod	BOOL	Filtro incorporato modificato (TRUE: Modificato).

Vedere anche

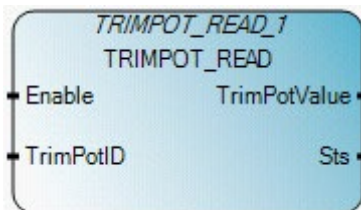
[SYS_INFO](#) a pagina 407

TRIMPOT_READ (lettura trimpot)

Legge il valore TrimPot da un trimpot specifico.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850 e Micro870.

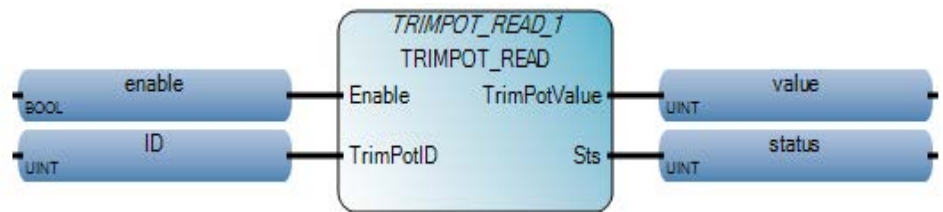


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

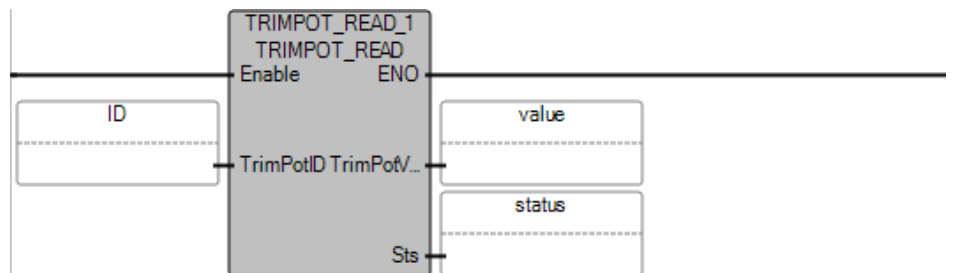
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue la lettura di Trimpot. FALSE: nessuna operazione di lettura e il valore di uscita TrimPot non è valido.
TrimPotID	Ingresso	UINT	L'ID del Trimpot da leggere. TrimPotID è definito nelle definizioni ID Trimpot.

TrimPotValue	Uscita	UINT	Valore trimpot corrente.
Sts	Uscita	UINT	Stato dell'operazione di lettura Trimpot. Codici dello stato (Sts) TRIMPOT: <ul style="list-style-type: none"> • 0x00 - Blocco funzione non abilitato (nessuna operazione di lettura/scrittura). • 0x01- Operazione di lettura/scrittura riuscita. • 0x02 - Operazione di lettura/scrittura non riuscita a causa di un ID Trimpot non valido. • 0x03 - Operazione di scrittura non riuscita a causa di un valore fuori intervallo.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Programmazione a blocchi funzionali TRIMPOT



Esempio di Diagramma ladder TRIMPOT



Esempio di Testo strutturato TRIMPOT

```

TRIMPOT_READ_1 (
void TRIMPOT_READ_1(BOOL Enable, UINT TrimPotID)
Tipo: TRIMPOT_READ, Leggere il valore TrimPot da un TrimPot specifico.

1 TRIMPOT_READ_1(enable, ID);
2 value := TRIMPOT_READ_1.TrimPotValue;
3 status := TRIMPOT_READ_1.Sts;

```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

[Definizioni ID Trimpot](#) a pagina 411

Definizione ID Trimpot

La tabella seguente descrive la definizione ID Trimpot utilizzata nell'istruzione TRIMPOT_read.

Selezione uscita	Bit	Descrizione
Definizione ID Trimpot	15 - 13	Modulo tipo di trimpot: <ul style="list-style-type: none"> • 0x00 - Integrato. • 0x01 - Espansione. • 0x02 - Porta plug-in.
	12 - 8	ID slot del modulo: <ul style="list-style-type: none"> • 0x00 - Integrato. • 0x01-0x1F - ID del modulo di espansione. • 0x01-0x05 - ID della porta plug-in.
	7 - 4	Tipo di trimpot: <ul style="list-style-type: none"> • 0x00 - Riservata. • 0x01 - Trimpot digitale tipo 1 (modulo LCD 1). • 0x02 - Modulo meccanico trimpot 1.
	3 - 0	ID trimpot all'interno del modulo: <ul style="list-style-type: none"> • 0x00-0x0F - Integrato. • 0x00-0x07 - ID trimpot per espansione. • 0x00-0x07 - ID trimpot per porta plug-in. ID trimpot inizia da 0.

Vedere anche

[TRIMPOT_READ](#) a pagina 410

Istruzioni di interruzione

Utilizzare le istruzioni di interrupt per segnalare al processore la necessità di considerare un evento. Il segnale di interrupt, di solito, è utilizzato per condizioni di elevata priorità per cui è richiesta l'interruzione del codice corrente mentre il processore è in funzionamento.

Funzione	Descrizione
STIS a pagina 413	Avvia il timer Selected Timed Interrupt (STI) utente dal programma di controllo invece che con l'avvio automatico.
UIC a pagina 415	Annulla la perdita di bit per l'interrupt utente selezionato.
UID a pagina 416	Disabilita un interrupt utente specifico.
UIE a pagina 418	Abilita un interrupt utente specifico.
UIF a pagina 420	Scarica o rimuove un ingresso utente in sospeso.

Vedere anche

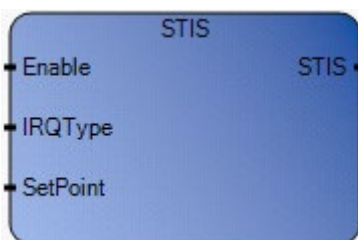
[Istruzione impostate in ordine alfabetico](#) a pagina 18

STIS (selezione avvio temporizzato)

Avvia il timer Selectable Timed Interrupt (STI) utente dal programma di controllo invece che con l'avvio automatico.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

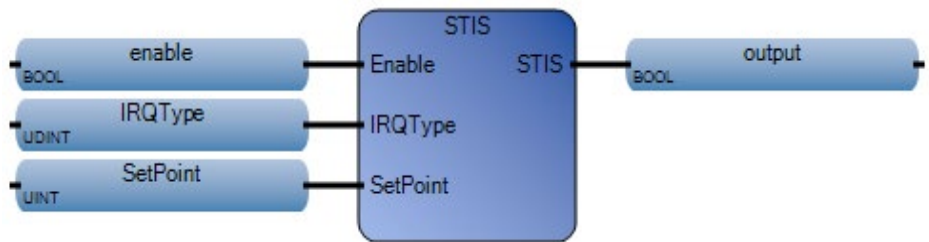


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

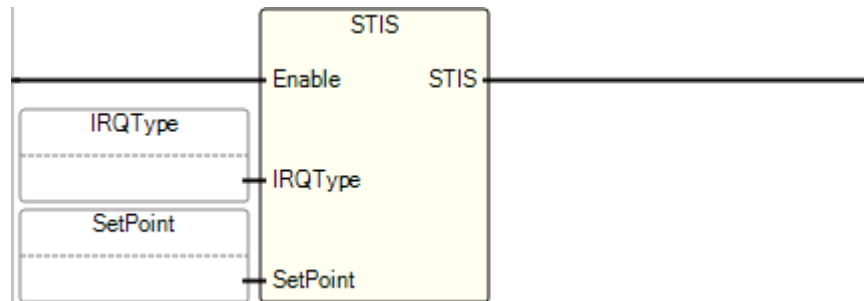
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione istruzioni. TRUE: avvia il timer STI dal programma di controllo. FALSE: non esegue la funzione.

IRQType	Ingresso	UDINT	Utilizzare le parole STI definite. - IRQ_STI0 - IRQ_STI1 - IRQ_STI2 - IRQ_STI3
SetPoint	Ingresso	UINT	Intervallo di tempo (in ms) che deve trascorrere prima di eseguire l'interrupt temporizzato selezionabile. Il valore 0 disabilita la funzione STIS. Un valore compreso tra 1 e 65535 abilita la funzione STIS.
STIS	Uscita	BOOL	Status del piolo (lo stesso di Abilita).

Esempio di programmazione a blocchi funzionali STIS



Esempio di diagramma ladder STIS



Esempio di testo strutturato STIS

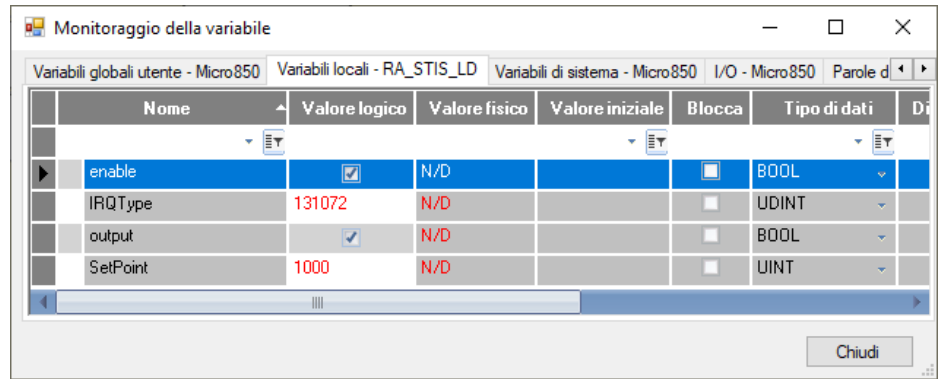
```
STIS (
  BOOL STIS(BOOL Enable, UDINT IRQType, UINT SetPoint)
  Avviare il timer STI dal programma di controllo (anziché automaticamente).
```

```
1 enable := TRUE;
2 IRQType := IRQ_STI2;
3 SetPoint := 1000;
4 output := STIS(enable, IRQType, SetPoint);
```

(* Equivalenza ST: *)

```
TESTOUTPUT := STIS(TESTENABLE, 2, 1000);
```

Risultati



Vedere anche

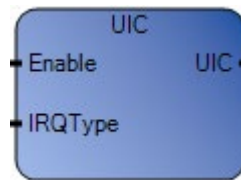
[Istruzioni di interrupt](#) a pagina 413

UIC (bit clear interrupt lost)

Cancella il bit perso per l'interrupt utente selezionato.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



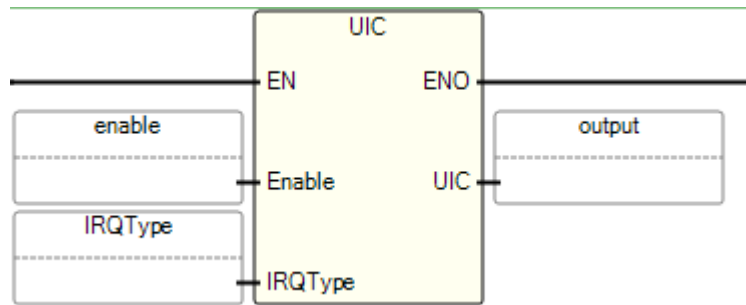
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione istruzioni. TRUE: avvia l'operazione Cancella bit. FALSE: non esegue la funzione.
IRQType	Ingresso	UDINT	Utilizzare le parole STI definite. - IRQ_HSC3 - IRQ_HSC4 - IRQ_HSC5 - IRQ_STI0 - IRQ_STI1 - IRQ_STI2 - IRQ_STI3 - IRQ_STI4 - IRQ_STI5 - IRQ_UFR - IRQ_UPM0 - IRQ_UPM1 - IRQ_UPM2 - IRQ_UPM3 - IRQ_UPM4
UIC	Uscita	BOOL	Status del piolo (lo stesso di Abilita).

Esempio di programmazione a blocchi funzionali UIC



Esempio di diagramma ladder UIC



Esempio di testo strutturato UIC

```

1  enable := TRUE;
2  IRQType := 2;
3  output := UIC (enable, IRQType);
    
```

Risultati

Monitoraggio della variabile						
Variabili globali utente - Micro870						
Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
output		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
IRQType		2	N/D		<input type="checkbox"/>	UDINT
enable		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL

Vedere anche

[Istruzioni di interrupt](#) a pagina 413

Disabilita uno o più interrupt utente specifici.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

UID (disattivazione interrupt)

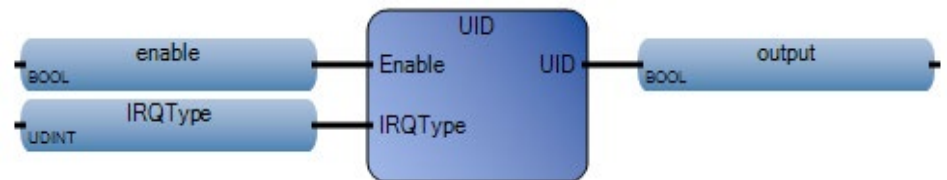
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



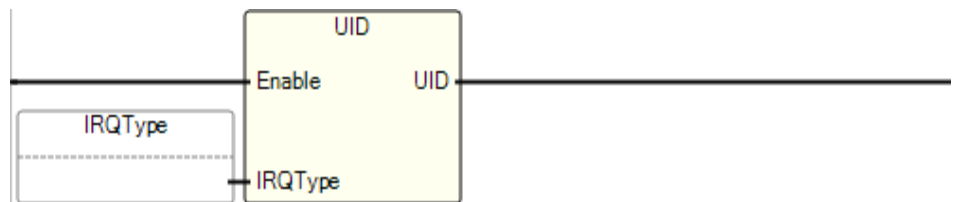
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione istruzioni. TRUE: avvia l'operazione di disattivazione. FALSE: non esegue la funzione.
IRQType	Ingresso	UDINT	Utilizzare le parole STI definite. - IRQ_EI10 - IRQ_EI11 - IRQ_EI12 - IRQ_EI13 - IRQ_EI14 - IRQ_EI15 - IRQ_EI16 - IRQ_EI17 - IRQ_HSC0 - IRQ_HSC1 - IRQ_HSC2 - IRQ_HSC3 - IRQ_HSC4 - IRQ_HSC5 - IRQ_ST10 - IRQ_ST11 - IRQ_ST12 - IRQ_ST13 - IRQ_UFR - IRQ_UPM0 - IRQ_UPM1 - IRQ_UPM2 - IRQ_UPM3 - IRQ_UPM4
UID	Uscita	BOOL	Status del piolo (lo stesso di Abilita).

Esempio di programmazione a blocchi funzionali UID



Esempio di diagramma ladder UID



Esempio di testo strutturato UID

```
UID (
  BOOL UID(BOOL Enable, UDINT IRQType)
  Disabilitare interrupt utente specifico.
```

```
1 enable := TRUE;
2 IRQType := 2;
3 output := UID(enable, IRQType);
```

(* Equivalenza ST: *)

```
TESTOUTPUT := UID(TESTENABLE, 2);
```

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di da
output	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
IRQType	2	N/D		<input type="checkbox"/>	UDINT
enable	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL

Vedere anche

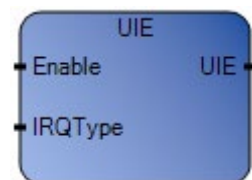
[Istruzioni di interrupt](#) a pagina 413

UIE (abilitazione interrupt)

Abilita un interrupt utente specifico.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

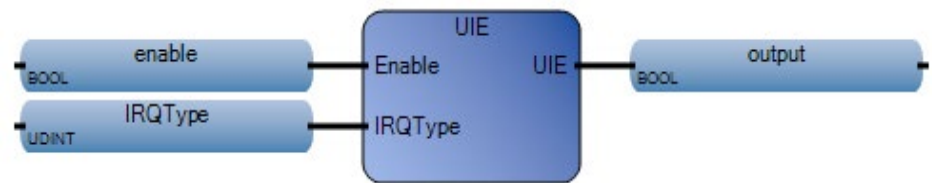
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



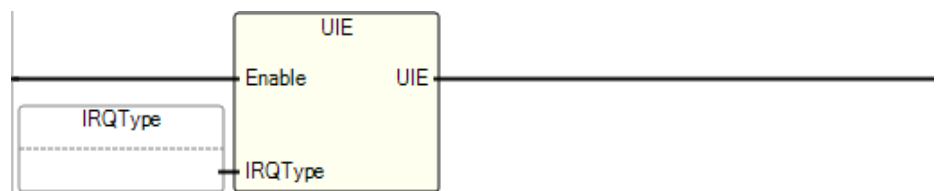
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione istruzioni. TRUE: avvia l'operazione di abilitazione. FALSE: non esegue la funzione.
IRQType	Ingresso	UDINT	Utilizzare le parole STI definite. - IRQ_EI10 - IRQ_EI11 - IRQ_EI12 - IRQ_EI13 - IRQ_EI14 - IRQ_EI15 - IRQ_EI16 - IRQ_EI17 - IRQ_HSC0 - IRQ_HSC1 - IRQ_HSC2 - IRQ_HSC3 - IRQ_HSC4 - IRQ_HSC5 - IRQ_ST10 - IRQ_ST11 - IRQ_ST12 - IRQ_ST13 - IRQ_UFR - IRQ_UPM0 - IRQ_UPM1 - IRQ_UPM2 - IRQ_UPM3 - IRQ_UPM4
UIE	Uscita	BOOL	Status del piolo (lo stesso di Abilita).

Esempio di programmazione a blocchi funzionali UIE



Esempio di diagramma ladder UIE



Esempio di testo strutturato UIE

```

UIE (
  BOOL UIE(BOOL Enable, UDINT IRQType)
  Abilitare interrupt utente specifico.

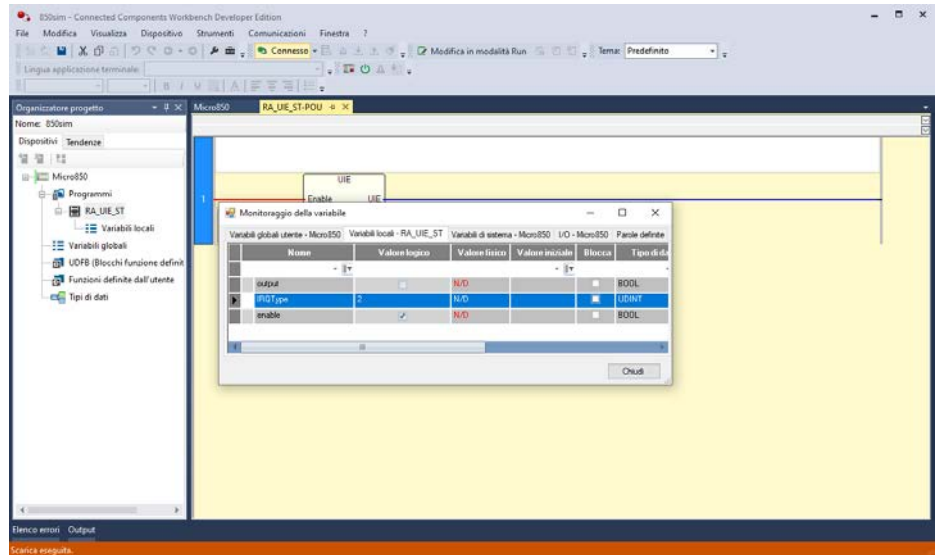
1: enable := TRUE;
2: IRQType := 2;
3: output := UIE(enable, IRQType);

```

(* Equivalenza ST: *)

TESTOUTPUT := UIE(TESTENABLE, 2) ;

Risultati



Vedere anche

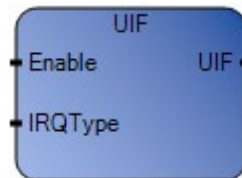
[Istruzioni di interrupt](#) a pagina 413

UIF (interrupt flush in attesa)

Scarica o rimuove un interrupt utente in sospeso.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

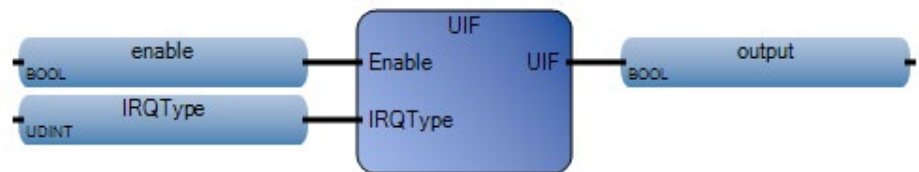


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

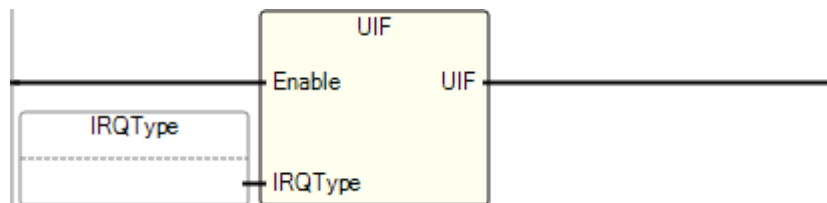
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione istruzioni. TRUE: avvia l'operazione UIF. FALSE: non esegue la funzione.

IRQType	Ingresso	UDINT	Utilizzare le parole STI definite. - IRQ_EI10 - IRQ_EI11 - IRQ_EI12 - IRQ_EI13 - IRQ_EI14 - IRQ_EI15 - IRQ_EI16 - IRQ_EI17 - IRQ_HSC0 - IRQ_HSC1 - IRQ_HSC2	- IRQ_HSC3 - IRQ_HSC4 - IRQ_HSC5 - IRQ_ST10 - IRQ_ST11 - IRQ_ST12 - IRQ_ST13 - IRQ_UFR - IRQ_UPM0 - IRQ_UPM1 - IRQ_UPM2 - IRQ_UPM3 - IRQ_UPM4
UIF	Uscita	BOOL	Status del piolo (lo stesso di Abilita).	

Esempio di programmazione a blocchi funzionali UIF



Esempio di diagramma ladder UIF



Esempio di testo strutturato UIF

```

UIF (
  BOOL UIF(BOOL Enable, UDINT IRQType)
  Scaricare interrupt utente specifico.

1  enable := TRUE;
2  IRQType := 2;
3  output := UIF(enable, IRQType);

```

(* Equivalenza ST: *)

```

TESTOUTPUT := UIF(TESTENABLE, 2);

```

Risultati

The screenshot shows a window titled "Monitoraggio della variabile" with a tabbed interface. The active tab is "Variabili locali - RA_UIF_ST". Below the tabs is a table with the following columns: "Nome", "Valore logico", "Valore fisico", "Valore iniziale", "Blocca", and "Tipo di da". The table contains three rows of data:

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di da
enable	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
IRQType	2	N/D		<input checked="" type="checkbox"/>	UDINT
output	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL

At the bottom right of the window is a "Chiudi" button.

Vedere anche

[Istruzioni di interrupt](#) a pagina 413

Istruzioni di controllo del movimento

Utilizzare le istruzioni Controllo movimento per programmare e progettare il movimento di un particolare asse. Controllo movimento richiede Connected Components Workbench 2.0 o versione successiva.



Suggerimento:

- le funzioni amministrative supportano il controllo assi PTO e Motion feedback.
- Le istruzioni di controllo movimento che supportano un FBAxis sono: MC_ReadActualPosition e MC_ReadActualVelocity.

Istruzione	Descrizione
<i>Administrative</i>	
MC_AbortTrigger a pagina 434	Interrompe i blocchi funzione Controllo assi collegati agli eventi di attivazione.
MC_Power a pagina 457	Controllo dello stato di alimentazione (ON o OFF).
MC_ReadAxisError a pagina 468	Legge gli errori dell'asse non in relazione ai blocchi funzione Controllo movimento.
MC_ReadBoolParameter a pagina 473	Restituisce il valore di un parametro specifico del produttore di tipo BOOL.
MC_ReadParameter a pagina 475	Restituisce il valore di un parametro specifico del fornitore di tipo Real.
MC_ReadStatus a pagina 478	Restituisce lo stato dell'asse rispetto al movimento attualmente in corso.
MC_Reset a pagina 484	Trasforma lo stato dell'asse da ErrorStop a StandStill ripristinando tutti gli errori interni legati all'asse.
MC_SetPosition a pagina 486	Sposta il sistema delle coordinate di un asse tramite la manipolazione della posizione effettiva.
MC_TouchProbe a pagina 492	Registra una posizione dell'asse in corrispondenza di un evento di attivazione.
MC_WriteBoolParameter a pagina 496	Modifica il valore di un parametro specifico del fornitore di tipo BOOL.
MC_WriteParameter a pagina 499	Modifica del valore di un parametro specifico del fornitore di tipo REAL.
<i>Controllo assi</i>	
MC_Halt a pagina 437	Comanda un arresto controllato del movimento in condizioni operative normali.
MC_Home a pagina 440	Comanda l'asse per l'esecuzione della sequenza <i><search home></i> .
MC_MoveAbsolute a pagina 445	Determina un movimento controllato verso una posizione assoluta specificata.
MC_MoveRelative a pagina 448	Determina un movimento controllato di una distanza specificata rispetto alla posizione attuale nel momento dell'esecuzione.
MC_MoveVelocity a pagina 453	Determina un movimento controllato infinito a una velocità specificata.
MC_ReadActualPosition a pagina 460	Restituisce la posizione effettiva dell'asse di feedback. Richiede Connected Components Workbench 8 o versione successiva.
MC_ReadActualVelocity a pagina 464	Ritorna alla velocità corrente dell'asse di feedback. Richiede Connected Components Workbench 8 o versione successiva.
MC_Stop a pagina 489	Comanda un arresto di movimento controllato e trasferisce lo stato dell'asse su Arresto in corso.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

Regole generali per i blocchi funzione di controllo movimento

Le regole generali per i blocchi funzione di controllo movimento Micro800 seguono le specifiche di controllo movimento PLCopen. La seguente tabella fornisce le regole generali sull'interfaccia dei blocchi funzione di controllo movimento.

La regola si applica a	Regola
Parametri di ingresso	<p>Con Execute: i parametri vengono utilizzati con il fronte di salita dell'input di esecuzione. Per modificare i parametri, cambiare i parametri di input e attivare o disattivare il movimento.</p> <p>se un'istanza di un blocco funzione riceve un nuovo comando Esegui prima del completamento (con una serie di comandi sulla stessa istanza), il nuovo comando Esegui viene ignorato e continua l'esecuzione dell'istruzione precedente.</p> <p>Con Enable: i parametri vengono utilizzati con il fronte di salita dell'input di abilitazione e possono essere modificati continuamente.</p>
Parametri di input mancanti	Gli input mancanti verranno catturati durante la compilazione dell'applicazione utente. A livello di controllore la gestione degli errori di input mancanti non avviene.
Input che superano i limiti dell'applicazione	Se un blocco funzione riceve i comandi mediante parametri che provocano la violazione dei limiti dell'applicazione, l'istanza del blocco funzione genera un errore. In tal caso, viene contrassegnata la presenza di un errore, le cui informazioni verranno indicate dall'ID dell'errore di uscita. Nella gran parte dei casi il controllore rimane in modalità di esecuzione e non vengono riportati errori gravi per il controllore.
Regole del segno per gli input	Gli input di accelerazione, decelerazione e strappo sono sempre valori positivi. Input di velocità, posizione e distanza possono avere valori positivi e negativi.
Posizione e distanza	Il valore di posizione è definito in un sistema di coordinate. La distanza è una misura relativa rapportata alle unità tecniche. La distanza è la differenza tra due posizioni.
Input di posizione/distanza	I controllori Micro800 supportano solo il movimento lineare. Per il blocco funzione MC_MoveAbsolute, l'input di posizione rappresenta la posizione assoluta per i comandi sugli assi. Per MC_MoveRelative, l'input di distanza rappresenta la collocazione relativa (considerando la posizione attuale dell'asse come 0) rispetto alla posizione attuale.
Input di velocità	La velocità può essere un valore con segno, tuttavia è possibile utilizzare l'input di direzione per definire il segno della velocità (velocità negativa x direzione negativa = velocità positiva). La "direzione" del parametro E si riferisce all'ingresso e all'uscita di velocità per motivi di compatibilità.
Input di direzione	<p>Per il movimento di distanza (posizione), per cui è definita la posizione di destinazione (assoluta o relativa), la direzione di movimento è univoca. L'input di direzione per il movimento di distanza viene ignorato.</p> <p>Per il movimento di velocità, il valore dell'input di direzione può essere 1 (direzione positiva), 0 (direzione attuale) o -1 (direzione negativa). Per gli altri valori, viene preso in considerazione solo il segno. Ad esempio, -3 indica una direzione negativa, +2 una direzione positiva e così via.</p> <p>Per il movimento di velocità (MC_MoveVelocity), il segno (velocità x direzione) determina la direzione effettiva del movimento se il valore è diverso da 0. Ad esempio, se velocità x direzione = +300, la direzione è positiva.</p>
Input di accelerazione, decelerazione e Jerk	<ul style="list-style-type: none"> • Gli ingressi di accelerazione e decelerazione devono avere un valore positivo. Se decelerazione o accelerazione sono impostati con un valore non positivo, il blocco funzione riporterà un errore (ID errore: MC_FB_ERR_RANGE). • L'ingresso del jerk deve avere un valore non negativo. Se lo strappo è impostato a un valore negativo, il blocco funzione riporterà un errore (ID errore: MC_FB_ERR_RANGE). • Se il jerk massimo è configurato su zero, tutti i parametri del jerk per il blocco funzione del controllo movimento, inclusa l'impostazione jerk per MC_Stop, devono essere configurati su 0. In caso contrario, il blocco funzione riporterà un errore (ID errore: MC_FB_ERR_RANGE). • Se il jerk è impostato su un valore diverso da zero, viene generato un profilo di formato curva a S, mentre se il jerk è impostato su 0, viene generato un profilo trapezoidale. • La configurazione di jerk iniziale non è limitata alla configurazione di jerk massimo. • Se il motore movimento non riesce a generare il profilo di movimento stabilito dai parametri di ingresso dinamici, il blocco funzione riporta un errore (ID errore: MC_FB_ERR_PROFILE).

La regola si applica a	Regola
Esclusività dell'output	<p>Con Execute: quando Execute è TRUE, anche uno degli output Busy, Done, Error o CommandAborted deve essere TRUE. Le uscite si escludono a vicenda: su un blocco funzione solo una di esse può essere TRUE. È impostata solo uno degli output Active, Error, Done e CommandAborted per volta.</p> <p>Con Enable: gli output Valid ed Error si escludono a vicenda: su un blocco funzione solo una di esse può essere TRUE.</p>
Stato output	<p>Con Execute: gli output Done, Error, ErrorID e CommandAborted sono reimpostati con il fronte di discesa dell'istruzione Execute. Tuttavia, il fronte di discesa di Execute non arresta né influisce sull'esecuzione del blocco funzione effettivo. Anche se Execute viene reimpostata prima che il blocco funzione sia completato, gli output corrispondenti sono impostati per almeno un ciclo.</p> <p>Se un'istanza di un blocco funzione riceve un nuovo comando Execute prima del completamento (con una serie di comandi sulla stessa istanza), il nuovo comando Execute viene ignorato e continua l'esecuzione dell'istruzione precedente.</p> <p>Con Enable: gli output Valid, Enabled, Busy, Error ed ErrorID vengono reimpostati appena possibile con il fronte di discesa di Enable.</p>
Comportamento con l'output Done	<p>L'output Done viene impostato quando viene completata l'azione comandata.</p> <p>Quando sono in funzione più blocchi funzione in sequenza sullo stesso asse, si verifica quanto segue:</p> <ul style="list-style-type: none"> • Quando un movimento su un asse viene interrotto con un altro movimento sullo stesso asse, senza aver raggiunto l'obiettivo finale, sul primo blocco funzione non viene impostato Done.
Comportamento dell'uscita Busy	<p>Ciascun blocco funzione può avere un output Busy, che indica che il blocco funzione non è ancora terminato (per i blocchi funzione con un input Execute) o che non è in funzione ed è possibile aspettarsi nuovi valori di output (nel caso di input Enable).</p> <p>L'output Busy è impostato sul fronte di salita di Execute e viene reimpostato quando si imposta uno degli output Done, Aborted o Error. Si consiglia di tenere questo blocco funzione nel loop attivo del programma applicativo finché l'output Busy è TRUE, dato che gli output possono ancora cambiare.</p> <p>I blocchi funzione con la stessa istanza di quelli busy non possono essere eseguiti finché sono busy. I blocchi funzione con diverse istanze possono interrompere il blocco funzione in esecuzione.</p>
Comportamento dell'uscita CommandAborted	<p>L'output CommandAborted viene impostato quando un movimento impartito viene interrotto da un altro comando di movimento.</p> <p>Il comportamento di reimpostazione di CommandAborted è simile a quello dell'output Done. Quando si verifica un CommandAborted, gli altri segnali di output quali InVelocity vengono reimpostati.</p>
Output Active	<p>L'output Active è necessario per i blocchi funzione in buffer, e viene impostato nel momento in cui il blocco funzione prende il controllo movimento dell'asse corrispondente.</p> <p>Per la modalità senza buffer, gli output Active e Busy possono avere lo stesso valore.</p>
Stati Enable e Valid	<p>L'input Enable è associato a un output Valid. Enable è sensibile al livello, Valid invece mostra che è disponibile un gruppo di output validi sul blocco funzione.</p> <p>L'output Valid è TRUE finché è disponibile un valore di output valido e l'input Enable è TRUE. Il relativo valore di output può essere aggiornato fino a quando l'input Enable è TRUE.</p> <p>Se si verifica un errore del blocco funzione, l'uscita non è valida (Valid impostato su FALSE). Quando la condizione di errore scompare, i valori ricompaiono e l'uscita Valid viene impostata di nuovo.</p>
Gestione errori di uscita	<p>Uscite utilizzate per definire gli errori</p> <p>Tutti i blocchi hanno le due uscite descritte di seguito, le quali gestiscono gli errori che potrebbero verificarsi durante l'esecuzione:</p> <ul style="list-style-type: none"> • Error - il fronte di salita di "Error" informa che si è verificato un errore durante l'esecuzione del blocco funzione. • ErrorID - numero dell'errore. <p>gli output Done e InVelocity servono per il completamento senza errori, pertanto dal punto di vista logico escludono Error. Gli errori delle istanze non sempre danno luogo a errori di assi (causando un ErrorStop dell'asse).</p> <p>Come ripristinare le uscite degli errori</p> <ul style="list-style-type: none"> • Le uscite degli errori che riguardano importanti blocchi funzione vengono reimpostate con fronte di discesa Execute ed Enable. • Le uscite degli errori dei blocchi funzione con Enable possono essere reimpostate durante il funzionamento, senza dover reimpostare Enable.

La regola si applica a	Regola
	<p>Tipi di errori</p> <ul style="list-style-type: none"> • Logica dei blocchi funzione (ad esempio, parametri fuori intervallo, tentativi di violazione dello stato e così via) • Limite HW o SW • Meccanismo/Motore • Unità
<p>Convenzioni per la denominazione dei tipi ENUM</p>	<p>A causa dei limiti nella denominazione previsti dallo standard IEC relativo alla univocità dei nomi delle variabili, per gli ENUM viene utilizzato il riferimento 'mc' allo spazio dei nomi per il controllo movimento PLCopen.</p> <p>In tal modo si evita il conflitto derivante dall'utilizzo dei tipi ENUM 'positive' e 'negative' per istanze con variabili con tali nomi per tutto il resto del progetto, dato che vengono denominate rispettivamente mcPositive e mcNegative.</p>

Vedere anche

[Istruzioni di controllo del movimento](#) a pagina 423

Dettagli relativi ai parametri del blocco funzione per il controllo del movimento

I seguenti argomenti contengono dettagli sui parametri del controllo del movimento rilevanti per tutti i relativi blocchi funzione.

[Stati asse di controllo movimento](#) a pagina 426

[Numeri dei parametri del blocco funzione per il controllo del movimento](#) a pagina 429

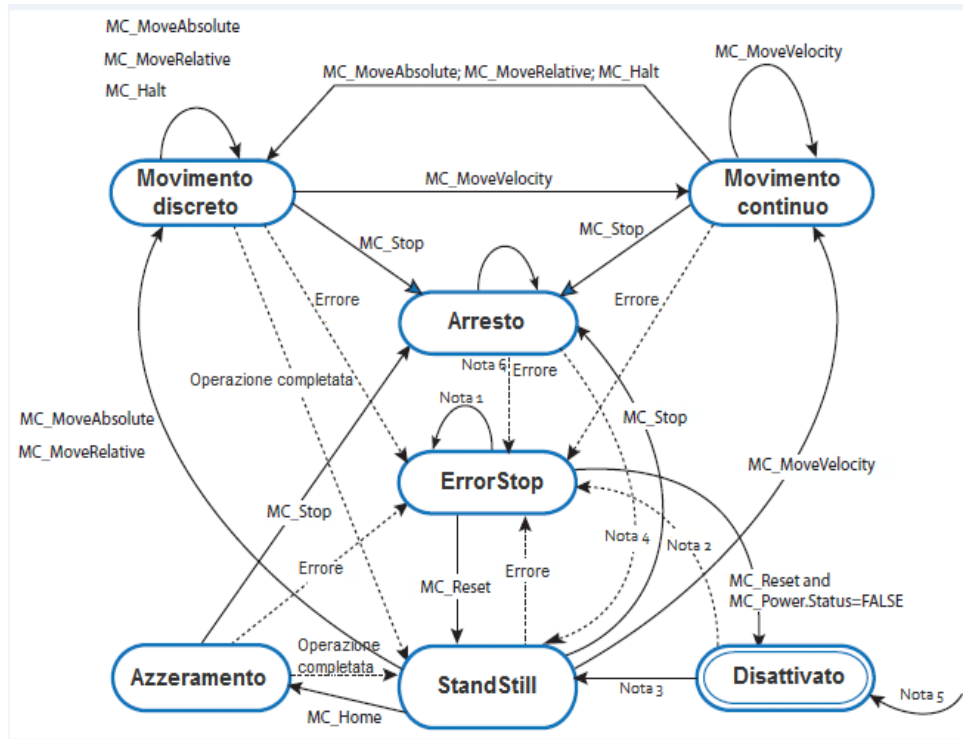
[ID errori del blocco funzione di controllo movimento](#) a pagina 430

Stati asse di controllo movimento

Secondo la regola di base per il comportamento dell'asse ad alto livello, quando sono attivati più blocchi funzione per il controllo movimento, i comandi del movimento sono accettati sempre in modo sequenziale, anche se il controllore ha la capacità di elaborare contemporaneamente i valori reali. Qualsiasi comando di movimento è un transitorio che modifica lo stato dell'asse e, di conseguenza, modifica il modo in cui viene calcolato il movimento corrente.

Diagramma di stato dell'asse di controllo movimento

L'asse è sempre in uno degli stati definiti, come descritto nel seguente diagramma.



Comportamento dello stato dell'asse di controllo movimento

No	Nota
1	Negli stati ErrorStop e Stopping, possono essere chiamati tutti i blocchi funzione (eccetto MC_Reset), ma non saranno eseguiti. MC_Reset genera una transizione allo stato StandStill. Se si verifica un errore mentre la macchina è nello stato Stopping, si verifica una transizione allo stato ErrorStop.
2	Power.Enable = TRUE ed è presente un errore nell'asse.
3	Power.Enable = TRUE e non sono presenti errori nell'asse.
4	MC_Stop.Done AND NOT MC_Stop.Execute.
5	Quando viene chiamato il blocco funzione MC_Power con Enable = False, lo stato dell'asse passa a Disabled per ogni stato, incluso ErrorStop.
6	Se si verifica un errore mentre la macchina è nello stato Stopping, si verifica una transizione allo stato ErrorStop.

Valori dei codici di stato dell'asse di controllo movimento

È possibile monitorare lo stato dell'asse tramite la funzione Monitoraggio asse. Nella seguente tabella sono identificati i valori utilizzati per definire ciascuno degli stati dell'asse predefiniti.

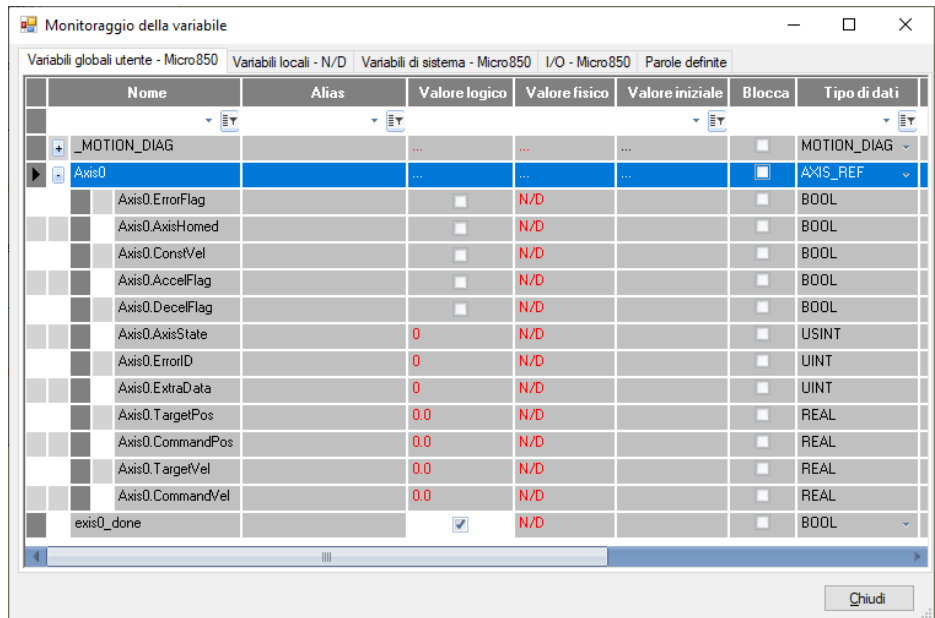
Valore stato	Nome stato
0x00	Disattivato
0x01	Fermo
0x02	Movimento discreto

0x03	Movimento continuo
0x04	Azzeramento
0x06	Arresto in corso
0x07	Arresto errore

Aggiornamenti di stato degli assi

All'esecuzione del movimento, l'aggiornamento di stato dell'asse dipende da quando il blocco funzione del movimento in questione viene chiamato dalla scansione POU. Questo vale anche quando il profilo di movimento è controllato dal Motore movimento come attività di background, indipendentemente dalla scansione POU.

Ad esempio, su un asse in movimento di una scala POU (stato di un piolo = true) viene analizzato un blocco funzione MC_MoveRelative nel piolo e l'asse inizia a muoversi. Prima del completamento di MC_MoveRelative, lo stato del piolo passa a False e MC_MoveRelative non viene più scansionato. In questo caso lo stato dell'asse non può passare da Discrete Motion a StandStill, neanche dopo l'arresto completo dell'asse e la velocità a 0.



Vedere anche

[Istruzioni di controllo del movimento](#) a pagina 423

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

Numeri dei parametri del blocco funzione per il controllo del movimento

Durante la programmazione dei blocchi funzione i seguenti blocchi funzione utilizzano numeri di parametri specifici.

- MC_ReadParameter
- MC_ReadBoolParameter
- MC_WriteParameter
- MC_WriteBoolParameter

Identificazione del numero del parametro

I numeri del parametro tra 0 e 999 sono riservati ai parametri standard. Sono consentite anche aggiunte da parte di fornitori o utenti, tuttavia ciò potrebbe incidere sulla trasportabilità da una piattaforma all'altra. Se il numero del parametri è maggiore di 999, il parametro è specifico del fornitore.

Numero parametro	Nome parametro	Tipo di dati	R/W	Descrizione
1	Posizione comandata	REAL	R	Posizione comandata.
2	SWLimitPos	REAL	R/W	Posizione interruttore limite software positivo.
3	SWLimitNeg	REAL	R/W	Posizione interruttore limite software negativo.
4	EnableLimitPos	BOOL	R/W	Abilita interruttore limite software positivo.
5	EnableLimitNeg	BOOL	R/W	Abilita interruttore limite software negativo.
8	MaxVelocitySystem	REAL	R	Velocità massima consentita per l'asse nel sistema di movimento.
9	MaxVelocityAppl	REAL	R/W	Velocità massima consentita per l'asse nell'applicazione.
11	CommandedVelocity	REAL	R	Velocità comandata.
12	MaxAccelerationSystem	REAL	R	Accelerazione massima consentita per l'asse nel sistema di movimento.
13	MaxAccelerationAppl	REAL	R/W	Accelerazione massima consentita per l'asse nell'applicazione.
14	MaxDecelerationSystem	REAL	R	Decelerazione massima consentita per l'asse nel sistema di movimento.
15	MaxDecelerationAppl	REAL	R/W	Decelerazione massima consentita per l'asse nell'applicazione.
16	MaxJerk	REAL	R/W	Strappo massimo dell'asse consentito.
1001	TargetPosition	REAL	R	La posizione di destinazione finale per il blocco funzione del movimento corrente attivo
1002	TargetVelocity1	REAL	R	La velocità di destinazione finale per il blocco funzione del movimento corrente attivo
1005	Ciclo di lavoro	REAL	R/W	Il ciclo di carico per un impulso. Il valore valido è compreso tra 0 e 100, per indicare l'intervallo 0% - 100%. (la funzione PWM può essere realizzata regolando questo valore). Questo parametro può essere configurato solo utilizzando questo blocco funzione. Il valore predefinito è impostato su 50.0 dal controllore. Nota: Per Ciclo di carico, il valore verrà sovrascritto dall'impostazione predefinita 50.0 quando il controllore passa dalla modalità RUN alla modalità PRG e nuovamente alla modalità RUN, o quando il controllore viene spento e riacceso.
1006	PulsePerRevolution	REAL	R	L'impostazione Impulsi per giro immessa dall'utente nell'interfaccia grafica CCW.
1007	TravelPerRevolution	REAL	R	L'impostazione Corsa per giro immessa dall'utente nell'interfaccia grafica CCW.

Vedere anche

[Istruzioni di controllo del movimento](#) a pagina 423

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

ID errori del blocco funzione di controllo movimento

Quando un blocco funzione Controllo movimento termina con un errore e lo stato dell'asse è ErrorStop, utilizzare il blocco funzione MC_Reset o MC_Power Off/On e MC_Reset per recuperare l'asse. L'asse può essere reimpostato sul movimento normale senza interrompere il funzionamento del controllore.

Utilizzare questa tabella per determinare gli errori per i blocchi funzione Controllo movimento.

Valore	ID MACRO	Descrizione
00	MC_FB_ERR_NO	Blocco funzione eseguito correttamente.
01	MC_FB_ERR_WRONG_STATE	Non è possibile eseguire il blocco funzione, in quanto l'asse non è nello stato corretto. Verificare lo stato dell'asse.
02	MC_FB_ERR_RANGE	Non è possibile eseguire il blocco funzione per via di parametri dinamici dell'asse non validi (velocità, accelerazione, decelerazione o strappo) impostati in un blocco funzione. Correggere l'impostazione dei parametri dinamici nel blocco funzione sulla pagina di configurazione Dinamica asse.
03	MC_FB_ERR_PARAM	Non è possibile eseguire il blocco funzione per via di un parametro non valido che non riguarda velocità, accelerazione, decelerazione o strappo, impostato in un blocco funzione. Correggere l'impostazione dei parametri (ad es. modalità o posizione) per il blocco funzione.
04	MC_FB_ERR_AXISNUM	Non è possibile eseguire il blocco funzione perché l'asse non esiste, i dati di configurazione dell'asse sono danneggiati oppure la configurazione dell'asse non è corretta.
05	MC_FB_ERR_MECHAN	Non è possibile eseguire il blocco funzione in quanto l'asse va in errore a causa di problemi meccanici o dell'unità. Controllare il collegamento tra unità e controllore (segnali Pronto unità e In posizione), quindi verificare che l'unità funzioni normalmente.
06	MC_FB_ERR_NOPOWER	Non è possibile eseguire il blocco funzione, in quanto l'asse non è alimentato. Alimentare l'asse con il blocco funzione MC_Power.
07	MC_FB_ERR_RESOURCE	Non è possibile eseguire il blocco funzione in quanto la risorsa richiesta dal blocco funzione è controllata da un altro blocco funzione o non è disponibile. Verificare che la risorsa richiesta dal blocco funzione sia disponibile per l'uso. Esempi: <ul style="list-style-type: none"> • MC_Power tenta di controllare lo stesso asse. • Sono eseguiti MC_Stop sullo stesso asse e nello stesso momento. • Sono eseguiti MC_TouchProbe sullo stesso asse e nello stesso momento. • Viene eseguito MC_TouchProbe quando l'ingresso della sonda a contatto non è abilitato in Configurazione movimento.
08	MC_FB_ERR_PROFILE	Non è possibile eseguire il blocco funzione, in quanto non è possibile ottenere il profilo di movimento definito nel blocco funzione. Correggere il profilo nel blocco funzione.

Valore	ID MACRO	Descrizione
09	MC_FB_ERR_VELOCITY	Non è possibile eseguire il blocco funzione, in quanto non è possibile ottenere il profilo di movimento richiesto nel blocco funzione a causa della velocità attuale dell'asse. Esempi: <ul style="list-style-type: none"> • Il blocco funzione richiede che l'asse inverta la direzione mentre l'asse è in movimento. • Il profilo di movimento richiesto non è raggiungibile per via della velocità corrente, troppo ridotta o troppo elevata. Verificare l'impostazione del profilo di movimento nel blocco funzione e correggerlo oppure rieseguire il blocco funzione quando la velocità dell'asse è compatibile con il profilo di movimento richiesto.
0A	MC_FB_ERR_SOFT_LIMIT	Non è possibile eseguire questo blocco funzione in quanto il movimento supererà il limite software, oppure il blocco funzione è stato interrotto a causa del raggiungimento del limite software. Controllare le impostazioni di velocità o posizione destinazione nel blocco funzione, oppure regolare il limite software.
0B	MC_FB_ERR_HARD_LIMIT	Il blocco funzione è stato interrotto in quanto è stato rilevato lo stato attivo dell'interruttore del limite hardware durante il movimento dell'asse o prima dell'avvio del movimento dell'asse. Spostare l'asse dall'interruttore del limite hardware, nella direzione opposta.
0C	MC_FB_ERR_LOG_LIMIT	Non è possibile eseguire questo blocco funzione in quanto il movimento supererà il limite logico dell'accumulatore PTO, oppure il blocco funzione è stato interrotto a causa del raggiungimento del limite logico dell'accumulatore PTO. Controllare le impostazioni di velocità o posizione destinazione per il blocco funzione. In alternativa, utilizzare il blocco funzione MC_SetPosition per regolare il sistema di coordinate dell'asse.
0D	MC_FB_ERR_ERR_ENGINE	È stato rilevato un errore di esecuzione del motore di movimento durante l'esecuzione di questo blocco funzione. Esegui un ciclo di alimentazione dell'intero sistema di movimento, inclusi controllori, unità e attuatori e scaricare nuovamente l'applicazione utente. Se l'errore persiste, rivolgersi all'assistenza tecnica.
10	MC_FB_ERR_NOT_HOMED	Non è possibile eseguire il blocco funzione, perché è necessario porre prima l'asse in sede. Riposizionare nuovamente l'asse in sede usando il blocco funzione MC.Home.
80	MC_FB_ERR_PARAM_MODIFIED	Avvertenza: la velocità richiesta per l'asse è stata regolata su un valore inferiore. Il blocco funzione è stato eseguito correttamente a una velocità ridotta.

Vedere anche

[Istruzioni di controllo del movimento](#) a pagina 423

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Stati asse di controllo movimento](#) a pagina 426

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

Scenari di errore asse

Nella maggior parte dei casi, quando l'istruzione di un blocco funzione movimento trasmessa a un asse comporta un errore del blocco funzione, l'asse viene contrassegnato come in stato Error e il corrispondente elemento ErrorID è impostato sui dati AXIS_REF per l'asse.

Nelle seguenti situazioni, l'asse non passa sempre allo stato di Errore ed è possibile per l'applicazione utente trasmettere un corretto blocco funzione movimento all'asse dopo che lo stato dell'asse cambia.

Scenario	Esempio
Un blocco funzione movimento comanda un asse, ma l'asse si trova in uno stato in cui il blocco funzione non può essere eseguito correttamente.	L'asse non ha potenza o l'asse si trova in una sequenza Homing oppure in uno stato Error Stop.
Un blocco funzione movimento comanda un asse, ma l'asse viene ancora controllato da un altro blocco funzione movimento. L'asse non può consentire che il movimento venga controllato dal nuovo blocco funzione senza raggiungere l'arresto completo.	Il nuovo blocco funzione specifica all'asse di cambiare la direzione di movimento.
Quando un blocco funzione prova a controllare un asse, ma l'asse è ancora controllato da un altro blocco funzione movimento e il nuovo profilo movimento definito non può essere realizzato da un controllore.	L'applicazione utente rilascia un blocco funzione curvato a S MC_MoveAbsolute su un asse, per diminuire la distanza fornita quando l'asse è in movimento.

Vedere anche

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

Tipo di dati AXIS_REF

Il tipo di dati AXIS_REF è una struttura dati che contiene informazioni per il controllo assi e viene utilizzata come variabile di ingresso e uscita in tutti i blocchi funzione di controllo movimento. Un'istanza di un tipo di dati AXIS_REF viene creata automaticamente con l'aggiunta di un asse di movimento alla configurazione.

Parametri	Tipo di dati	Descrizione
Axis_ID	AXIS_REF	L'ID asse logico è assegnato automaticamente da Connected Components Workbench. Non è possibile né modificarlo né visualizzarlo.
Flag di errore	BOOL	Indica se è presente un errore nell'asse. Se un asse è contrassegnato da un errore e l'ID errore non è zero, è necessario ripristinare l'asse tramite MC_Reset prima di attivare un altro blocco funzione di movimento.
AxisHomed	BOOL	Indica se l'operazione di autoguida per l'asse è stata eseguita correttamente. Quando l'utente tenta di rieseguire l'autoguida per un asse per cui è già impostato AxisHomed (autoguida eseguita correttamente), e l'esito è negativo, lo stato AxisHomed viene cancellato.
ConstVel	BOOL	Indica se il movimento dell'asse è in velocità costante. Un asse fermo non è considerato in velocità costante.
AccFlag	BOOL	Indica se il movimento dell'asse è in accelerazione.
DecelFlag	BOOL	Indica se il movimento dell'asse è in decelerazione.
AxisState	USINT	Indica lo stato attuale dell'asse.
ErrorID	UINT	Specifica la causa di un errore dell'asse quando ErrorFlag indica un errore. L'errore in genere deriva da un problema nell'esecuzione del blocco funzione di controllo movimento.
ExtraData	UINT	Riservato.
TargetPos	REAL	Indica la posizione di destinazione finale dell'asse per i blocchi funzione MoveAbsolute e MoveRelative. Per i blocchi funzione MoveVelocity, Stop e Halt, TargetPos è uguale a 0 tranne quando il valore TargetPos impostato da precedenti blocchi funzione di posizione non è stato eliminato.
CommandPos	REAL	Durante il movimento, rappresenta la posizione attuale che il controllore comanda all'asse di assumere. Potrebbe verificarsi un leggero ritardo tra la posizione effettiva dell'asse e questo CommandPos.
TargetVel	REAL	La velocità di destinazione massima impartita all'asse per un blocco funzione di movimento. Il valore di TargetVel nel blocco funzione corrente o un valore inferiore, a seconda degli altri parametri per lo stesso blocco funzione.

CommandVel	REAL	Durante il movimento, questo elemento indica la velocità corrente che il controllore comanda all'asse di utilizzare. Si tenga presente che potrebbe sussistere una leggera differenza tra la velocità effettiva dell'asse e CommandVel, a causa del ritardo dell'unità o della sua sovraelongazione di assestamento.
------------	------	---

Vedere anche

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[Variabili dell'asse](#) a pagina 434

Tipo di dati FB_AXIS_REF

Il tipo di dati FB_AXIS_REF è una struttura dati contenente informazioni per un asse di retroazione di movimento. Viene utilizzato come variabile di input e output nei blocchi funzione di controllo movimento. Un'istanza di un tipo di dati FB_AXIS_REF viene creata automaticamente con l'aggiunta di un modulo HSC e la modalità viene configurata come modalità Asse di retroazione.

IMPORTANTE se un asse di retroazione è contrassegnato da un errore e l'ID errore non è zero, è necessario ripristinare l'FBAxis tramite MC_Reset, prima di attivare un altro blocco funzione movimento.

Parametro	Tipo di dati	Descrizione
Axis_ID	FB_AXIS_REF	L'ID asse logico è assegnato automaticamente da Connected Components Workbench. Non è possibile né modificarlo né visualizzarlo.
ErrorFlag	BOOL	Indica se è presente un errore nell'asse di retroazione.
ConstVel	BOOL	Indica se il movimento dell'asse di retroazione è a velocità costante o meno. Un asse fermo non è considerato in velocità costante.
AccelFlag	BOOL	Indica se il movimento dell'asse di retroazione è in accelerazione o meno.
DecelFlag	BOOL	Indica se il movimento dell'asse di retroazione è in decelerazione o meno.
AxisState	USINT	Indica lo stato attuale dell'asse di retroazione.
ErrorID	UINT	Specifica la causa di un errore dell'asse quando ErrorFlag indica un errore. L'errore in genere deriva da un problema nell'esecuzione del blocco funzione di controllo movimento.
ExtraData	UINT	Riservato.
ActualPos	REAL	Lettura della posizione meccanica effettiva dal canale di retroazione movimento (HSC).
ActualVel	REAL	Lettura della velocità meccanica effettiva dal canale di retroazione movimento (HSC).

Vedere anche

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[Variabili dell'asse](#) a pagina 434

Variabili dell'asse

Le variabili dell'asse servono per controllare posizione, velocità, accelerazione ed errori per uno specifico asse di controllo movimento.

Assegnare una variabile a un parametro di output Axis

- In un diagramma a blocchi funzionali
Per comodità, connettere graficamente il parametro di output Axis di un blocco funzione per il controllo movimento al parametro di input AxisIn di un altro blocco funzione per il controllo movimento. Ad esempio, connettere il parametro di uscita MC_POWER Axis al parametro di ingresso MC_HOME AxisIn.
- In un ladder diagram
Non è possibile assegnare una variabile al parametro di output Axis di un blocco funzione per il controllo movimento in quanto è di sola lettura.

Monitorare una variabile AXIS_REF

Monitorare una variabile AXIS_REF o FBAXIS_REF nel software durante una connessione al controllore, quando il motore di movimento è attivo o nell'applicazione utente nell'ambito della logica utente. È possibile monitorare la variabile AXIS_REF o FBAXIS_REF anche da remoto tramite i diversi canali di comunicazione.

Vedere anche

[Istruzioni di controllo del movimento](#) a pagina 423

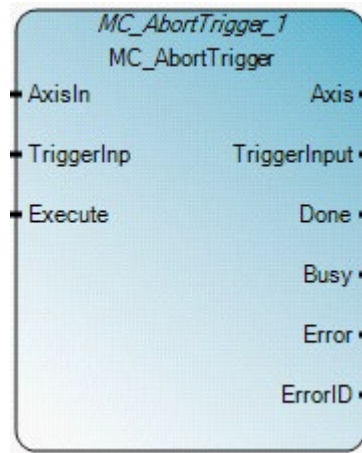
MC_AbortTrigger (trigger di interruzione del controllo movimento)

Interrompe i blocchi funzione Controllo assi collegati agli eventi di attivazione.

MC_AbortTrigger viene eseguito solo quando assegnato a un asse che viene controllato da MC_TouchProbe.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

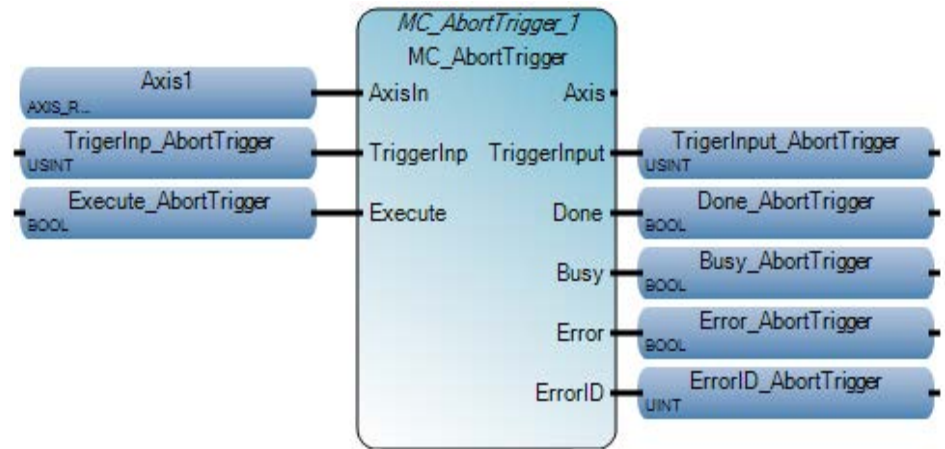
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



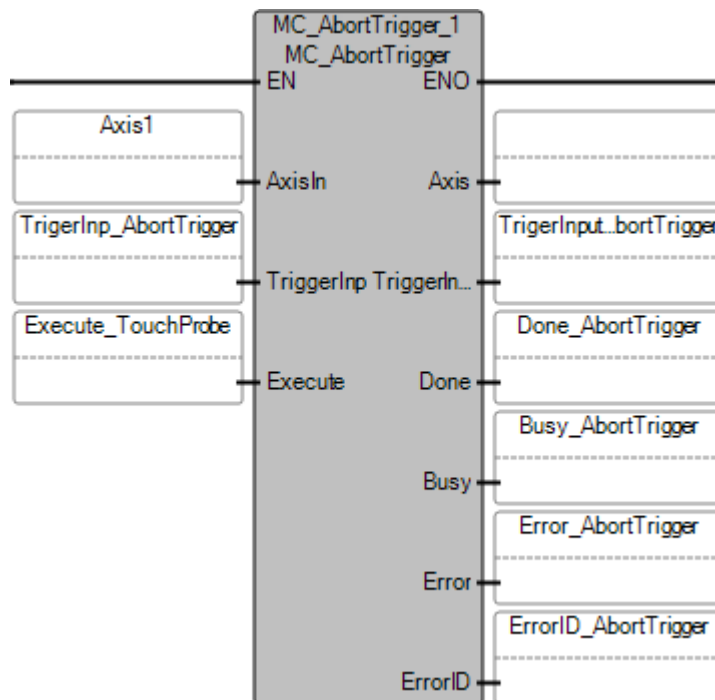
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Parametri	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_AbortTrigger corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare il tipo di dati AXIS_REF per definire il parametro AxisIn. Se l'asse è un FB_Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire i parametri di AxisIn.
TriggerInp	Ingresso	USINT	Questo parametro è ignorato.
Execute	Ingresso	BOOL	Quando TRUE, interrompe l'evento di trigger durante il fronte di salita.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
TriggerInput	Uscita	USINT	Questo parametro è ignorato.
Finito	Uscita	BOOL	TRUE: quando l'evento di trigger viene interrotto.
Busy	Uscita	BOOL	TRUE il blocco funzione non è finito.
Errore	Uscita	BOOL	TRUE: è stato rilevato un errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempi di programmazione a blocchi funzionali MC_AbortTrigger



Esempio di diagramma ladder MC_AbortTrigger



Esempio di testo strutturato MC_AbortTrigger

```

MC_AbortTrigger_1(
void MC_AbortTrigger_1([AXIS_REF|FB_AXIS_REF] AxisIn, USINT TriggerInp, BOOL Execute)
Tipo: MC_AbortTrigger, Interrompere i blocchi funzione collegati a eventi di attivazione (ad esempio MC_TouchProbe).

MC_AbortTrigger_1(Axis1, TrigerInp_AbortTrigger, Execute_AbortTrigger);
Done_AbortTrigger := MC_AbortTrigger_1.Done;
Busy_AbortTrigger := MC_AbortTrigger_1.Busy;
Error_AbortTrigger := MC_AbortTrigger_1.Error;
ErrorID_AbortTrigger := MC_AbortTrigger_1.ErrorID;
    
```


Risultati

Nome	Valore logico	Valore fisico	Blocca	Tipo di dati
TriggerInput_AbortTrigger	0	N/D	<input type="checkbox"/>	USINT
Execute_AbortTrigger	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
TriggerInp_AbortTrigger	0	N/D	<input type="checkbox"/>	USINT
Done_AbortTrigger	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
Busy_AbortTrigger	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
Error_AbortTrigger	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
ErrorID_AbortTrigger	0	N/D	<input type="checkbox"/>	UINT

Vedere anche

[Tipo di dati AXIS_REF](#) a pagina 432

[Tipo di dati FB_AXIS_REF](#) a pagina 433

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Istruzioni di controllo del movimento](#) a pagina 423

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

MC_Halt (halt di controllo movimento)

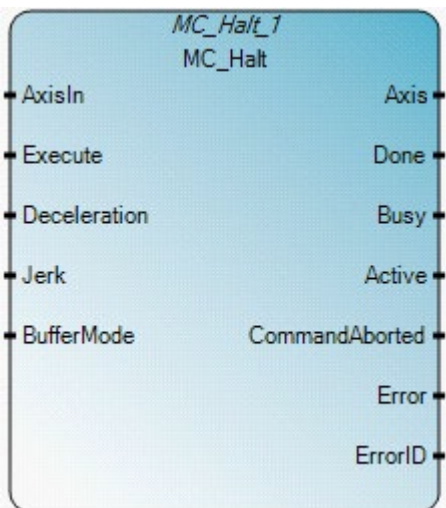
Comanda un arresto controllato del movimento in condizioni operative normali.

Dettagli operazioni:

- Lo stato dell'asse cambia a DiscreteMotion, fino a quando la velocità non è zero. Quando la velocità raggiunge lo zero, Done viene impostato su True e lo stato dell'asse cambia a StandStill.
- Durante la decelerazione dell'asse è possibile eseguire un altro comando di controllo assi che interrompe MC_Halt.
- Se MC_Halt viene rilasciato quando lo stato dell'asse è Homing, il blocco funzione riporta un errore e il processo di homing non viene interrotto.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

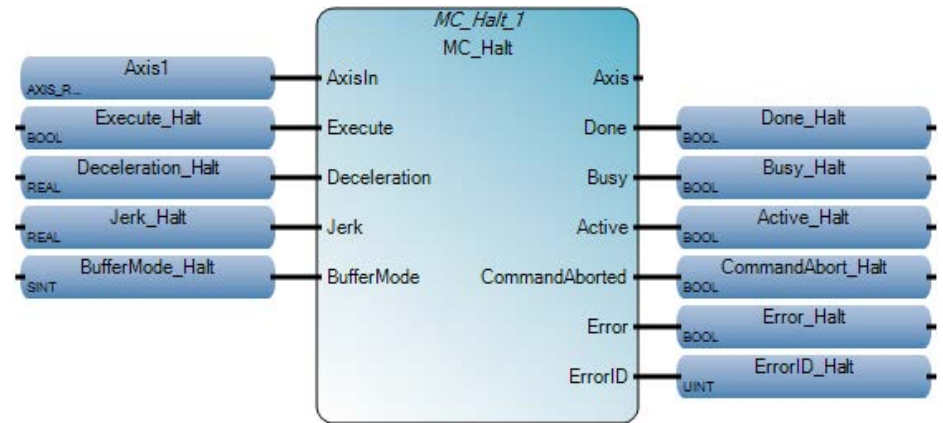
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



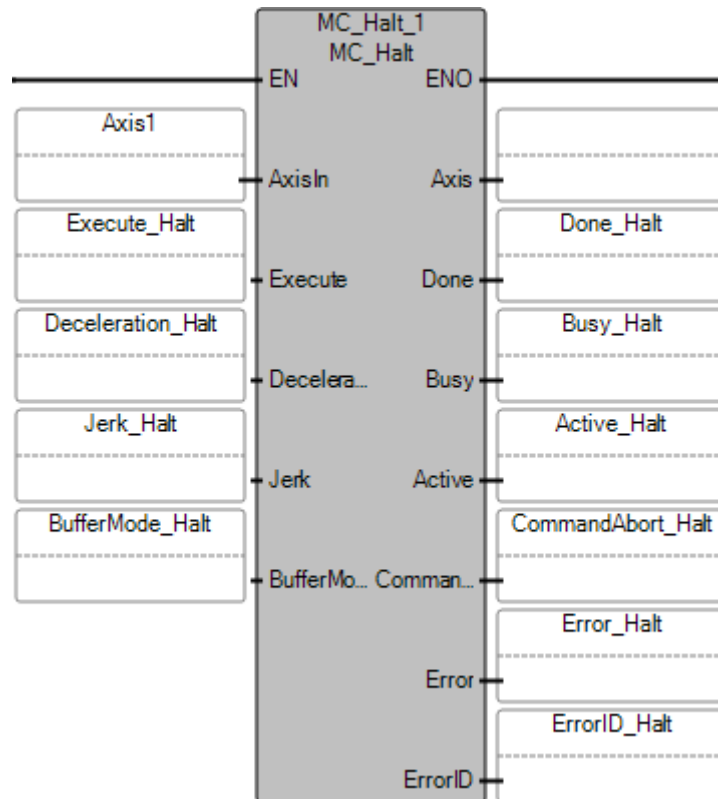
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo di MC_Halt corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF	Utilizzare i parametri del tipo di dati AXIS_REF per definire AxisIn.
Execute	Ingresso	BOOL	Indica quando avviare il movimento. TRUE: avviare il movimento durante il fronte di salita. FALSE: non avviare il movimento. l'esecuzione di MC_Halt durante il ritorno in posizione iniziale, MC_Halt è impostato su MC_FB_ERR_STATE e il processo di ritorno in posizione iniziale continua.
Deceleration	Ingresso	REAL	Valore della decelerazione (sempre positivo) (energia in riduzione del motore). se Deceleration <= 0 e lo stato dell'asse non è StandStill, MC_Halt è impostato su MC_FB_ERR_RANGE.
Jerk	Ingresso	REAL	Valore Jerk (sempre positivo). se Deceleration <= 0 e lo stato dell'asse è StandStill, MC_Halt è impostato su MC_FB_ERR_RANGE.
BufferMode	Ingresso	SINT	Non utilizzato. La modalità è sempre MC_Aborting.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Finito	Uscita	BOOL	Raggiungimento della velocità zero.
Busy	Uscita	BOOL	Il blocco istruzione non è finito.
Attivo	Uscita	BOOL	Indica che il blocco istruzione ha il controllo sull'asse.
CommandAborted	Uscita	BOOL	Il comando è interrotto da un altro comando o arresto di errore.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempio di programmazione a blocchi funzionali MC_Halt



Esempio di diagramma ladder MC_Halt

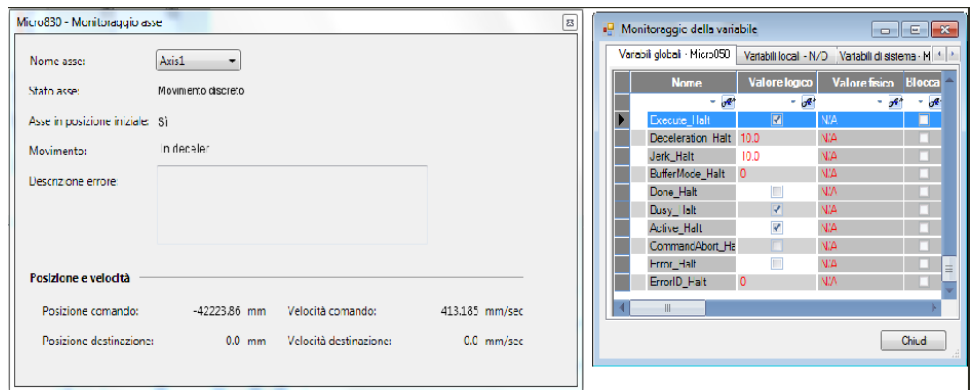


Esempio di testo strutturato MC_Halt

```
MC_Halt_1(
void MC_Halt_1(AXIS_REF AxisIn, BOOL Execute, REAL Deceleration, REAL Jerk, SINT BufferMode)
Tipo: MC_Halt, Determina un'interruzione del movimento controllata. Lo stato dell'asse viene modificato in Movimento discreto, fino a quando raggiunge la velocità zero.

Deceleration_Halt := 10.0;
Jerk_Halt := 10.0;
MC_Halt_1(Axis1, Execute_Halt, Deceleration_Halt, Jerk_Halt, BufferMode_Halt);
Done_Halt := MC_Halt_1.Done;
Busy_Halt := MC_Halt_1.Busy;
Active_Halt := MC_Halt_1.Active;
CommandAbort_Halt := MC_Halt_1.CommandAborted;
Error_Halt := MC_Halt_1.Error;
ErrorID_Halt := MC_Halt_1.ErrorID;
```

Risultati



Vedere anche

- [Tipo di dati AXIS_REF](#) a pagina 432
- [Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424
- [Istruzioni di controllo del movimento](#) a pagina 423
- [ID errori del blocco funzione di controllo movimento](#) a pagina 430
- [Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

MC_Home (home del controllo movimento)

Comanda l'asse per l'esecuzione della sequenza <search home>. I dettagli della sequenza dipendono dal produttore e possono essere impostati mediante i parametri degli assi. L'ingresso "Position" è usato per impostare la posizione assoluta quando viene rilevato un segnale di riferimento e raggiunto l'offset Azzeramento configurato.

Dettagli operazione:

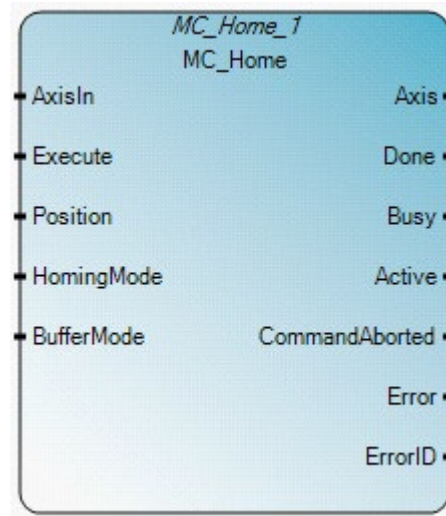
- Dopo l'emissione di MC_Power, lo stato dell'asse Azzerato viene ripristinato a 0 (non azzerato). Nella maggior parte dei casi, dopo

che l'asse viene alimentato, il blocco funzione MC_Home deve essere eseguito per calibrare la posizione dell'asse e il riferimento di posizione iniziale.

- Il blocco funzione MC_Home può essere interrotto solo usando i blocchi funzione MC_Stop o MC_Power. Se interrotto prima di essere completato, la posizione iniziale precedentemente cercata verrà considerata non valida e lo stato di posizione iniziale dell'asse sarà cancellato.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo corrente MC_Home. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF	Utilizzare i parametri del tipo di dati AXIS_REF per definire AxisIn.
Execute	Ingresso	BOOL	Indica quando avviare il movimento. TRUE: avviare il movimento durante il fronte di salita. FALSE: non avviare il movimento.
Position	Ingresso	REAL	La posizione assoluta viene impostata quando viene rilevato il segnale di riferimento e raggiunto l'offset della posizione iniziale configurata. L'intervallo di valori per questo ingresso è -0x40000000 - 0x40000000 impulsi fisici, dopo che la posizione è convertita dall'unità di posizione utente all'impulso PTO. Impostare il valore del parametro Position entro Soft Limit. Un valore di ingresso non valido ha generato un errore. Error ID = MC_FB_ERR_PARAM .
HomingMode	Ingresso	SINT	Ingresso numerale per la modalità Homing.
BufferMode	Ingresso	SINT	Non utilizzato. La modalità è sempre mcAborting.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.

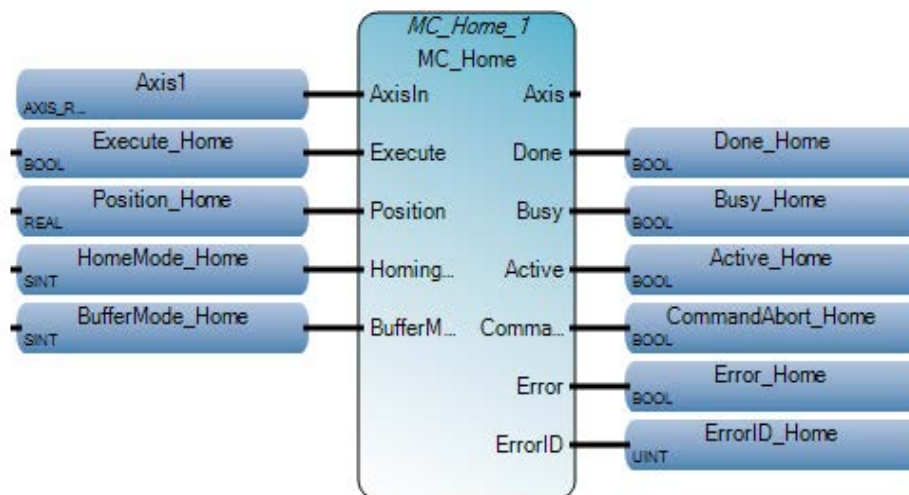
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Finito	Uscita	BOOL	TRUE: l'operazione di Homing è terminata con successo e lo stato dell'asse è impostato su Fermo. FALSE: l'operazione di Homing è in corso o è incompleta.
Busy	Uscita	BOOL	TRUE il blocco istruzione non è finito. FALSE: il blocco istruzione è completato.
Attivo	Uscita	BOOL	TRUE: indica che il blocco istruzione ha il controllo sull'asse.
CommandAborted	Uscita	BOOL	TRUE: il comando è stato interrotto da un altro comando, o da un arresto su errore.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UNIT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

HomingModes

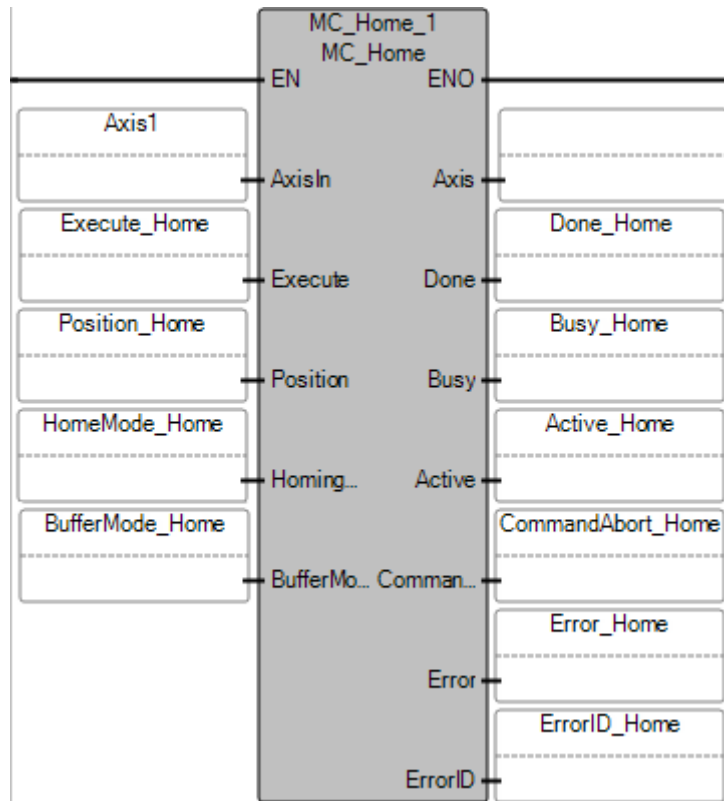
Utilizzare questa tabella per determinare i valori per il parametro HomingMode nell'istruzione di controllo movimento MC_Home.

Valore	Nome	Descrizione
0x00	MC_HOME_ABS_SWITCH	Processo Homing mediante ricerca dell'interruttore Home Absolute
0x01	MC_HOME_LIMIT_SWITCH	Processo Homing mediante ricerca dell'interruttore limite
0x02	MC_HOME_REF_WITH_ABS	Processo Homing mediante ricerca dell'interruttore Home Absolute e uso dell'impulso di riferimento encoder
0x03	MC_HOME_REF_PULSE	Processo Homing mediante ricerca dell'interruttore limite e uso dell'impulso di riferimento encoder
0x04	MC_HOME_DIRECT	Processo Homing statico con forzatura diretta di una posizione iniziale da riferimento utente. Il blocco funzione imposterà la posizione corrente in cui si trova il meccanismo come posizione iniziale, con la sua posizione determinata dal parametro di ingresso "Position".

Esempi di programmazione a blocchi funzionali MC_Home



Esempio di diagramma ladder MC_Home



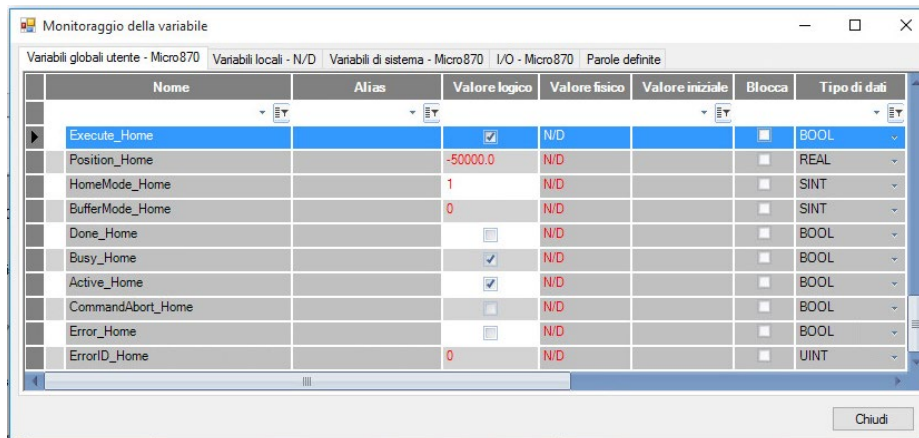
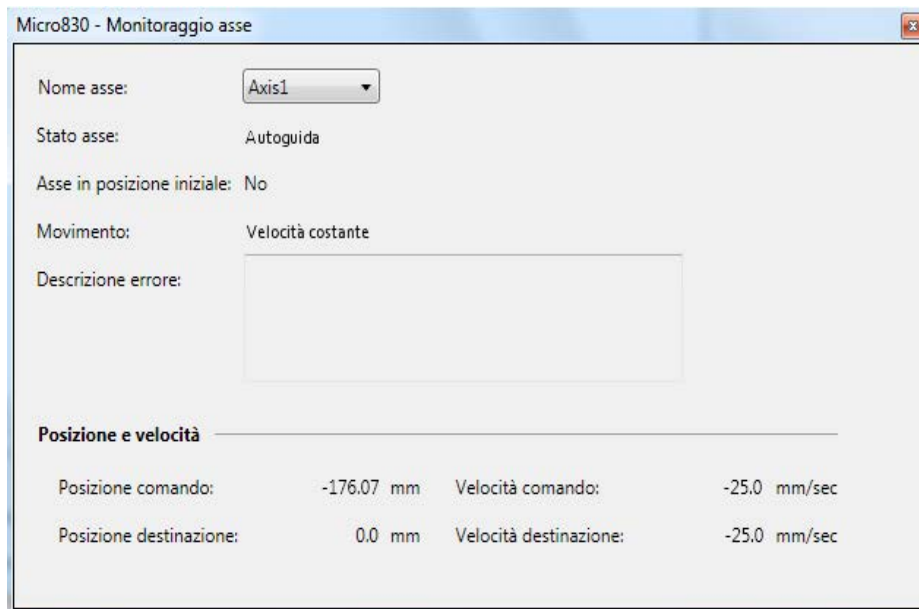
Esempio di testo strutturato MC_Home

```
MC_Home_1(
void MC_Home_1(AXIS_REF AxisIn, BOOL Execute, REAL Position, SINT HomingMode, SINT BufferMode)
Tipo: MC_Home, Determina l'esecuzione della sequenza di ricerca di Home da parte dell'asse.
```

```

Position_Home := -50000.0;
HomeMode_Home := 4; (*1*)
MC_Home_1(Axis1,Execute_Home,Position_Home,HomeMode_Home,BufferMode_Home);
Done_Home := MC_Home_1.Done;
Busy_Home := MC_Home_1.Busy;
Active_Home := MC_Home_1.Active;
CommandAbort_Home := MC_Home_1.CommandAborted;
Error_Home := MC_Home_1.Error;
ErrorID_Home := MC_Home_1.ErrorID;
    
```

Risultati



Vedere anche

[Tipo di dati AXIS_REF](#) a pagina 432

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Istruzioni di controllo del movimento](#) a pagina 423

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

MC_MoveAbsolute (movimento assoluto del controllo movimento)

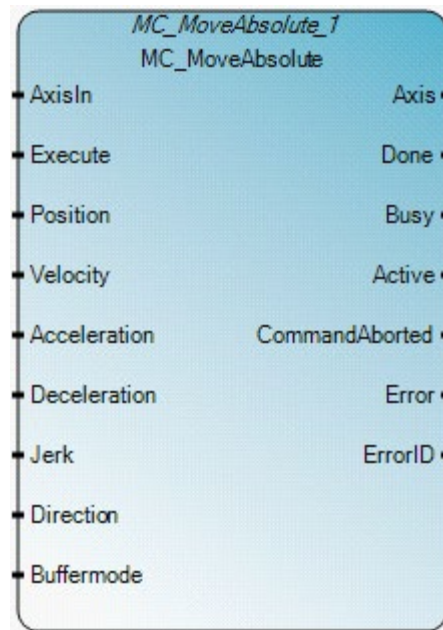
Determina un movimento controllato verso una posizione assoluta specificata.

Dettagli operazione:

- Per il controllore Micro800,
 - il segno dell'ingresso Velocità per un blocco funzione MC_MoveAbsolute viene ignorato, in quanto la direzione del movimento è determinata dalla posizione corrente e dalla posizione di destinazione.
 - L'ingresso Direzione per un blocco funzione MC_MoveAbsolute viene ignorato, in quanto esiste solo una soluzione matematica per raggiungere la posizione di destinazione.
- Se il blocco funzione MC_MoveAbsolute viene emesso quando lo stato dell'asse del controllore Micro800 è Fermo e la distanza relativa del movimento è zero, l'esecuzione del blocco funzione viene riportata immediatamente come Done.
- Se un blocco funzione MC_MoveAbsolute viene emesso per un asse che non si trova in posizione azzerata, il blocco funzione riporterà un errore.
- Il blocco funzione MoveAbsolute verrà completato con velocità zero se non viene interrotto da un altro blocco funzione.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

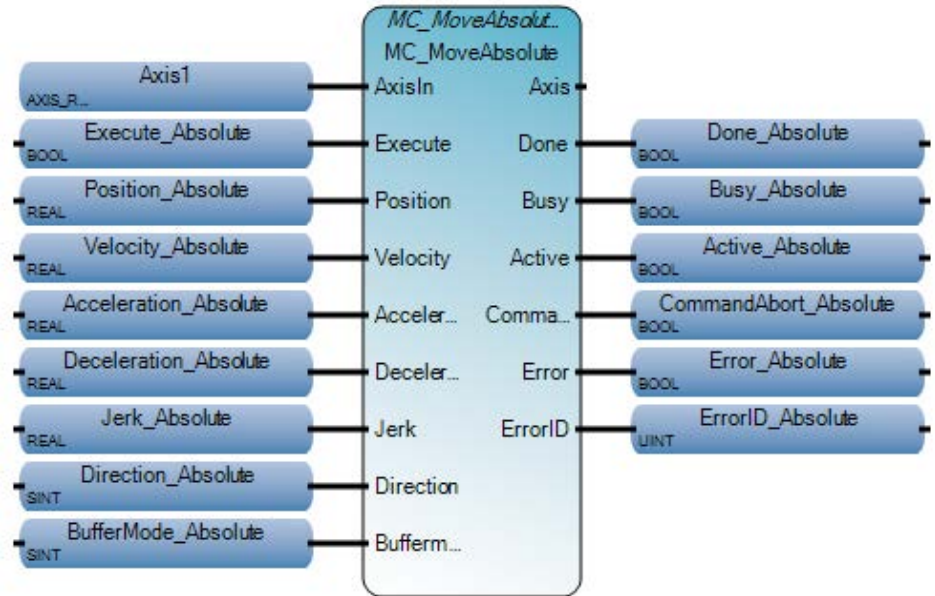
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo corrente MC_MoveAbsolute. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF	Utilizzare i parametri del tipo di dati AXIS_REF per definire AxisIn.
Execute	Ingresso	BOOL	Indica quando avviare il movimento. TRUE: avviare il movimento durante il fronte di salita. FALSE: non avviare il movimento. L'asse deve essere in posizione iniziale quando viene emesso il comando di esecuzione o quando si verifica un errore, MC_FB_ERR_NOT_HOMED.
Position	Ingresso	REAL	Posizione di destinazione per il movimento in unità tecnica (negativa o positiva). L'unità tecnica è definita nella pagina di configurazione Controllo assi - Generale per un asse.
Velocity	Ingresso	REAL	Valore della velocità massima. La velocità massima non può essere raggiunta se Jerk = 0. Il segno del parametro di velocità è ignorato, la direzione del moto è determinata dal valore di ingresso Position.
Acceleration	Ingresso	REAL	Valore dell'accelerazione (sempre positiva - aumento dell'energia al motore). unità utente/sec ²
Deceleration	Ingresso	REAL	Valore della decelerazione (sempre positiva - diminuzione dell'energia al motore). u/sec ²
Jerk	Ingresso	REAL	Valore Jerk (sempre positivo). u/sec ³ quando il valore dello strappo in ingresso = 0, il profilo del Trapezoide è calcolato da Motion Engine. Quando Jerk > 0, viene calcolato il profilo S-Curve.
Direzione	Ingresso	SINT	Questo parametro non è usato.
BufferMode	Ingresso	SINT	Questo parametro non è usato.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Finito	Uscita	BOOL	Se TRUE, raggiunta posizione di comando. Quando il parametro di ingresso In-Position per questo asse è configurato su Attivato, l'unità deve impostare il segnale di ingresso In-Position come attivo prima che il bit Done diventi True. Questa azione termina con velocità zero, a meno che non sia interrotta.
Busy	Uscita	BOOL	Se TRUE, il blocco funzione non è terminato.
Attivo	Uscita	BOOL	Se TRUE, indica che il blocco funzione ha il controllo dell'asse
CommandAborted	Uscita	BOOL	Se TRUE, il comando è stato interrotto da un altro comando, o da un arresto su errore.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempi di programmazione a blocchi funzionali MC_MoveAbsolute



Esempio di diagramma ladder MC_MoveAbsolute



Esempio di testo strutturato MC_MoveAbsolute

```

MC_MoveAbsolute_1(
void MC_MoveAbsolute_1(AXIS_REF AxisIn, BOOL Execute, REAL Position, REAL Velocity, REAL Acceleration, REAL Deceleration, REAL Jerk, SINT Direction, SINT BufferMode)
Tipo: MC_MoveAbsolute; Determina un movimento controllato verso una posizione assoluta specificata.

Position_Absolute := 50000.0;
Velocity_Absolute := 500.0;
Acceleration_Absolute := 1000.0;
Deceleration_Absolute := 1000.0;
Jerk_Absolute := 10.0;
MC_MoveAbsolute_1(Axis1, Execute_Absolute, Position_Absolute, Velocity_Absolute,
Acceleration_Absolute, Deceleration_Absolute, Jerk_Absolute, Direction_Absolute, BufferMode_Absolute);
Done_Absolute := MC_MoveAbsolute_1.Done;
Busy_Absolute := MC_MoveAbsolute_1.Busy;
Active_Absolute := MC_MoveAbsolute_1.Active;
CommandAbort_Absolute := MC_MoveAbsolute_1.CommandAborted;
Error_Absolute := MC_MoveAbsolute_1.Error;
ErrorID_Absolute := MC_MoveAbsolute_1.ErrorID;

```

Risultati

The left screenshot shows the 'Monitoraggio di asse' window for Axis 1. It displays the following information:

- Nome asse: Axis1
- Stato asse: Movimento iniziato
- Asse in posizione iniziale: Sì
- Movimento: velocità costante
- Descrizione errore:

The right screenshot shows the 'Monitoraggio della variabile' window, displaying a table of variable values for MC_MoveAbsolute_1. The table has the following columns: Nome, Valore logico, Valore fisico, Blocca, and Tipo di dati.

Nome	Valore logico	Valore fisico	Blocca	Tipo di dati
Execute_Absolute	1	N/A		BOOL
Position_Absolute	50000.0	N/A		REAL
Velocity_Absolute	500.0	N/A		REAL
Acceleration_Absolute	1000.0	N/A		REAL
Deceleration_Absolute	1000.0	N/A		REAL
Jerk_Absolute	10.0	N/A		REAL
Direction_Absolute	0	N/A		SINT
BufferMode_Absolute	0	N/A		SINT
Done_Absolute		N/A		BOOL
Busy_Absolute		N/A		BOOL
Active_Absolute		N/A		BOOL
CommandAbort_Absolute		N/A		BOOL
Error_Absolute		N/A		BOOL
ErrorID_Absolute		N/A		UINT

Vedere anche

[Istruzioni di controllo del movimento](#) a pagina 423

[Informazioni generali sulle istruzioni di controllo movimento](#) a pagina 426

[Tipo di dati AXIS_REF](#) a pagina 432

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

MC_MoveRelative (movimento relativo del controllo movimento)

Determina un movimento controllato di una distanza specificata rispetto alla posizione attuale nel momento dell'esecuzione.

Dettagli operazione:

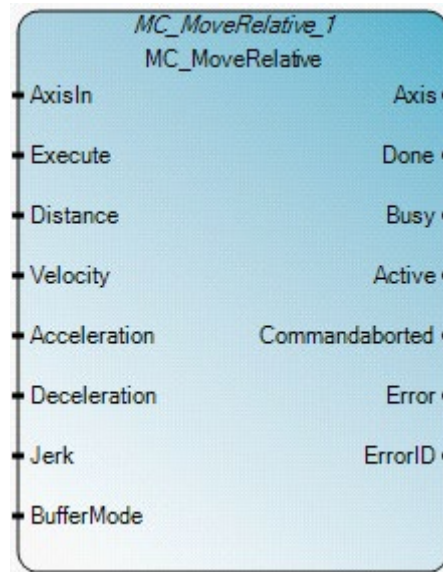
- Il segno della velocità viene ignorato, in quanto la direzione del movimento per MC_MoveRelative viene determinata dalla posizione corrente e dalla posizione di destinazione.
- MoveRelative viene completato con velocità zero, se non viene interrotto da un altro blocco funzione.
- Se viene emesso il comando MC_MoveRelative quando lo stato dell'asse del controllore Micro800 è Fermo e la distanza relativa

di movimento è zero, l'esecuzione del blocco funzione viene riportata immediatamente come Completata.

- Per un controllore Micro800, il segno dell'ingresso Velocità per MC_MoveRelative viene ignorato, in quanto la direzione del movimento viene determinata dalla posizione corrente e dalla posizione di destinazione.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



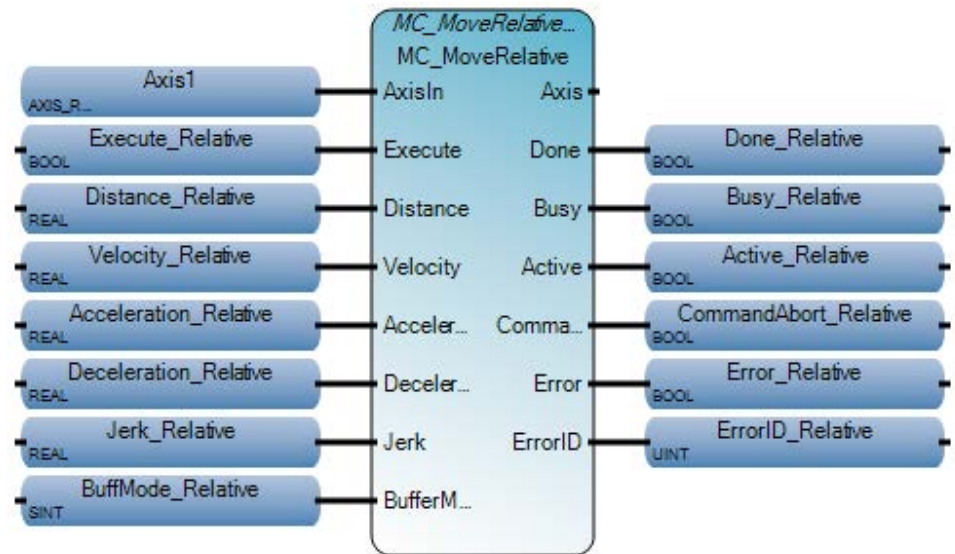
Operazione MC_MoveRelative

Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

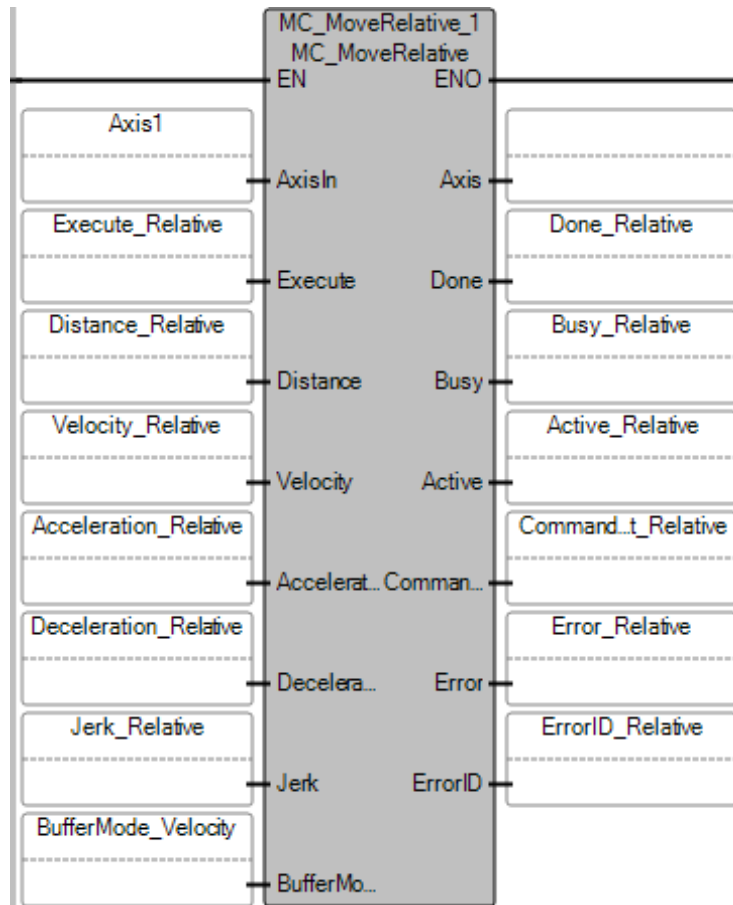
Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_MoveRelative corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF	Utilizzare i parametri del tipo di dati AXIS_REF per definire AxisIn.
Execute	Ingresso	BOOL	Indica quando avviare il movimento. TRUE: avviare il movimento durante il fronte di salita. FALSE: non avviare il movimento.
Distance	Ingresso	REAL	Distanza relativa del movimento (in unità tecnica [u]).
Velocity	Ingresso	REAL	Valore della velocità massima (non necessariamente raggiunta) [u/s]. Dato che la direzione del moto è determinata dal valore del parametro Position, il segno di Velocity è ignorato dal blocco funzione. La velocità massima non può essere raggiunta se Jerk = 0.
Acceleration	Ingresso	REAL	Valore dell'accelerazione (aumento dell'energia al motore) [u/s ²]
Deceleration	Ingresso	REAL	Valore della decelerazione (diminuzione dell'energia al motore) [u/s ²]
Jerk	Ingresso	REAL	Valore del jerk [u/s ³]

Parametri	Tipo di parametro	Tipo di dati	Descrizione
BufferMode	Ingresso	SINT	Questo parametro non è usato.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Finito	Uscita	BOOL	TRUE: raggiunta distanza comandata. Quando il parametro di ingresso In-Position per questo asse è abilitato, il segnale di ingresso In-Position deve essere impostato come attivo prima che Done = True.
Busy	Uscita	BOOL	TRUE il blocco istruzione non è finito. FALSE: il blocco istruzione è completato.
Attivo	Uscita	BOOL	TRUE: indica che il blocco istruzione ha il controllo sull'asse.
CommandAborted	Uscita	BOOL	TRUE: comando interrotto da un altro comando o da Arresto errore.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempi di programmazione a blocchi funzionali MC_MoveRelative



Esempio di diagramma ladder MC_MoveRelative



Esempio di testo strutturato MC_MoveRelative

```
MC_MoveRelative_1
void MC_MoveRelative_1(AXIS_REF AxisIn, BOOL Execute, REAL Distance, REAL Velocity, REAL Acceleration, REAL Deceleration, REAL Jerk, SINT BufferMode)
Tipo: MC_MoveRelative, Determina un movimento controllato di una distanza specificata rispetto alla posizione attuale nel momento dell'esecuzione.
```

```
Distance_Relative := 100000.0;
Velocity_Relative := 300.0;
Acceleration_Relative := 100.0;
Deceleration_Relative := 100.0;
Jerk_Relative := 100.0;
MC_MoveRelative_1(Axis1, Execute_Relative, Distance_Relative,
Velocity_Relative, Acceleration_Relative, Deceleration_Relative, Jerk_Relative, BuffMode_Relative);
Done_Relative := MC_MoveRelative_1.Done;
Busy_Relative := MC_MoveRelative_1.Busy;
Active_Relative := MC_MoveRelative_1.Active;
CommandAbort_Relative := MC_MoveRelative_1.Commandaborted;
Error_Relative := MC_MoveRelative_1.Error;
ErrorID_Relative := MC_MoveRelative_1.ErrorID;
```


Risultati

Micro850 - Monitoraggio asse

Nome asse:

Stato asse: Disattivato

Asse in posizione iniziale: No

Movimento: Velocità costante

Descrizione errore:

Posizione e velocità

Posizione comandata: 60875.68 mm Comando di velocità: 300.0 mm/sec

Posizione di destinazione: 100000.0 mm Velocità destinazione: 300.0 mm/sec

Monitoraggio della variabile

Variabili globali utente - Micro870 Variabili locali - N/D Variabili di sistema - Micro870 I/O - Micro870 Parole definite

	Nome	Valore logico	Valore fisico	Blocca	Tipo di dati
▶	Execute_Relative	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
	Distance_Relative	100000.0	N/D	<input type="checkbox"/>	REAL
	Velocity_Relative	300.0	N/D	<input type="checkbox"/>	REAL
	Acceleration_Relative	100.0	N/D	<input type="checkbox"/>	REAL
	Deceleration_Relative	100.0	N/D	<input type="checkbox"/>	REAL
	Jerk_Relative	100.0	N/D	<input type="checkbox"/>	REAL
	BufferMode_Relative	0	N/D	<input type="checkbox"/>	SINT
	Done_Relative	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
	Busy_Relative	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
	Active_Relative	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
	CommandAbort_Relative	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
	Error_Relative	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
	ErrorID_Relative	0	N/D	<input type="checkbox"/>	UINT

Chiudi

Vedere anche

[Tipo di dati AXIS_REF](#) a pagina 432

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Istruzioni di controllo del movimento](#) a pagina 423

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

MC_MoveVelocity (velocità di movimento del controllo movimento)

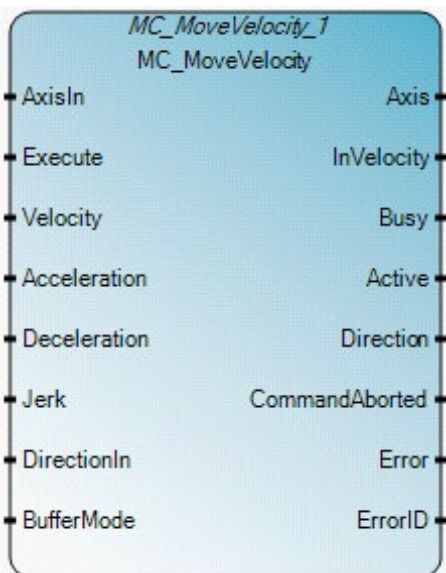
Determina un movimento controllato infinito a una velocità specificata.

Dettagli operazione:

- Se l'ingresso DirectionIn per MC_MoveVelocity è uguale a 0 e:
 - l'asse è in uno stato di movimento, il segno dell'ingresso Velocità viene ignorato, l'asse continua a muoversi nell'attuale direzione di movimento e vengono applicati i nuovi parametri dinamici.
 - L'asse non è in uno stato di movimento, MC_MoveVelocity riporta un errore.
- Se durante l'esecuzione di MC_MoveVelocity viene raggiunto il limite di Impulso PTO, il valore Accumulatore PTO viene riportato a 0 (o, se è attivato il limite, al limite software opposto) e l'esecuzione continua.
- Se l'asse è in stato di movimento e MC_MoveVelocity invia un movimento per cui la direzione (segno di velocità * direzione) è opposta all'attuale direzione di movimento, MC_MoveVelocity riporterà un errore.
- Una volta impostato, il segnale "InVelocity" indica che MC_MoveVelocity è stato completato. Qualsiasi evento di movimento successivo non avrà effetto sulle uscite MC_MoveVelocity, ad eccezione del segnale "InVelocity".
- L'uscita InVelocity di MC_MoveVelocity resta True dopo che la velocità dell'asse raggiunge la velocità comandata o finché il MC_MoveVelocity non viene interrotto.
- Il segno di (Velocità * Direzione) determina la direzione del movimento di MC_MoveVelocity. Se il segno di Velocità e il segno di Direzione sono uguali, viene emesso un movimento positivo. Se il segno di Velocità e Direzione non sono uguali, viene emesso un movimento negativo.
- Il segnale "InVelocity" viene reimpostato quando MC_MoveVelocity viene interrotto da un altro blocco funzione o evento di movimento o sul fronte di discesa di "Esegui".
- Per arrestare o modificare il controllo assi iniziato da MC_MoveVelocity, il blocco istruzione deve essere interrotto o arrestato da un altro blocco istruzione, inclusa la ripetizione dell'esecuzione di MC_MoveVelocity con parametri differenti.
- Se MC_MoveVelocity viene emesso con l'asse in stato Fermo (non controllato da un altro blocco funzione) e si verifica un errore nel blocco funzione, lo stato dell'asse passa a ErrorStop.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

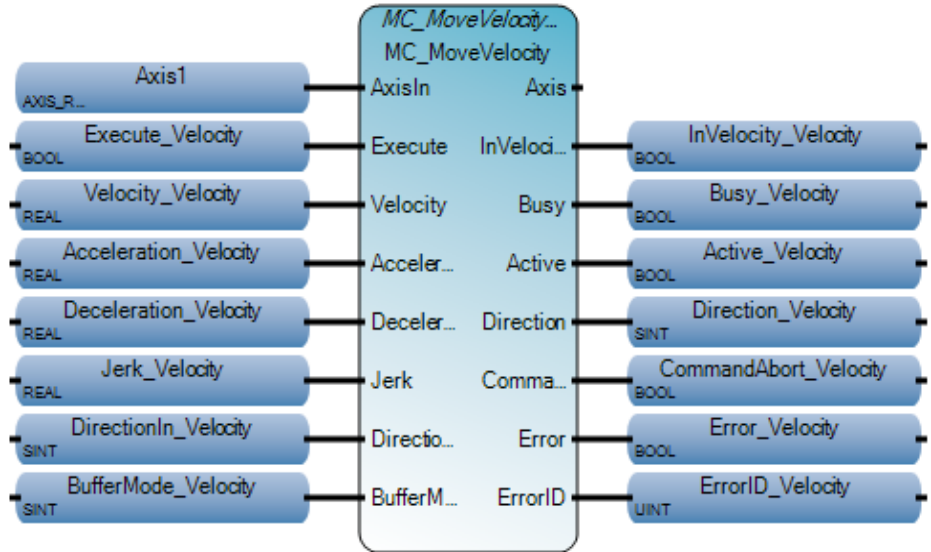
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



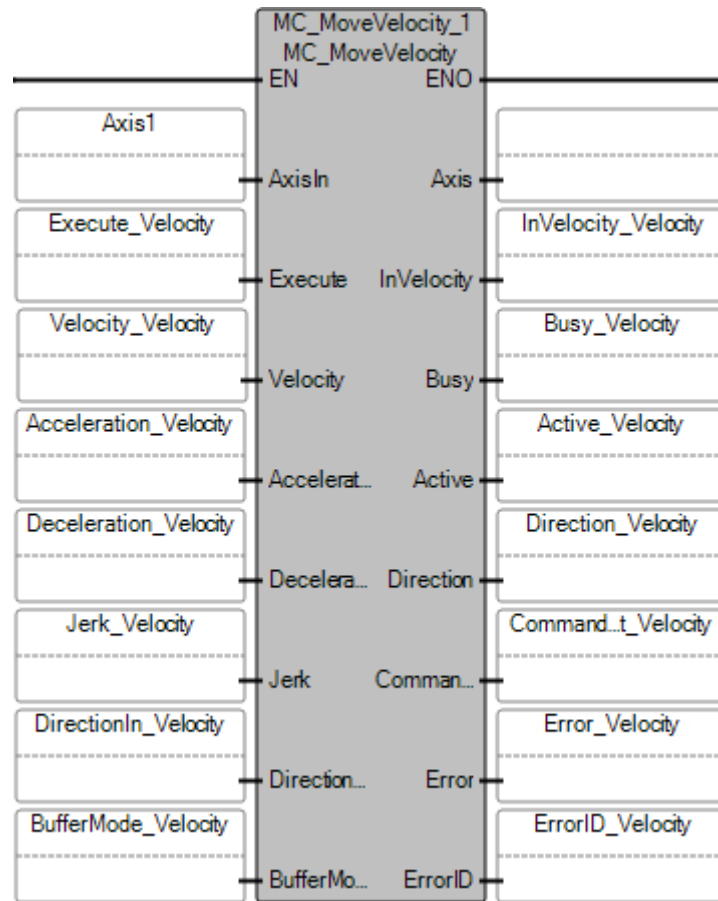
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_MoveVelocity corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF	Utilizzare i parametri del tipo di dati AXIS_REF per definire AxisIn.
Execute	Ingresso	BOOL	Indica quando avviare il movimento. TRUE: avviare il movimento durante il fronte di salita. FALSE: non avviare il movimento.
Velocity	Ingresso	REAL	Valore della velocità massima [u/s].
Acceleration	Ingresso	REAL	Valore dell'accelerazione (aumento dell'energia al motore) [u/s ²]
Deceleration	Ingresso	REAL	Valore della decelerazione (diminuzione dell'energia al motore) [u/s ²]
Jerk	Ingresso	REAL	Valore del jerk [u/s ³]
DirectionIn	Ingresso	SINT	I valori validi sono: -1, 0, 1.
BufferMode	Ingresso	SINT	Questo parametro non è usato.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
InVelocity	Uscita	BOOL	TRUE: è stata raggiunta la velocità comandata (prima volta).
Busy	Uscita	BOOL	TRUE il blocco istruzione non è finito. FALSE: il blocco istruzione è completato.
Attivo	Uscita	BOOL	TRUE: indica che il blocco funzione ha il controllo sull'asse.
Direzione	Uscita	SINT	I valori validi sono: -1, 0, 1.
CommandAborted	Uscita	BOOL	TRUE: comando interrotto da un altro comando o da Arresto errore.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempi di programmazione a blocchi funzionali MC_MoveVelocity



Esempio di diagramma ladder MC_MoveVelocity



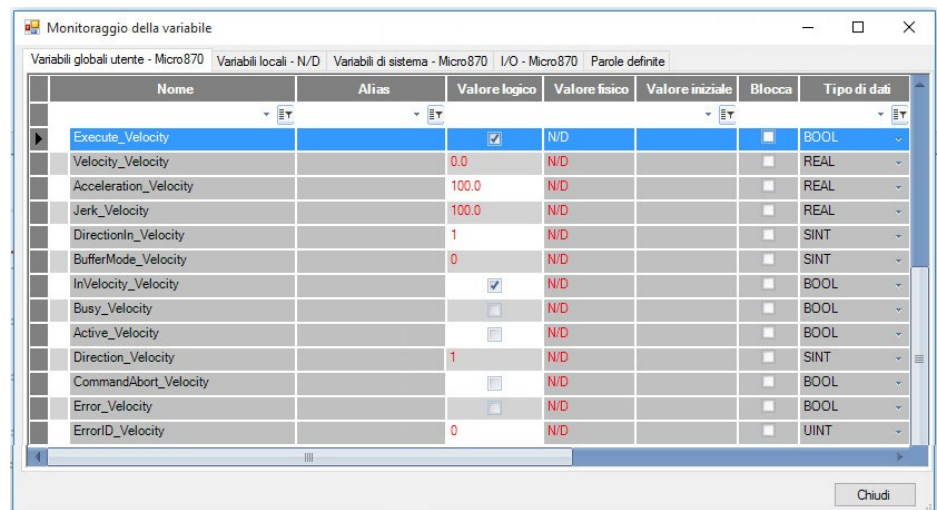
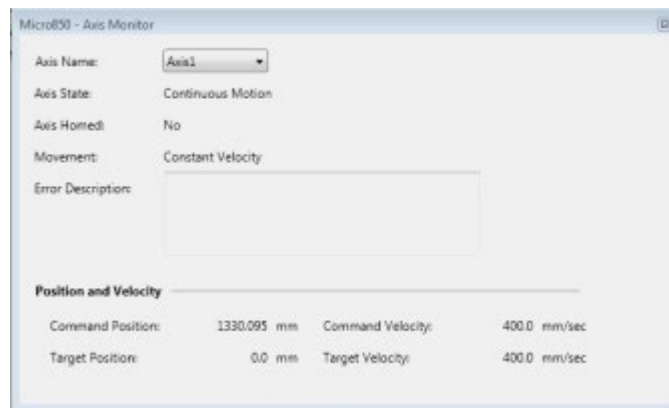
Esempio di testo strutturato MC_MoveVelocity

```

MC_MoveVelocity_1(
void MC_MoveVelocity_1(Axis_REF AxisIn, BOOL Execute, REAL Velocity, REAL Acceleration, REAL Deceleration, REAL Jerk, SINT DirectionIn, SINT BufferMode)
Type : MC_MoveVelocity, Commands a never ending controlled motion at a specified velocity.

Velocity_Velocity := 400.0;
Acceleration_Velocity := 100.0;
Deceleration_Velocity := 100.0;
Jerk_Velocity := 100.0;
DirectionIn_Velocity := 1;
MC_MoveVelocity_1(Axis1,Execute_Velocity,Velocity_Velocity,
Acceleration_Velocity,Deceleration_Velocity,Jerk_Velocity,DirectionIn_Velocity,BufferMode_Velocity);
InVelocity_Velocity := MC_MoveVelocity_1.InVelocity;
Busy_Velocity := MC_MoveVelocity_1.Busy;
Active_Velocity := MC_MoveVelocity_1.Active;
Direction_Velocity := MC_MoveVelocity_1.Direction;
CommandAbort_Velocity := MC_MoveVelocity_1.CommandAborted;
Error_Velocity := MC_MoveVelocity_1.Error;
ErrorID_Velocity := MC_MoveVelocity_1.ErrorID;
    
```

Risultati



Vedere anche

[Tipo di dati AXIS_REF](#) a pagina 432

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Istruzioni di controllo del movimento](#) a pagina 423

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

MC_Power (potenza del controllo movimento)

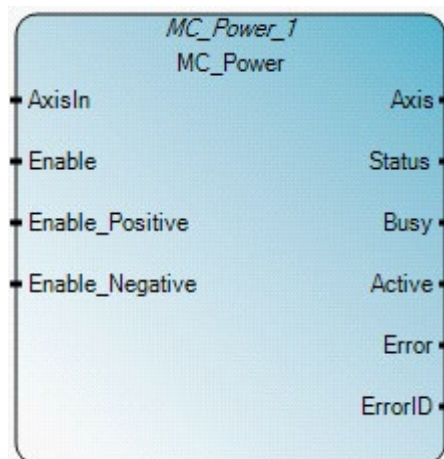
Controlla lo stato di alimentazione, ON o OFF.

Dettagli operazione:

- Se si importa un progetto creato in CCW 7 in CCW 8, viene visualizzato il nuovo parametro di input Mc_Power chiamato __DTI_AxisIn. Se si verifica un errore di compilazione, rizelezionare le istruzioni e ricompilare.
- Dopo che l'asse è stato acceso, lo stato dell'asse Azzerato viene ripristinato a 0 (non azzerato).
- Gli ingressi Enable_Positive ed Enable_Negative per MC_Power sono entrambi attivati su livello e vengono controllati quando l'ingresso di abilitazione passa da OFF a ON. Il passaggio in movimento dell'ingresso Enable_Positive e dell'ingresso Enable_Negative senza attivazione/disattivazione di Abilita ingresso, non è controllato.
- Se si verifica un errore di alimentazione durante il funzionamento (quando viene rilevato Servo pronto), lo stato dell'asse passa a ErrorStop.
- L'istruzione MC_Power ha un valore di timeout di 2 minuti. MC_Power restituisce un errore quando l'intervallo di timeout scade e l'ingresso pronto servomotore è FALSE.
- Se il blocco funzione MC_Power con abilitazione impostata su True viene richiamato mentre lo stato dell'asse è Disabled, lo stato dell'asse passa a Fermo in assenza di errori rilevati, altrimenti lo stato dell'asse passa a ErrorStop in caso di errore.
- Deve essere emesso un solo blocco di funzione MC_Power per asse. L'uso di un differente blocco funzione MC_Power per controllare contemporaneamente lo stesso asse viene rifiutato da Motore movimento.
- In presenza di un commutatore di stato accensione o spegnimento per un asse, la posizione assoluta dell'asse non viene reimpostata.
- Se viene richiamato il blocco funzione MC_Power con Enable impostato su False, lo stato dell'asse passa a Disabled per ogni stato, incluso ErrorStop.
- Il blocco funzione MC_Power può accendersi sull'asse se Enable è impostato su True e spegnere l'asse se Enable è impostato su False.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

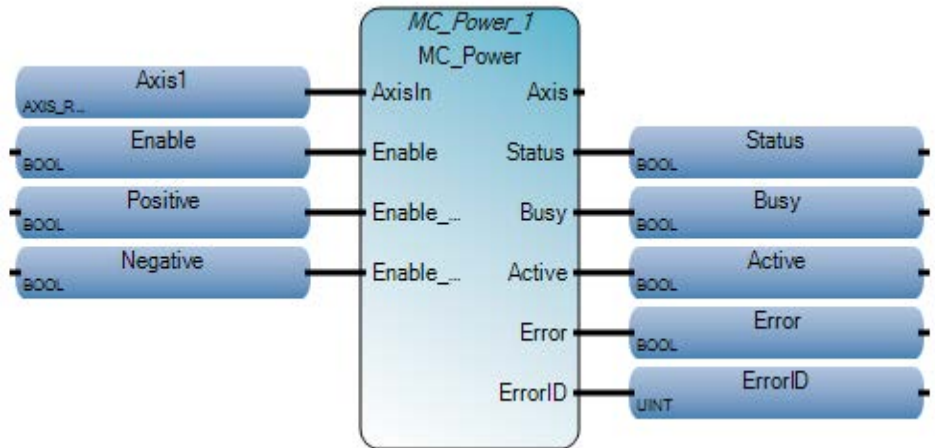
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



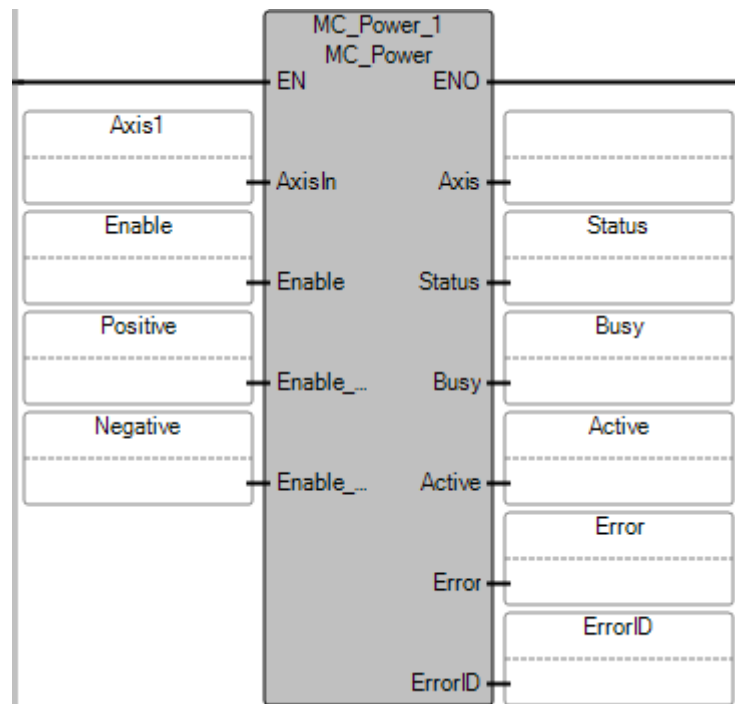
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo corrente MC_Power. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare i parametri del tipo di dati AXIS_REF per definire AxisIn. Per un FB_Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire AxisIn.
Enable	Ingresso	BOOL	TRUE: l'alimentazione è ON. FALSE: l'alimentazione è OFF
Enable_Positive	Ingresso	BOOL	TRUE: il controllo assi è consentito nella direzione positiva.
Enable_Negative	Ingresso	BOOL	TRUE: il controllo assi è consentito nella direzione negativa.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura. Tipo di dati AXIS_REF.
Stato	Uscita	BOOL	Stato dello stadio di alimentazione: <ul style="list-style-type: none"> • TRUE: è stata effettuata l'accensione dell'unità. • FALSE: l'accensione dell'unità non è stata eseguita.
Busy	Uscita	BOOL	TRUE il blocco istruzione non è finito. FALSE: il blocco istruzione è completato.
Attivo	Uscita	BOOL	TRUE: indica che il blocco funzione ha il controllo sull'asse.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempio di diagramma Blocco funzione MC_Power



Esempio di diagramma ladder MC_Power



Esempio di testo strutturato MC_Power

```

MC_Power_1(
void MC_Power_1([AXIS_REF|FB_AXIS_REF] AxisIn, BOOL Enable, BOOL Enable_Positive, BOOL Enable_Negative)
Tipo: MC_Power, Controlla lo stadio di alimentazione (On o Off)

Positive := True;
Negative := True;
MC_Power_1(Axis1, Enable, Positive, Negative);
Status := MC_Power_1.Status;
Busy := MC_Power_1.Busy;
Active := MC_Power_1.Active;
Error := MC_Power_1.Error;
ErrorID := MC_Power_1.ErrorID;

```

Risultati

Nome	Valore logico	Valore base	Bitwise	Tipo di dati
MOTION_DIAG				MOTION_D
Axis1				AXIS_REF
Enable	<input checked="" type="checkbox"/>	N/A		BOOL
Positive	<input checked="" type="checkbox"/>	N/A		BOOL
Negative	<input checked="" type="checkbox"/>	N/A		BOOL
Status	<input checked="" type="checkbox"/>	N/A		BOOL
Busy	<input type="checkbox"/>	N/A		BOOL
Active	<input checked="" type="checkbox"/>	N/A		BOOL
Error	<input type="checkbox"/>	N/A		BOOL
ErrorID	<input type="checkbox"/>	N/A		UINT

Vedere anche

[Istruzioni di controllo del movimento](#) a pagina 423

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

[Tipo di dati AXIS_REF](#) a pagina 432

[Tipo di dati FB_AXIS_REF](#) a pagina 433

MC_ReadActualPosition (posizione effettiva di lettura del controllo movimento)

Restituisce la posizione effettiva dell'asse di feedback. MC_ReadActualPosition è applicabile solo al movimento di retroazione.

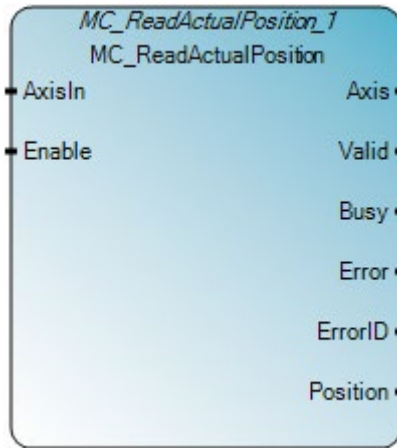
Dettagli operazione:

- Prima di eseguire MC_ReadActualPosition, verificare l'asse in uno dei seguenti stati dell'asse:
 - Disattivato
 - Fermo
 - Movimento discreto
 - Arresto errore

- La posizione effettiva per un asse di feedback non viene reimpostata su 0 dopo un download. Per eseguire il reset o cancellare la posizione per un asse di feedback, utilizzare l'istruzione MC_Home o l'istruzione MC_SetPosition.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

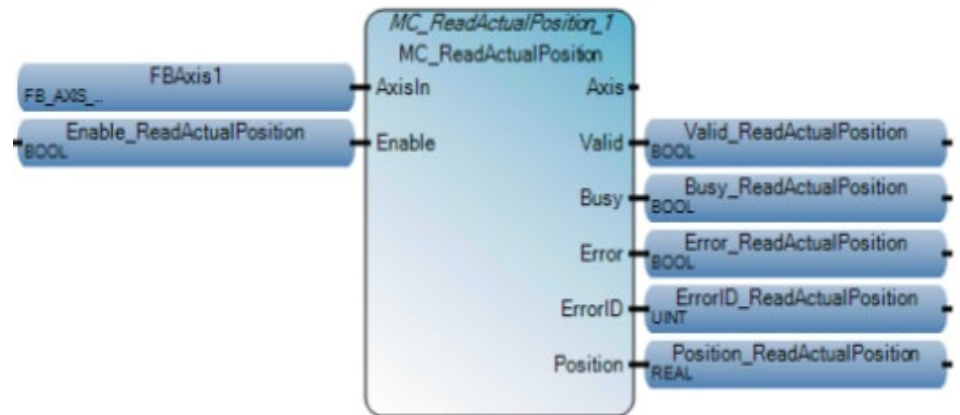
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



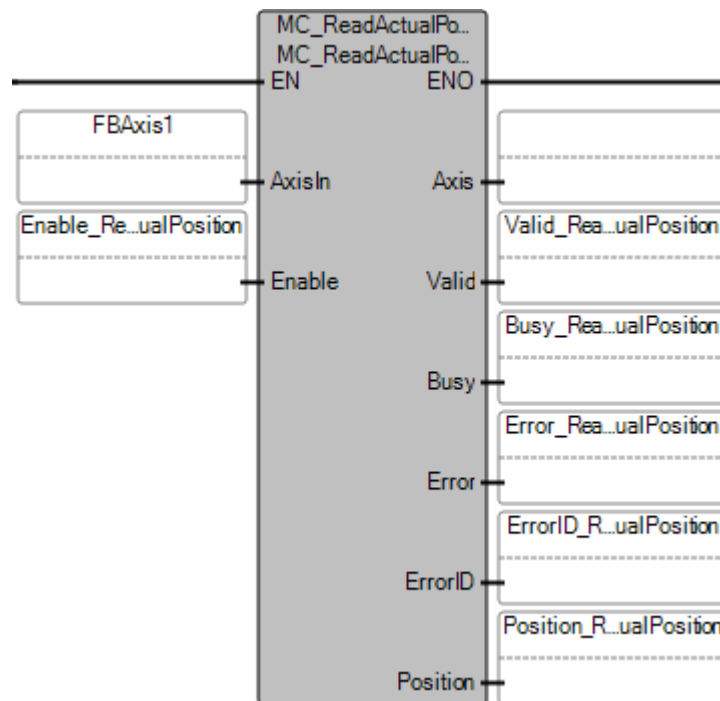
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
AxisIn	Ingresso	FB_AXIS_REF	Per un FB_Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire AxisIn.
Enable	Ingresso	BOOL	TRUE: ottenere continuamente il valore del parametro quando abilitato. FALSE: non attivo.
Asse	Uscita	FB_AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura. I parametri di uscita dell'asse sono definiti nel tipo di dati FB_AXIS_REF.
Valid	Uscita	BOOL	TRUE: il blocco istruzione è attivo e sono previsti nuovi valori in uscita. FALSE: il blocco istruzione non è attivo.
Busy	Uscita	BOOL	TRUE il blocco istruzione non è finito. FALSE: il blocco istruzione è completato.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UNIT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.
Position	Uscita	REAL	Valore della posizione assoluta effettiva per l'asse di movimento di retroazione (in unità dell'asse [u]).

Esempio di Programmazione a blocchi funzionali MC_ReadActualPosition



Esempio di Diagramma ladder MC_ReadActualPosition



Esempio di Testo strutturato MC_ReadActualPosition

```
void MC_ReadActualPosition_1(
    void MC_ReadActualPosition_1(FB_AXIS_REF AxisIn, BOOL Enable)
    Tipo: MC_ReadActualPosition, Ripristina la posizione effettiva per un asse.

    MC_ReadActualPosition_1(FBAxis, Enable_ReadActualPosition)
    Valid_ReadActualPosition := MC_ReadActualPosition_1.Valid
    Busy_ReadActualPosition := MC_ReadActualPosition_1.Busy
    Error_Valid_ReadActualPosition := MC_ReadActualPosition_1.Error
    ErrorID_Valid_ReadActualPosition := MC_ReadActualPosition_1.ErrorID
    Position_Valid_ReadActualPosition := MC_ReadActualPosition_1.Position
```

Risultati

Micro830 - Monitoraggio asse

Nome asse:

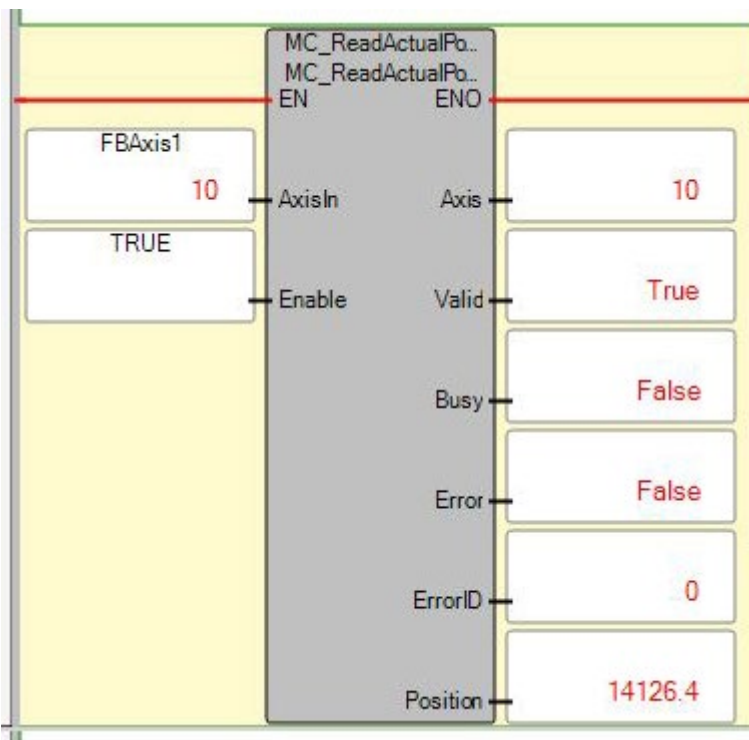
Stato asse: Disattivato

Movimento: Inattivo

Descrizione errore:

Posizione e velocità

Posizione effettiva: 26169.91 mm Velocità effettiva: 24.39024 mm/sec



	Nome	Alias	Valore logico
-	MC_ReadActualPosition_2		...
▶ +	MC_ReadActualPosition_2.AxisIn		...
	MC_ReadActualPosition_2.Enable		<input checked="" type="checkbox"/>
+	MC_ReadActualPosition_2.Axis		...
	MC_ReadActualPosition_2.Valid		<input checked="" type="checkbox"/>
	MC_ReadActualPosition_2.Busy		<input type="checkbox"/>
	MC_ReadActualPosition_2.Error		<input type="checkbox"/>
	MC_ReadActualPosition_2.ErrorID		0
	MC_ReadActualPosition_2.Position		14126.3945

Vedere anche

[Tipo di dati FB_AXIS_REF](#) a pagina 433

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Istruzioni di controllo del movimento](#) a pagina 423

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

MC_ReadActualVelocity (velocità effettiva di lettura del controllo movimento)

Restituisce il valore della velocità corrente dell'asse di feedback.

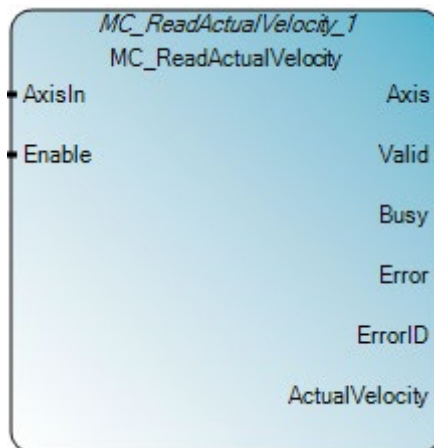
MC_ReadActualVelocity è applicabile solo al movimento di retroazione.

Prima di eseguire MC_ReadActualVelocity, verificare l'asse in uno dei seguenti stati dell'asse:

- Fermo
- Movimento discreto
- Arresto errore

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.

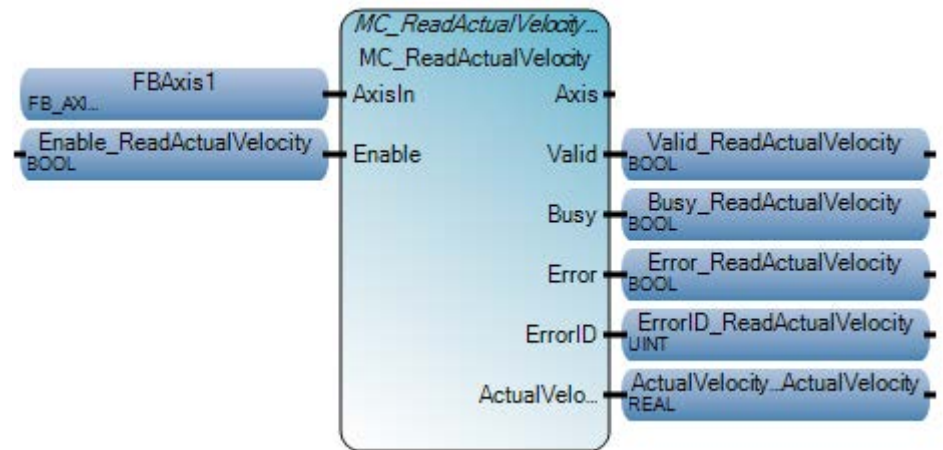


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

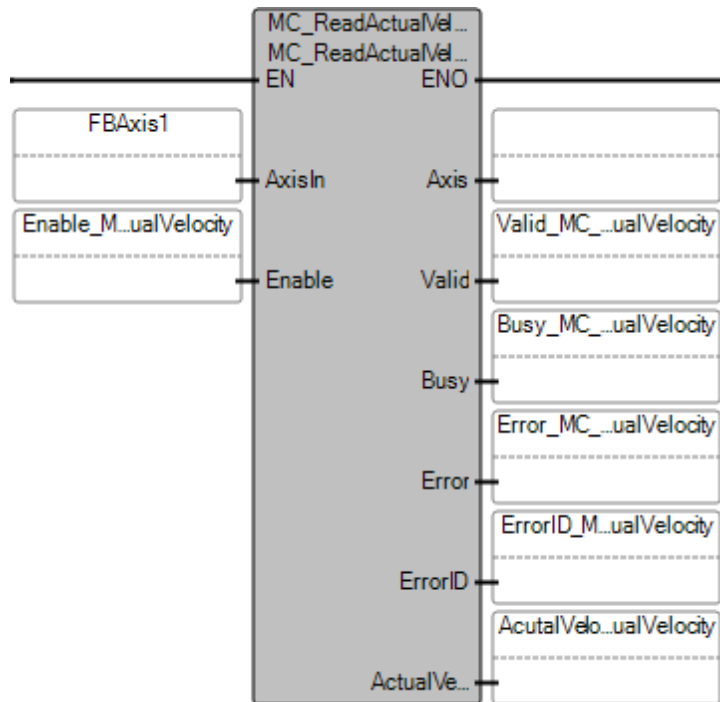
Parametro	Tipo di parametro	Tipo di dati	Descrizione
AxisIn	Ingresso	FB_AXIS_REF	Per un FB.Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire AxisIn.
Enable	Ingresso	BOOL	TRUE: ottiene continuamente il valore per la velocità effettiva. FALSE: i dati non sono più validi, tutte le uscite vengono ripristinate su 0 e il parametro Valid viene impostato su False.
Asse	Uscita	FB_AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura. I parametri di uscita dell'asse sono definiti nel tipo di dati FB_AXIS_REF.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Valid	Uscita	BOOL	TRUE: il blocco funzione è attivo e sono previsti nuovi valori in uscita. FALSE: la funzione non prevede nuovi valori in uscita.
Busy	Uscita	BOOL	TRUE il blocco funzione non è finito. FALSE: la funzione non è attiva.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UNIT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.
ActualVelocity	Uscita	REAL	Il valore della velocità effettiva per l'asse Motion feedback (in unità dell'asse [u/s]). ActualVelocity è un valore con segno, che include le informazioni di direzione.

Esempio di Programmazione a blocchi funzionali MC_ReadActualVelocity



Esempio di Diagramma ladder MC_ReadActualVelocity



Esempio di Testo strutturato MC_ReadActualVelocity

```
void MC_ReadActualVelocity_1(
    void MC_ReadActualVelocity_1(FB_AXIS_REF AxisIn, BOOL Enable)
    Tipo: MC_ReadActualVelocity, Ripristina la velocità effettiva per un asse.

    MC_ReadActualVelocity_1(FBAxis1, Enable_ReadActualVelocity)
    Valid_ReadActualVelocity := MC_ReadActualVelocity_1.Valid
    Busy_ReadActualVelocity := MC_ReadActualVelocity_1.Busy
    Error_ReadActualVelocity := MC_ReadActualVelocity_1.Error
    ErrorID_ReadActualVelocity := MC_ReadActualVelocity_1.ErrorID
    Position_ReadActualVelocity := MC_ReadActualVelocity_1.ErrorID
```

Risultati

Micro830 - Monitoraggio asse

Nome asse:

Stato asse: Movimento discreto

Movimento: In accelerazione

Descrizione errore:

Posizione e velocità

Posizione effettiva: 16611.12 mm Velocità effettiva: 25.12563 mm/sec

	Nome	Alias	Valore logico
▶	- MC_ReadActualVelocity_1		...
	+ MC_ReadActualVelocity_1.AxisIn		...
	MC_ReadActualVelocity_1.Enable		<input checked="" type="checkbox"/>
	+ MC_ReadActualVelocity_1.Axis		...
	MC_ReadActualVelocity_1.Valid		<input checked="" type="checkbox"/>
	MC_ReadActualVelocity_1.Busy		<input type="checkbox"/>
	MC_ReadActualVelocity_1.Error		<input type="checkbox"/>
	MC_ReadActualVelocity_1.ErrorID		0
	MC_ReadActualVelocity_1.ActualVelocity		24.1545887

Vedere anche

[Tipo di dati FB_AXIS_REF](#) a pagina 433

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Istruzioni di controllo del movimento](#) a pagina 423

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

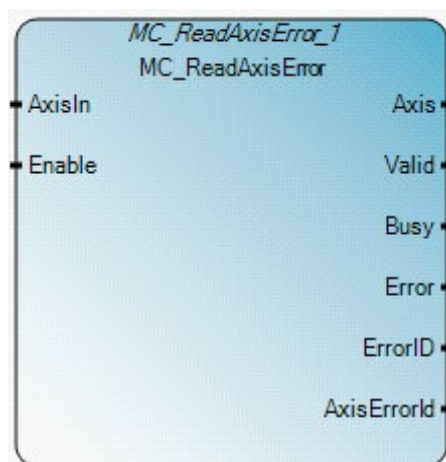
MC_ReadAxisError (errore dell'asse di lettura del controllo movimento)

Legge gli errori dell'asse non in relazione ai blocchi funzione Controllo movimento.

Quando un asse è in stato Disabilitato, MC_ReadAxisError può ottenere o meno un ID errore diverso da zero per l'asse, in quanto un asse Disabilitato può contenere o meno errori.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.

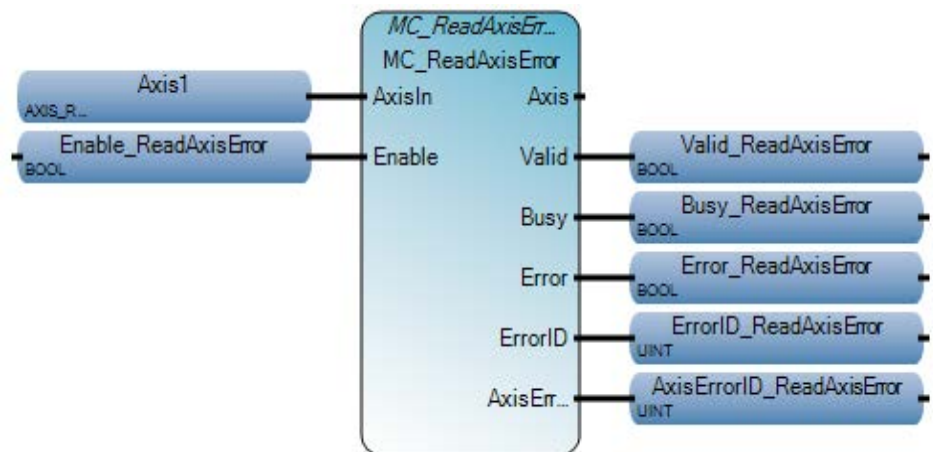


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

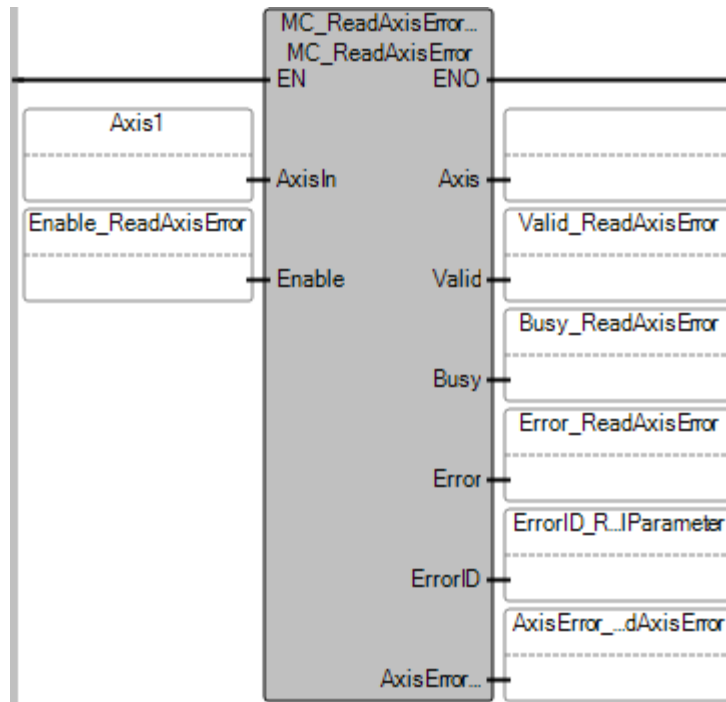
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo corrente MC_ReadAxisError. FALSE: Error, ErrorID e AxisErrorID vengono ripristinati su False (o 0). Applicabile solo alle programmazioni LD.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare i parametri del tipo di dati AXIS_REF per definire AxisIn. Per un FB_Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire AxisIn.
Enable	Ingresso	BOOL	TRUE: ottenere continuamente il valore del parametro quando abilitato. FALSE: reimposta Error, ErrorID, e AxisErrorID uscite su 0.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura. I parametri di uscita dell'asse sono definiti nel tipo di dati FB_AXIS_REF.
Valid	Uscita	BOOL	TRUE: il blocco istruzione è attivo e sono previsti nuovi valori in uscita. FALSE: il blocco istruzione non è attivo.
Busy	Uscita	BOOL	TRUE il blocco istruzione non è finito. FALSE: il blocco istruzione è completato.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.
AxisErrorID	Uscita	UINT	Numero univoco che identifica l'errore dell'asse. Gli errori per questa istruzione sono definiti nei codici di errore AxisErrorID.

Esempio di Programmazione a blocchi funzionali **MC_ReadAxisError**



Esempio di Diagramma ladder **MC_ReadAxisError**



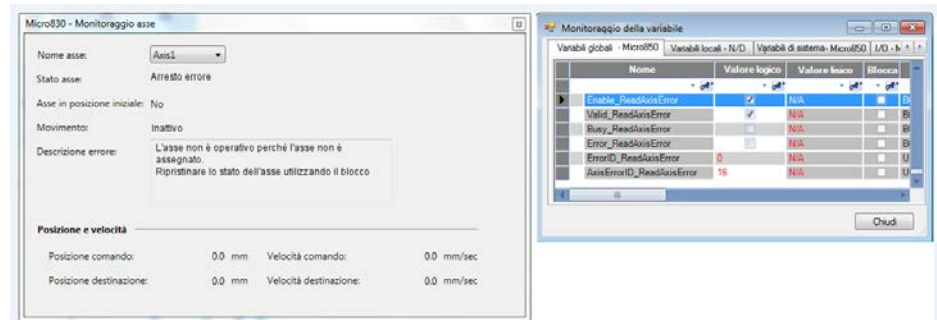
Esempio di Testo strutturato **MC_ReadAxisError**

```

MC_ReadAxisError_1(
void MC_ReadAxisError_1([AXIS_REF|FB_AXIS_REF] AxisIn, BOOL Enable)
Tipo: MC_ReadAxisError, Legge le informazioni dell'errore per un asse.

MC_ReadAxisError_1(Axis1, Enable_ReadAxisError);
Valid_ReadAxisError := MC_ReadAxisError_1.Valid;
Busy_ReadAxisError := MC_ReadAxisError_1.Busy;
Error_ReadAxisError := MC_ReadAxisError_1.Error;
ErrorID_ReadAxisError := MC_ReadAxisError_1.ErrorID;
AxisErrorID_ReadAxisError := MC_ReadAxisError_1.AxisErrorId;
    
```

Risultati



Vedere anche

[Tipo di dati AXIS_REF](#) a pagina 432

[Codici di errore AxisErrorID](#) a pagina 471

[Tipo di dati FB_AXIS_REF](#) a pagina 433

[Istruzioni di controllo del movimento](#) a pagina 423

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

Codici di errore AxisErrorID

Nella seguente tabella sono descritti i codici di errore degli assi identificati in AxisErrorID per MC_ReadAxisError.

Valore	ID MACRO	Descrizione
00	MC_FB_ERR_NO	L'asse è in stato operativo (nulla da visualizzare).
01	MC_FB_ERR_WRONG_STATE	L'asse non funziona a causa di uno stato dell'asse non corretto rilevato durante l'esecuzione di un blocco funzione. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power.
02	MC_FB_ERR_RANGE	L'asse non funziona a causa di un parametro dinamico dell'asse non valido (velocità, accelerazione, decelerazione o strappo), impostato in un blocco funzione. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power. Nel blocco funzione correggere tutte le impostazioni dei parametri dinamici in conflitto con le impostazioni della pagina di configurazione Dinamica asse.
03	MC_FB_ERR_PARAM	L'asse non funziona a causa di un parametro non valido (diverso da velocità, accelerazione, decelerazione o strappo), impostato in un blocco funzione. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power. Nel blocco funzione, correggere le impostazioni dei parametri, come modalità o posizione.
04	MC_FB_ERR_AXISNUM	Errore interno movimento, ID errore = 0x04. Contattare il rappresentante del supporto tecnico di zona di Rockwell Automation. Per le informazioni sui contatti, visitare: http://www.rockwellautomation.com/support
05	MC_FB_ERR_MECHAN	L'asse non funziona perché è stato rilevato un problema all'unità o meccanico. Controllare il collegamento tra unità e controllore (segnali Pronto unità e In posizione), quindi verificare che l'unità funzioni normalmente. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power.
06	MC_FB_ERR_NOPOWER	L'asse non è alimentato. Alimentare l'asse con il blocco funzione MC_Power. Ripristinare lo stato dell'asse utilizzando il blocco funzione MC_Reset.
07	MC_FB_ERR_RESOURCE	L'asse non funziona in quanto esso o la risorsa relativa richiesta dal blocco funzione sono sotto controllo di un altro blocco funzione, o non disponibili. Verificare che l'asse o la relativa risorsa richiesta dal blocco funzione sia disponibile per l'uso. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power. Riesaminare e correggere l'applicazione in caso di istanze multiple dello stesso blocco funzione che tentano di controllare l'asse contemporaneamente.

Valore	ID MACRO	Descrizione
08	MC_FB_ERR_PROFILE	L'asse non funziona in quanto un profilo di movimento definito in un blocco funzione non è valido. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power. Correggere il profilo nel blocco funzione.
09	MC_FB_ERR_VELOCITY	L'asse non funziona in quanto un profilo di movimento richiesto in un blocco funzione è in conflitto con la velocità corrente dell'asse. Possibili cause: <ul style="list-style-type: none"> • Il blocco funzione richiede che l'asse inverta la direzione mentre l'asse è in movimento. • La velocità corrente è troppo bassa o troppo alta per il profilo di movimento richiesto. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power. Correggere il profilo di movimento nel blocco funzione o rieseguire il blocco funzione quando la velocità dell'asse è compatibile con il profilo di movimento richiesto.
10	MC_FB_ERR_SOFT_LIMIT	L'asse non funziona per via di un errore di limite software rilevato o per un errore di limite software causato dall'esecuzione del blocco funzione. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power. Controllare le impostazioni di velocità o posizione destinazione per il blocco funzione, oppure regolare il Limite software.
11	MC_FB_ERR_HARD_LIMIT	L'asse non funziona a causa del rilevamento di un errore Limite hardware. Reimpostare lo stato dell'asse utilizzando il blocco funzione MC_Reset, quindi spostare l'asse dall'interruttore Limite hardware in direzione opposta.
12	MC_FB_ERR_LOG_LIMIT	L'asse non funziona per via di un errore limite logico dell'accumulatore PTO rilevato o per un errore di limite logico dell'accumulatore PTO causato dall'esecuzione del blocco funzione. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power. Controllare le impostazioni di velocità o posizione destinazione per il blocco funzione. Utilizzare il blocco funzione MC_SetPosition per regolare il sistema di coordinate dell'asse.
13	MC_FB_ERR_ERR_ENGINE	L'asse non funziona in seguito al rilevamento di un errore di esecuzione del motore di movimento. Eseguire il ciclo di alimentazione dell'intera macchina e scaricare nuovamente l'applicazione utente nel controllore. Se l'errore persiste, contattare il rappresentante di assistenza tecnica di zona di Rockwell Automation. Per le informazioni sui contatti, visitare: http://www.rockwellautomation.com/support .
16	MC_FB_ERR_NOT_HOMED	L'asse non funziona in quanto l'asse non è in sede. Ripristinare lo stato dell'asse utilizzando i blocchi funzione MC_Reset e MC_Power. Riposizionare nuovamente l'asse in sede usando il blocco funzione MC_Home.
128	MC_FB_ERR_PARAM_MODIFIED	Avviso interno movimento, ID avviso = 0x80. Contattare il rappresentante del supporto tecnico di zona di Rockwell Automation. Per le informazioni sui contatti, visitare: http://www.rockwellautomation.com/support .

Vedere anche

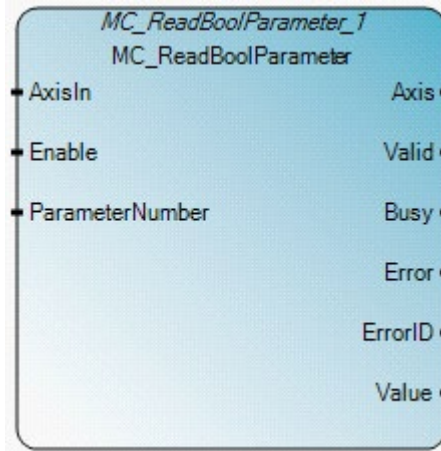
[MC_ReadAxisError](#) a pagina 468

MC_ReadBoolParameter (parametro BOOL di lettura del controllo movimento)

Restituisce il valore di un parametro specifico del fornitore di tipo BOOL.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

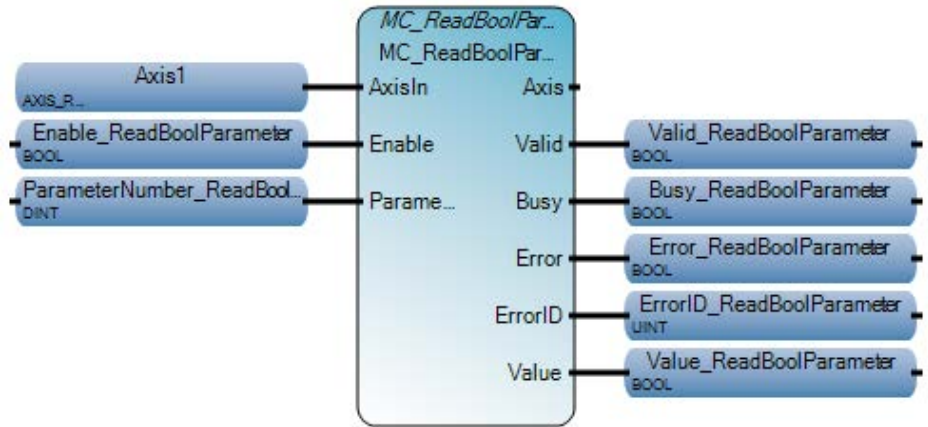
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



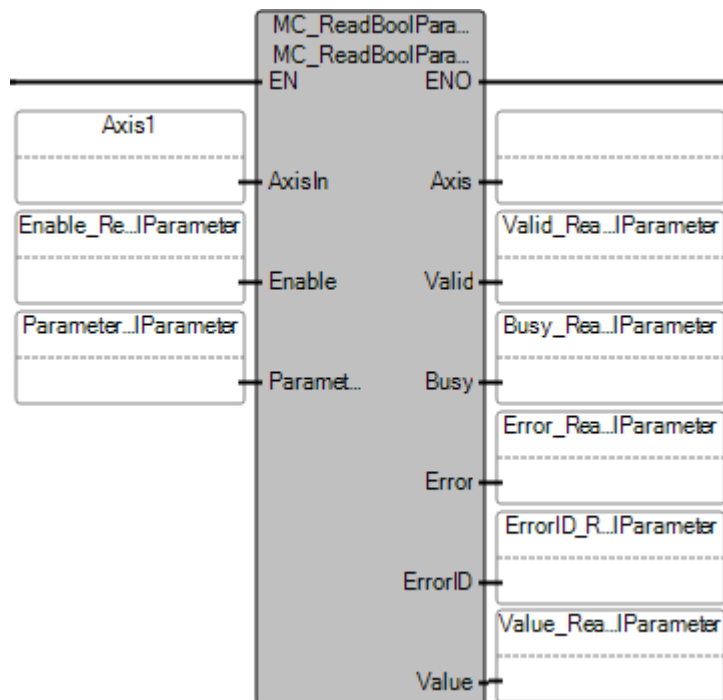
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_ReadBoolParameter corrente. FALSE: ripristina l'uscita del valore su 0. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare il tipo di dati AXIS_REF per definire AxisIn. Per FB_Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire AxisIn.
Enable	Ingresso	BOOL	TRUE: ottenere continuamente il valore del parametro quando abilitato. FALSE: l'uscita del valore viene ripristinata su 0.
ParameterNumber	Ingresso	DINT	Identificazione del parametro. Le definizioni dei numeri di parametro sono definite nei numeri dei parametri del blocco funzione Controllo movimento.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Valid	Uscita	BOOL	TRUE: il valore del parametro è disponibile. FALSE: il valore del parametro non è disponibile.
Busy	Uscita	BOOL	TRUE: il blocco funzione è operativo e sono previsti nuovi valori in uscita. FALSE: la funzione non è attiva.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.
Valore	Uscita	BOOL	Valore del parametro specificato nel tipo di dato, come specificato dal fornitore.

Esempio di Programmazione a blocchi funzionali **MC_ReadBoolParameter**



Esempio di Diagramma ladder **MC_ReadBoolParameter**



Esempio di Testo strutturato **MC_ReadBoolParameter**

```

MC_ReadBoolParameter_1(
void MC_ReadBoolParameter_1([AXIS_REF|FB_AXIS_REF] AxisIn, BOOL Enable, DINT ParameterNumber)
Tipo: MC_ReadBoolParameter, Ripristina il valore di un parametro BOOL specifico del movimento.

ParameterNumber_ReadBoolParameter := 4;
MC_ReadBoolParameter_1(Axis1, Enable_ReadBoolParameter, ParameterNumber_ReadBoolParameter);
Valid_ReadBoolParameter := MC_ReadBoolParameter_1.Valid;
Busy_ReadBoolParameter := MC_ReadBoolParameter_1.Busy;
Error_ReadBoolParameter := MC_ReadBoolParameter_1.Error;
ErrorID_ReadBoolParameter := MC_ReadBoolParameter_1.ErrorID;
Value_ReadBoolParameter := MC_ReadBoolParameter_1.Value;
    
```

Risultati

Nome	Valore logico	Valore fisico	Blocca	T
Enable_ReadBoolParameter	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOI
ParameterNumber_ReadBoolParameter	4	N/D	<input type="checkbox"/>	DINT
Valid_ReadBoolParameter	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOI
Busy_ReadBoolParameter	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOI
Error_ReadBoolParameter	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOI
ErrorID_ReadBoolParameter	0	N/D	<input type="checkbox"/>	UINT
Value_ReadBoolParameter	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOI

Vedere anche

[Tipo di dati AXIS_REF](#) a pagina 432

[Tipo di dati FB_AXIS_REF](#) a pagina 433

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

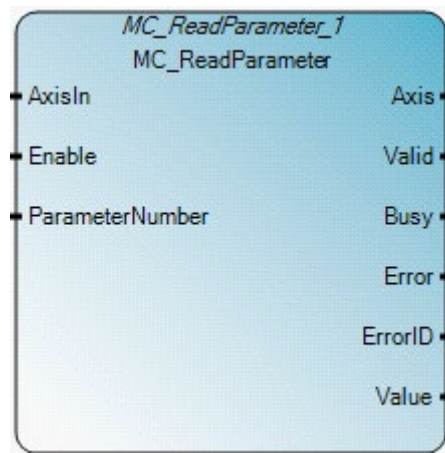
[Istruzioni di controllo del movimento](#) a pagina 423

MC_ReadParameter (parametro di lettura del controllo movimento)

Restituisce il valore di un parametro specifico del fornitore in un tipo Real.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

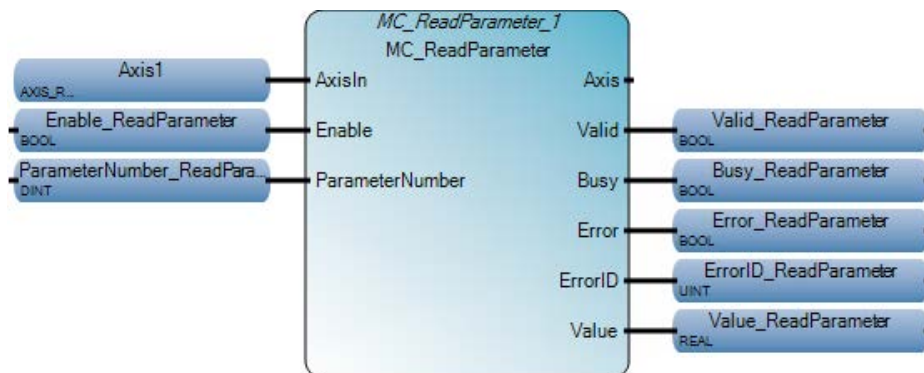
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



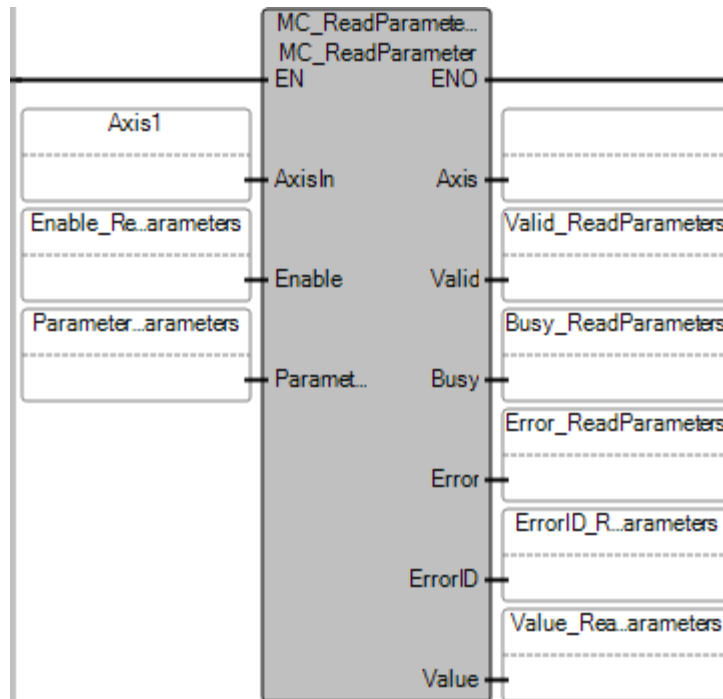
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_ReadParameter corrente. FALSE: l'uscita del valore viene ripristinata su 0. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare il tipo di dati AXIS_REF per definire AxisIn. Per FB_Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire AxisIn.
Enable	Ingresso	BOOL	TRUE: ottenere il valore del numero del parametro in modo continuo. FALSE: l'uscita del valore viene ripristinata su 0.
ParameterNumber	Ingresso	DINT	Identificazione del parametro. Le definizioni dei numeri di parametro sono definite nei numeri dei parametri del blocco funzione Controllo movimento.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Valid	Uscita	BOOL	TRUE: sono disponibili valori validi in uscita. FALSE: non sono disponibili valori validi in uscita.
Busy	Uscita	BOOL	TRUE: il blocco funzione è operativo e sono previsti nuovi valori in uscita. FALSE: il blocco funzione non è attivo.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.
Valore	Uscita	REAL	Valore del parametro specificato nel tipo di dato, come specificato dal fornitore.

Esempio di Programmazione a blocchi funzionali MC_ReadParameter



Esempio di Diagramma ladder MC_ReadParameter

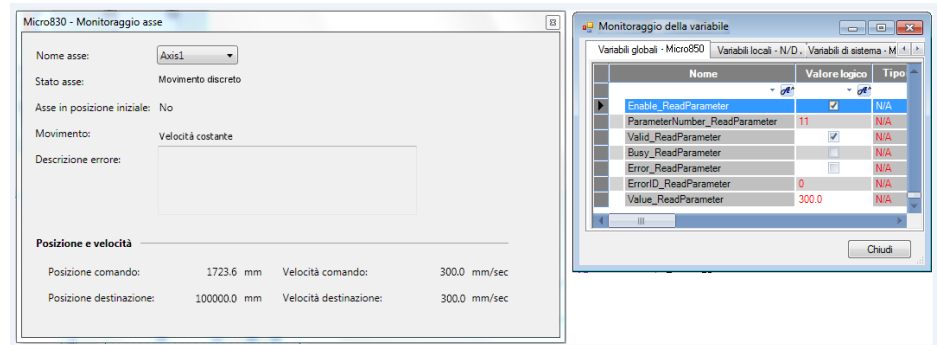


Esempio di Testo strutturato MC_ReadParameter

```
MC_ReadParameter_1(
void MC_ReadParameter_1([AXIS_REF|FB_AXIS_REF] AxisIn, BOOL Enable, DINT ParameterNumber)
Tipo: MC_ReadParameter, Ripristina il valore di un parametro REAL specifico del movimento.

ParameterNumber_ReadParameter := 11;
MC_ReadParameter_1(Axis1, Enable_ReadParameter, ParameterNumber_ReadParameter);
Valid_ReadParameter := MC_ReadParameter_1.Valid;
Busy_ReadParameter := MC_ReadParameter_1.Busy;
Error_ReadParameter := MC_ReadParameter_1.Error;
ErrorID_ReadParameter := MC_ReadParameter_1.ErrorID;
Value_ReadParameter := MC_ReadParameter_1.Value;
```

Risultati



Vedere anche

[Tipo di dati AXIS_REF](#) a pagina 432

[Tipo di dati FB_AXIS_REF](#) a pagina 433

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

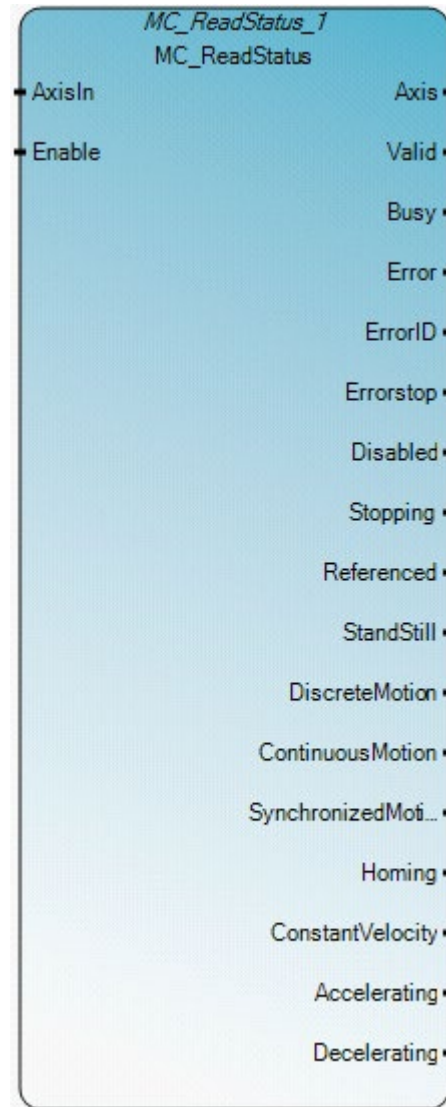
[Istruzioni di controllo del movimento](#) a pagina 423

MC_ReadStatus (stato di lettura del controllo movimento)

Restituisce lo stato dell'asse rispetto al movimento attualmente in corso.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.

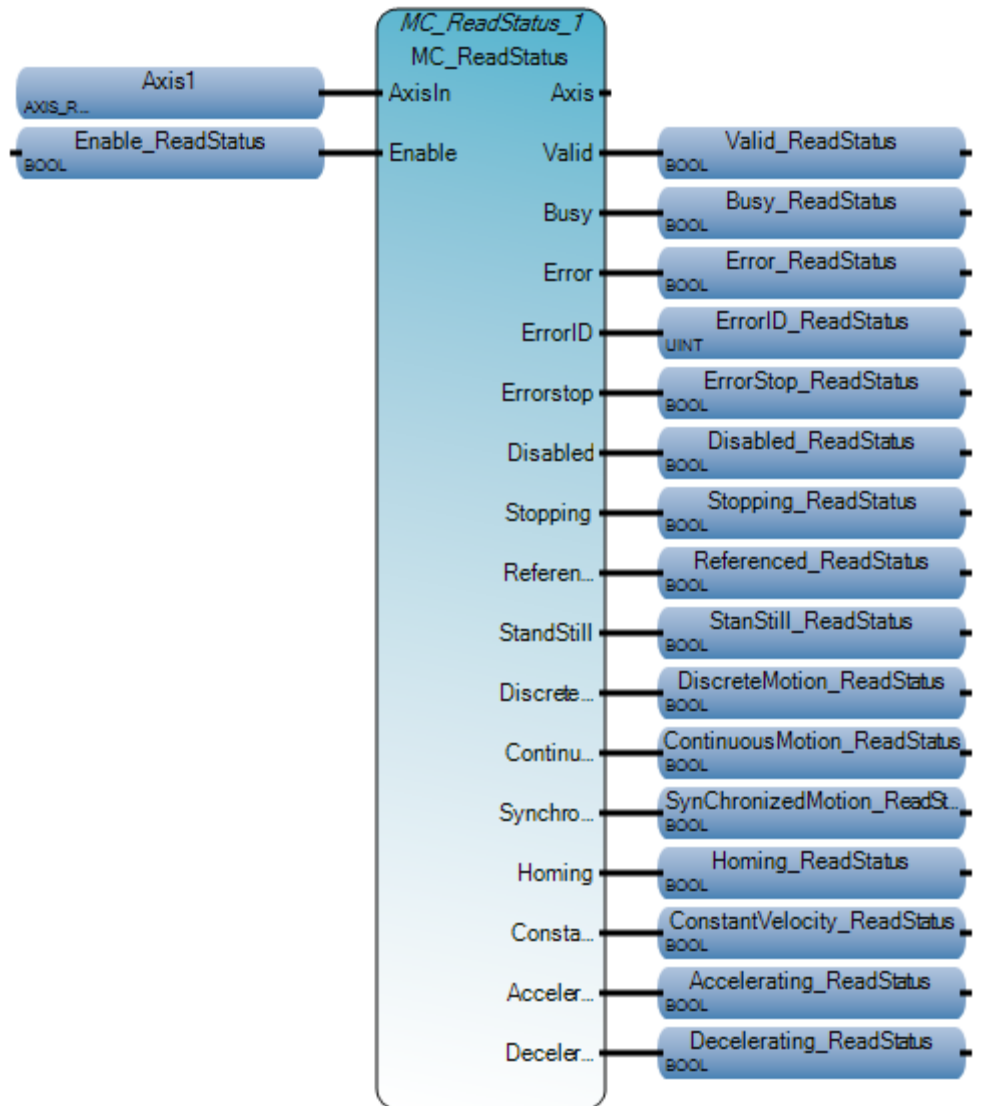


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

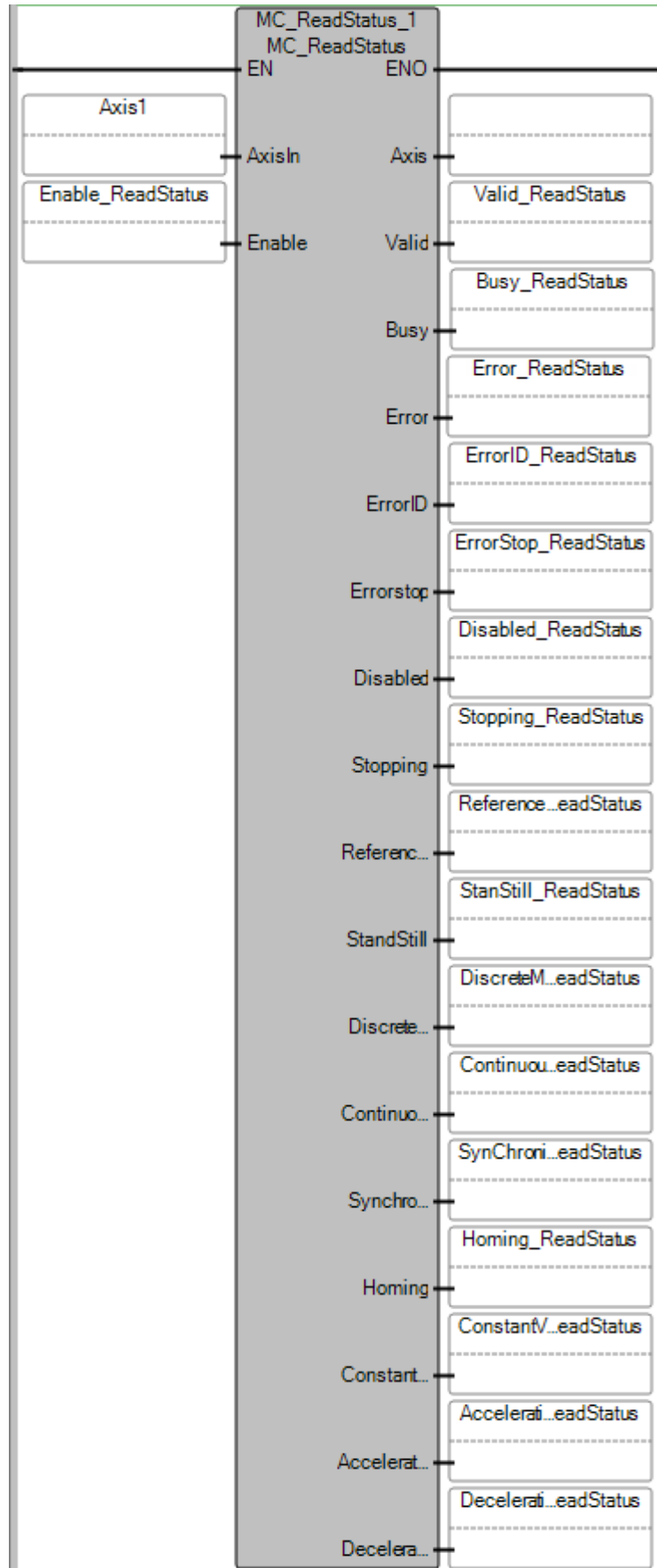
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_ReadStatus corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare i parametri del tipo di dati AXIS_REF per definire AxisIn. Per un FB_Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire AxisIn.
Enable	Ingresso	BOOL	TRUE: ottiene continuamente lo stato dell'asse. FALSE: tutte le uscite dello stato sono ripristinate su 0.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Valid	Uscita	BOOL	TRUE: sono disponibili valori validi in uscita. FALSE: uscite non disponibili.
Busy	Uscita	BOOL	TRUE: il blocco funzione è operativo e sono previsti nuovi valori in uscita. FALSE: il blocco funzione non è attivo.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.
ErrorStop	Uscita	BOOL	TRUE: lo stato dell'asse è ErrorStop. Gli stati dell'asse sono definiti nei nomi e nei valori di stato dell'asse controllo movimento.
Disattivato	Uscita	BOOL	TRUE: lo stato dell'asse è Disabilitato.
Arresto in corso	Uscita	BOOL	TRUE: lo stato dell'asse è Arresto.
Referenced	Uscita	BOOL	TRUE: lo stato dell'asse è in sede, la posizione del riferimento assoluto è nota per l'asse.
StandStill	Uscita	BOOL	TRUE: lo stato dell'asse è Fermo.
DiscreteMotion	Uscita	BOOL	TRUE: lo stato dell'asse è DiscreteMotion.
ContinuousMotion	Uscita	BOOL	TRUE: lo stato dell'asse è ContinuousMotion.
SynchronizedMotion	Uscita	BOOL	Il movimento sincronizzato non è supportato dai controllori Micro800. TRUE: mai true. FALSE: sempre false.
Azzeramento	Uscita	BOOL	TRUE: lo stato dell'asse è Homing.
ConstantVelocity	Uscita	BOOL	TRUE: la velocità del motore è costante.
In accelerazione	Uscita	BOOL	TRUE: l'asse è in accelerazione, aumento dell'energia al motore.
In decelerazione	Uscita	BOOL	TRUE: l'asse è in decelerazione, diminuzione dell'energia al motore.

Esempio di Programmazione a blocchi funzionali MC_ReadStatus



Esempio di Diagramma ladder MC_ReadStatus

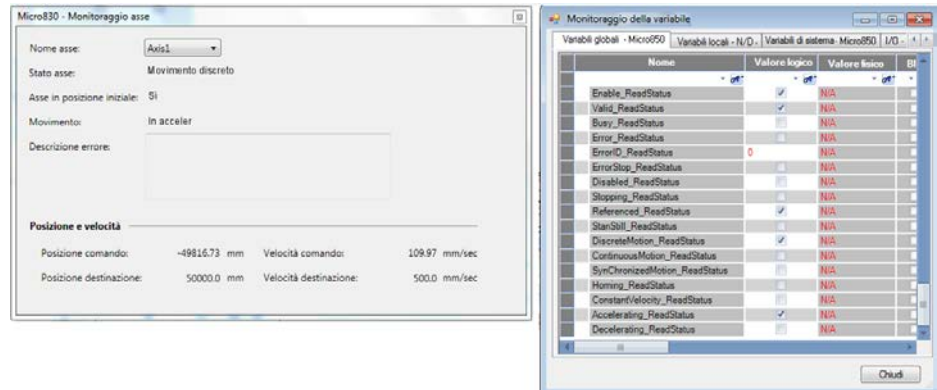


Esempio di Testo strutturato MC_ReadStatus

```
MC_ReadStatus_1(
void MC_ReadStatus_1([AXIS_REF|FB_AXIS_REF] AxisIn, BOOL Enable)
Tipo: MC_ReadStatus, Ripristina nel dettaglio lo stato dell'asse in relazione al movimento in corso.
```

```
MC_ReadStatus_1(Axis1, Enable_ReadStatus);
Valid_ReadStatus := MC_ReadStatus_1.Valid;
Busy_ReadStatus := MC_ReadStatus_1.Busy;
Error_ReadStatus := MC_ReadStatus_1.Error;
ErrorID_ReadStatus := MC_ReadStatus_1.ErrorID;
ErrorStop_ReadStatus := MC_ReadStatus_1.Errorstop;
Disabled_ReadStatus := MC_ReadStatus_1.Disabled;
Stopping_ReadStatus := MC_ReadStatus_1.Stopping;
Referenced_ReadStatus := MC_ReadStatus_1.Referenced;
StandStill_ReadStatus := MC_ReadStatus_1.StandStill;
DiscreteMotion_ReadStatus := MC_ReadStatus_1.DiscreteMotion;
ContinuousMotion_ReadStatus := MC_ReadStatus_1.ContinuousMotion;
SynchronizedMotion_ReadStatus := MC_ReadStatus_1.SynchronizedMotion;
Homing_ReadStatus := MC_ReadStatus_1.Homing;
ConstantVelocity_ReadStatus := MC_ReadStatus_1.ConstantVelocity;
Accelerating_ReadStatus := MC_ReadStatus_1.Accelerating;
Decelerating_ReadStatus := MC_ReadStatus_1.Decelerating;
```

Risultati



Vedere anche

- [Tipo di dati AXIS_REF](#) a pagina 432
- [Tipo di dati FB_AXIS_REF](#) a pagina 433
- [Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424
- [Istruzioni di controllo del movimento](#) a pagina 423
- [Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

MC_Reset (reset del controllo movimento)

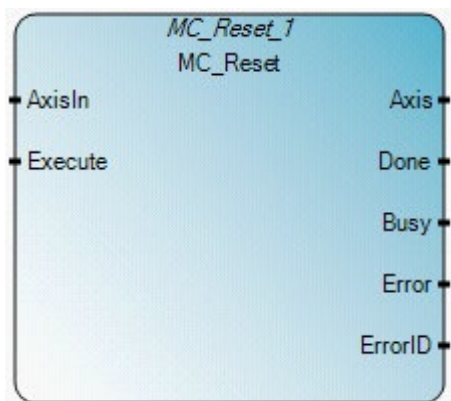
Trasforma lo stato dell'asse da ErrorStop a StandStill ripristinando tutti gli errori interni legati all'asse. Le uscite delle istanze del blocco di funzione non vengono cambiate.

Dettagli operazione:

- Se lo stato di allarme dell'asse non è stato modificato dopo aver eseguito MC_Reset, eseguire MC_Power quindi MC_Reset.
- MC_Reset ripristina solo lo stato dell'asse da ErrorStop a Fermo. L'esecuzione di MC_Reset quando l'asse è in altri stati, incluso Disabilitato, comporta un errore e non influisce sul movimento in corso o sullo stato dell'asse.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

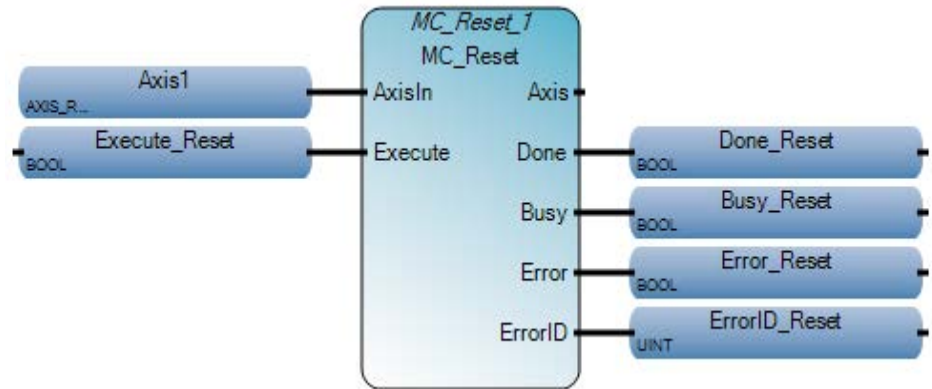
Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



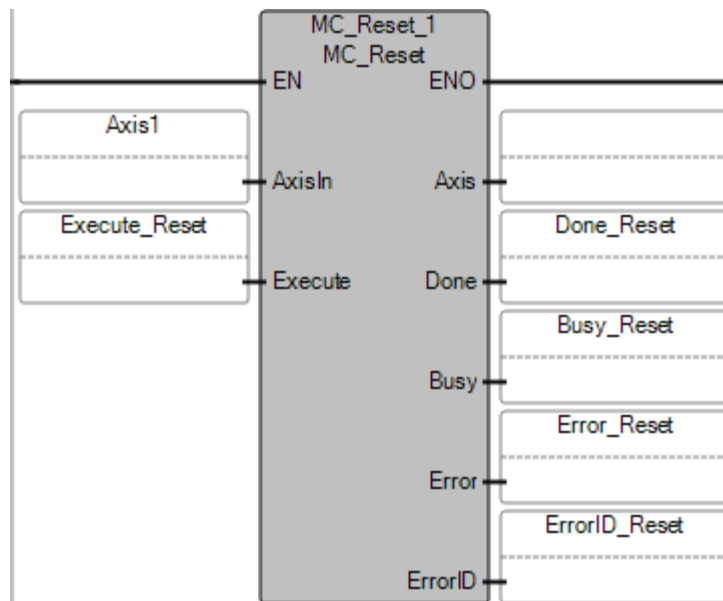
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_Reset corrente. FALSE: il calcolo non viene eseguito. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare il tipo di dati AXIS_REF per definire AxisIn. Per FB_Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire AxisIn.
Execute	Ingresso	BOOL	TRUE: ripristina l'asse al fronte di salita.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Finito	Uscita	BOOL	TRUE: lo stato dell'asse è Fermo o Disabilitato.
Busy	Uscita	BOOL	TRUE il blocco funzione non è finito.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempio di Programmazione a blocchi funzionali MC_Reset



Esempio di Diagramma ladder MC_Reset



Esempio di Testo strutturato MC_Reset

```

MC_Reset_1 (
void MC_Reset_1([AXIS_REF|FB_AXIS_REF] AxisIn, BOOL Execute)
Tipo: MC_Reset, Reimposta tutti gli errori interni legati all'asse.

MC_Reset_1(Axis1,Execute_Reset);
Done_Reset := MC_Reset_1.Done;
Busy_Reset := MC_Reset_1.Busy;
Error_Reset := MC_Reset_1.Error;
ErrorID_Reset := MC_Reset_1.ErrorID;
    
```

Risultati

Nome	Valore logico	Valore fisico	Blocca	T
Execute_Reset	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOI
Done_Reset	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOI
Busy_Reset	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOI
Error_Reset	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOI
ErrorID_Reset	0	N/D	<input type="checkbox"/>	UINT

Vedere anche

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[Istruzioni di controllo del movimento](#) a pagina 423

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

MC_SetPosition (posizione impostazione del controllo movimento)

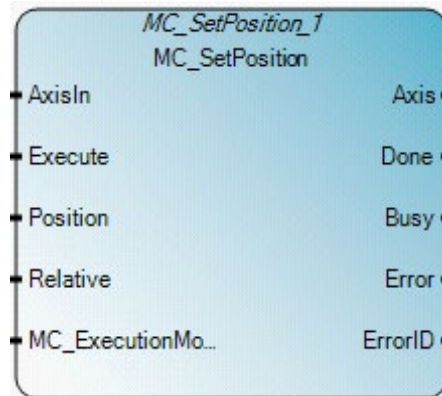
Sposta il sistema di coordinate di un asse modificando la relativa posizione effettiva con lo stesso valore, senza causare alcun movimento.

Dettagli operazione:

- MC_SetPosition può essere completato correttamente solo quando lo stato dell'asse è Fermo, Movimento continuo (MC_ExecutionMode = 0) o quando il movimento in corso viene completato e termina con lo stato Fermo (MC_ExecutionMode = 1).
- MC_SetPosition opera allo stesso modo di MC_Home quando HomingMode = MC_HOME_DIRECT (0x04), ad eccezione del fatto che il blocco funzione MC_Home imposta lo stato dell'asse su Azzerato.
- Quando MC_ExecutionMode = 0 (mcImmediately), l'esecuzione del blocco di funzione MC_SetPosition riporterà un errore in caso di movimento in corso non continuo dell'asse.
- Quando MC_ExecutionMode = 1 (mcQueued), l'impostazione della posizione corrente si verifica solo quando tutto il movimento precedente che è in corso si arresta. Ovvero, ciascun blocco funzione deve avere almeno una delle uscite Done, Aborted, o Error uguale a True.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.

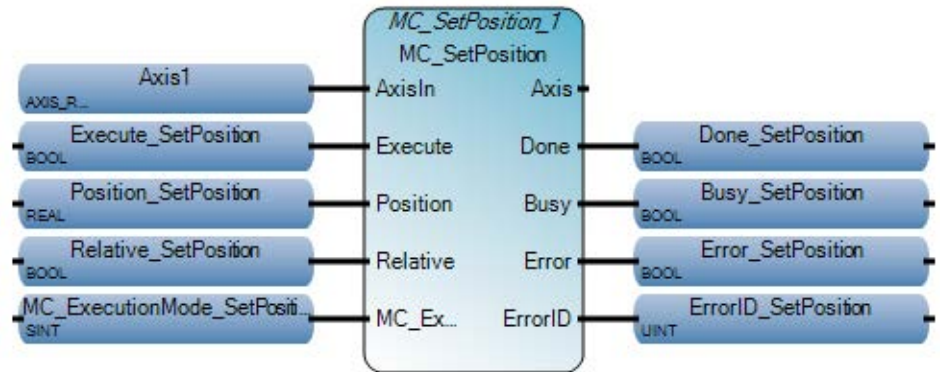


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

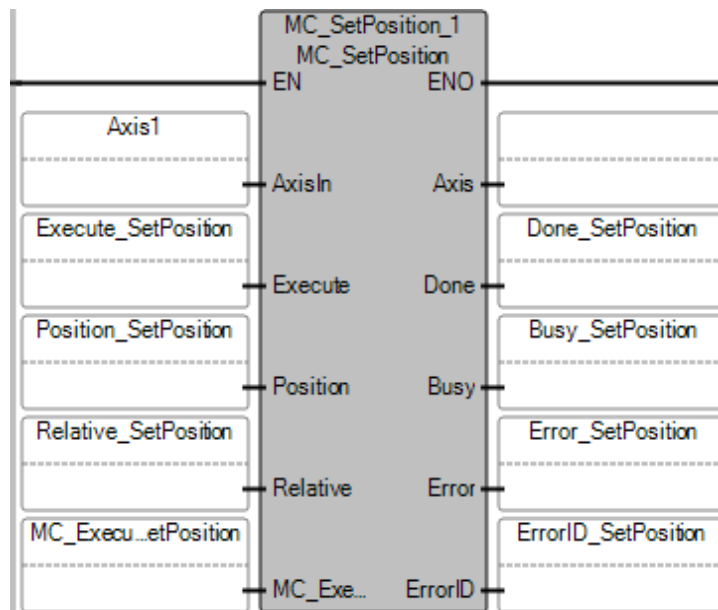
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_SetPosition corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare il tipo di dati AXIS_REF per definire AxisIn. Per FB_Axis (asse di feedback), utilizzare il tipo di dati FB_AXIS_REF per definire AxisIn.
Execute	Ingresso	BOOL	TRUE: avvia l'impostazione della posizione dell'asse.
Position	Ingresso	REAL	La posizione assoluta o la distanza relativa da impostare per l'asse.
Relative	Ingresso	BOOL	TRUE: imposta la distanza relativa per l'asse. FALSE: imposta la posizione assoluta per l'asse.
MC_ExecutionMode	Ingresso	SINT	I valori sono: <ul style="list-style-type: none"> • 0 (<i>mcImmediately</i>) - la funzionalità è immediatamente valida. • 1 (<i>mcQueued</i>) - la nuova funzionalità diventa valida quando: <ul style="list-style-type: none"> • tutti i comandi di movimento precedenti vengono impostati su uno dei seguenti parametri di uscita: Done, Aborted o Error. • l'asse non è in uno stato di movimento. Per (MC_ExecutionMode = 0), questo blocco funzione può essere completato con successo quando lo stato dell'asse è Disabilitato o Fermo. L'esecuzione di questo blocco funzione riporterà un errore in caso di movimento in corso non continuo con l'asse in questa modalità. Per (MC_ExecutionMode = 1), questo blocco di funzione può essere completato correttamente quando lo stato dell'asse è Disabilitato, Fermo o il movimento in corso può essere completato terminando con lo stato Fermo. Al momento gli altri valori di ingresso sono riservati e pertanto considerati come parametri non validi.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Finito	Uscita	BOOL	TRUE: la posizione ha un nuovo valore.
Busy	Uscita	BOOL	TRUE il blocco funzione non è finito.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempio di Programmazione a blocchi funzionali **MC_SetPosition**



Esempio di Diagramma ladder **MC_SetPosition**



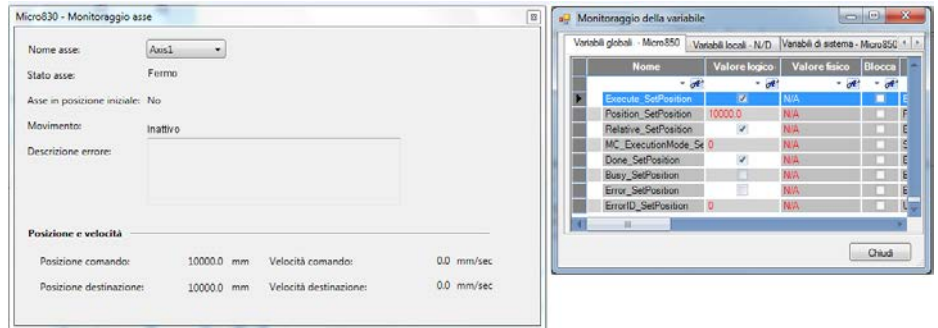
Esempio di Testo strutturato MC_SetPosition

```

MC_SetPosition_1
void MC_SetPosition_1(AXIS_REF AxisIn, BOOL Execute, REAL Position, BOOL Relative, SINT MC_ExecutionMode)
Type : MC_SetPosition, Shifts the coordinate system of an axis by manipulating the actual position of the axis with the same value without any movement caused

Position_SetPosition := 10000.0;
Relative_SetPosition := TRUE;
MC_ExecutionMode_SetPosition := 0;
MC_SetPosition_1(Axis1,Execute_SetPosition,Position_SetPosition,Relative_SetPosition,MC_ExecutionMode_SetPosition);
Done_SetPosition := MC_SetPosition_1.Done;
Busy_SetPosition := MC_SetPosition_1.Busy;
Error_SetPosition := MC_SetPosition_1.Error;
ErrorID_SetPosition := MC_SetPosition_1.ErrorID;
    
```

Risultati



Vedere anche

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[Istruzioni di controllo del movimento](#) a pagina 423

MC_Stop (arresto del controllo movimento)

Comanda un arresto di movimento controllato e trasferisce lo stato dell'asse su Arresto in corso. Tutti gli eventuali blocchi di funzione in esecuzione vengono interrotti. I comandi di movimento dei blocchi di funzione vengono tutti ignorati fino a che lo stato dell'asse non passa a StandStill.

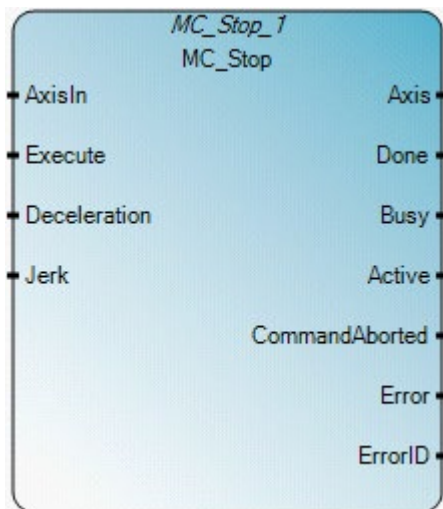
Dettagli operazione:

- Fin quando l'ingresso Execute è alto, l'asse rimane nello stato di Stopping. Mentre l'asse è nello stato Arresto, nessun altro blocco funzione di movimento può eseguire un movimento sullo stesso asse.
- Se Decelerazione è uguale a zero, i parametri MC_Stop vengono determinati dall'impostazione Arresto di emergenza della configurazione dell'asse, incluso tipo E-Stop, E-stop Deceleration e E-stop Jerk.

- Quando non sono presenti errori durante la sequenza di arresto, lo stato dell'asse esegue una transizione a Fermo dopo che il bit Completato viene IMPOSTATO e l'ingresso Esegue passa a False.
- Utilizzare MC_Stop per la funzione Arresto di emergenza o situazioni di eccezione. Utilizzare MC_Halt per il normale l'arresto del movimento.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.

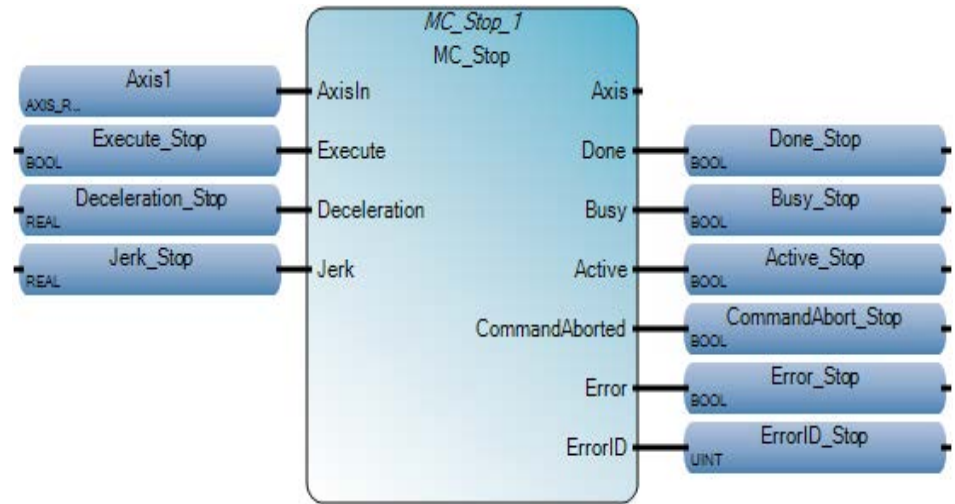


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

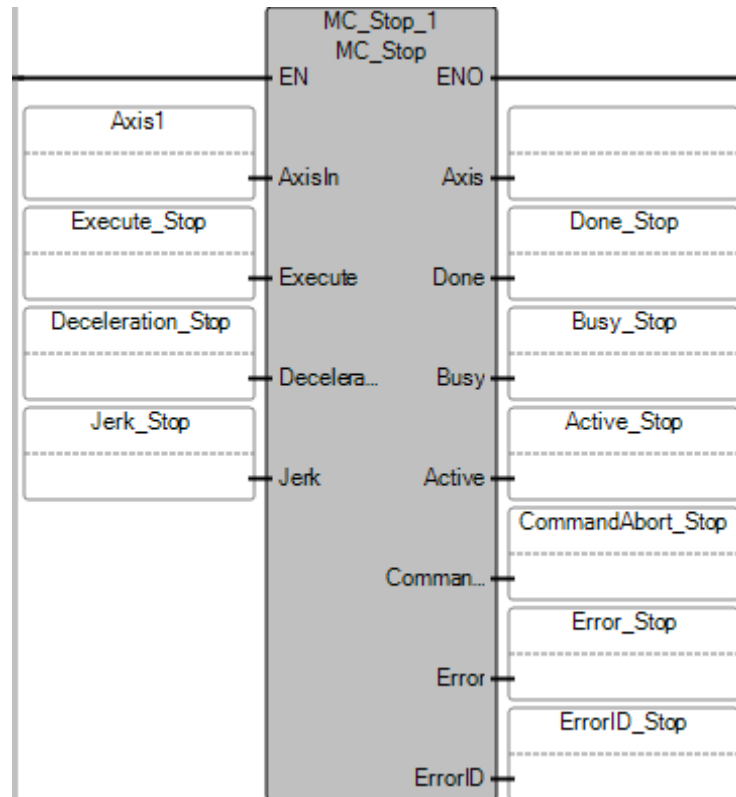
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_Stop corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF	Utilizzare il tipo di dati AXIS_REF per definire i parametri per AxisIn.
Execute	Ingresso	BOOL	TRUE: avvia l'azione fronte di salita. FALSE: non è in esecuzione.
Deceleration	Ingresso	REAL	Valore della decelerazione [u/s ²].
Jerk	Ingresso	REAL	Valore del jerk [u/s ³].
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Finito	Uscita	BOOL	TRUE: la velocità zero è stata raggiunta senza errori durante la sequenza di arresto.
Busy	Uscita	BOOL	TRUE il blocco funzione non è finito.
Attivo	Uscita	BOOL	TRUE: indica che il blocco funzione ha il controllo sull'asse.
CommandAborted	Uscita	BOOL	TRUE: il comando è stato interrotto dal blocco di funzione MC_Power(OFF) o da ErrorStop.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempio di Programmazione a blocchi funzionali MC_Stop



Esempio di Diagramma ladder MC_Stop



Esempio di Testo strutturato MC_Stop

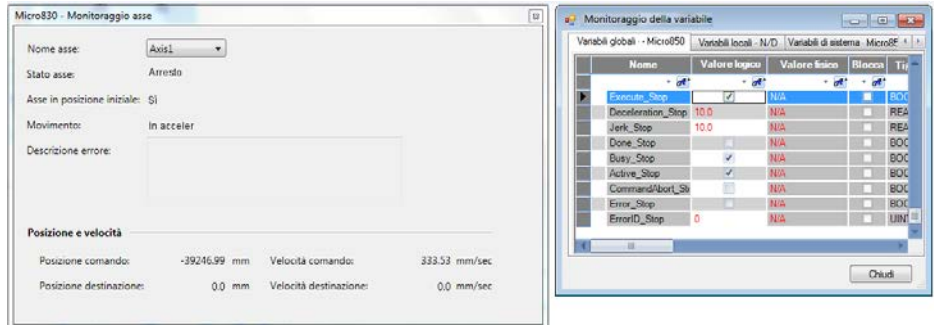
```

MC_Stop_1(
void MC_Stop_1(Axis_REF AxisIn, BOOL Execute, REAL Deceleration, REAL Jerk)
Tipo: MC_Stop, Determina un'interruzione del movimento controllata e il passaggio dell'asse allo stato Arresto in corso.

Deceleration_Stop := 10.0;
Jerk_Stop := 10.0;
MC_Stop_1(Axis1, Execute_Stop, Deceleration_Stop, Jerk_Stop);
Done_Stop := MC_Stop_1.Done;
Busy_Stop := MC_Stop_1.Busy;
Active_Stop := MC_Stop_1.Active;
CommandAbort_Stop := MC_Stop_1.CommandAborted;
Error_Stop := MC_Stop_1.Error;
ErrorID_Stop := MC_Stop_1.ErrorID;

```

Risultati



Vedere anche

[Tipo di dati AXIS_REF](#) a pagina 432

[Istruzioni di controllo del movimento](#) a pagina 423

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

MC_TouchProbe (sonda a contatto del controllo movimento)

Registra una posizione dell'asse in corrispondenza di un evento di attivazione.

Dettagli operazione:

- Se la direzione della finestra (prima posizione --> ultima posizione) è opposta a quella del movimento, la finestra della sonda a contatto non viene attivata.
- Se l'impostazione della finestra (FirstPosition o LastPosition) non è valida, il blocco di funzione MC_TouchProbe riporterà un errore.

- Se viene emessa una seconda istanza del blocco di funzione MC_TouchProbe sullo stesso asse e la prima istanza del blocco di funzione è in stato Busy, la seconda istanza del blocco di funzione riporterà un errore.
- Solo un'istanza del blocco di funzione MC_TouchProbe deve essere inviata a un asse.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

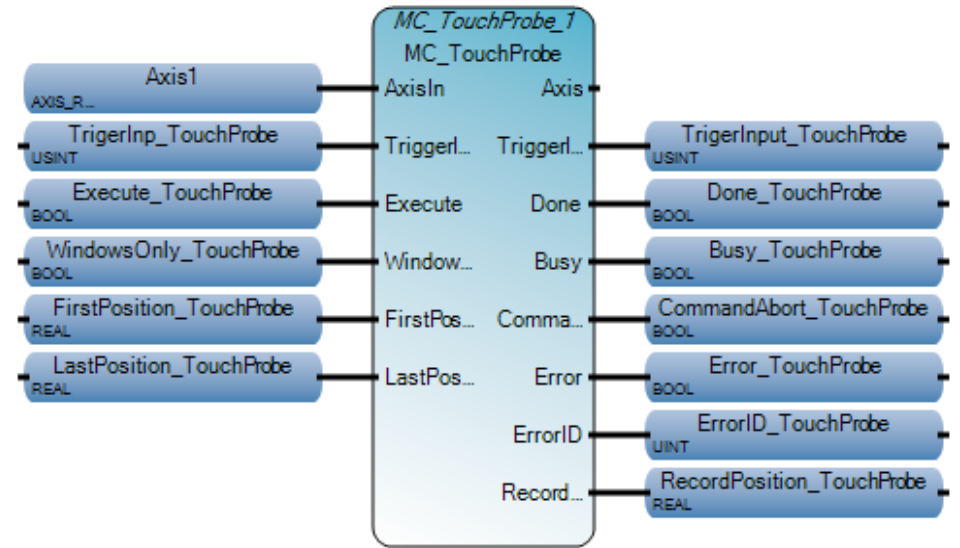
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_TouchProbe corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare il tipo di dati AXIS_REF per definire i parametri per AxisIn. Per un asse di feedback, utilizzare il tipo di dati FB_AXIS_REF per definire i parametri per AxisIn.
TriggerInp	Ingresso	USINT	Attualmente non supportato. Configurare trigger di ingresso nella configurazione Asse.
Execute	Ingresso	BOOL	TRUE: avvia la registrazione della sonda a contatto al fronte di salita. FALSE: non è in esecuzione.
WindowOnly	Ingresso	BOOL	TRUE: utilizza la finestra solo per accettare eventi trigger. La risoluzione del movimento è limitata all'intervallo Motore di movimento, configurato dall'utente. Per la funzionalità WindowOnly di TouchProbe, esiste un massimo ritardo del tempo di risposta che è uguale all'intervallo Motore di movimento, per l'attivazione sia di FirstPosition sia di LastPosition. Il ritardo massimo possibile nell'attivazione della posizione (sia FirstPosition sia LastPosition) può essere calcolato con la formula (Intervallo motore di movimento * velocità di movimento).
FirstPosition	Ingresso	REAL	Posizione iniziale della finestra in cui gli eventi trigger vengono accettati (in unità tecniche [u]). Valore incluso nella finestra.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
LastPosition	Ingresso	REAL	Posizione di arresto della finestra in cui gli eventi trigger vengono accettati (in unità tecniche [u]). Valore incluso nella finestra.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF FB_AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
TriggerInput	Uscita	USINT	Attualmente non supportato.
Finito	Uscita	BOOL	TRUE: l'evento trigger è stato registrato.
Busy	Uscita	BOOL	TRUE il blocco funzione non è finito.
CommandAborted	Uscita	BOOL	TRUE: il comando è stato annullato da MC_Power(OFF) o dal blocco di funzione Error Stop.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.
RecordedPosition	Uscita	REAL	Posizione in cui si è verificato l'evento trigger (in unità tecniche [u]) Motion è un movimento open-loop. Posizione dell'asse nell'istante in cui si verifica un evento di trigger. Se il controllo assi è un movimento a ciclo aperto, la posizione comandata (non la posizione corrente) nell'istante in cui si verifica un evento di trigger, presupponendo che non vi sia ritardo di movimento tra l'accensione e il motore.

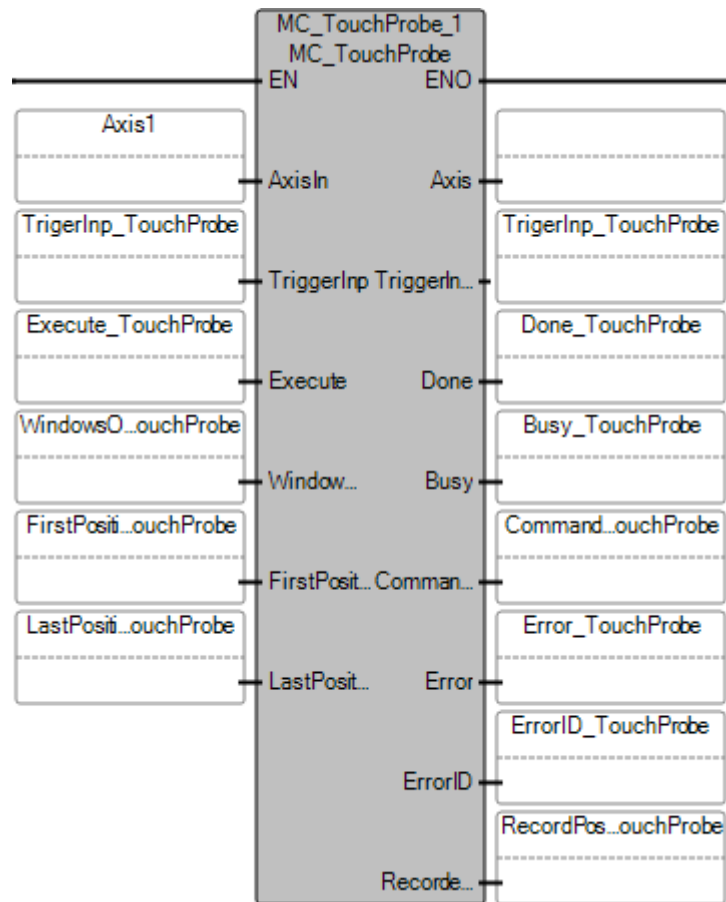
Ingresso/uscita movimento fisso

Segnali di movimento	PT00	PT01	PT02
Impulso PTO	Output_0	Output_1	Output2
PTO direction	Output_3	Output_4	Output_5
Interruttore limite inferiore (negativo)	Input_0	Input_4	Input_8
Interruttore limite superiore (positivo)	Input_1	Input_5	Input_9
Interruttore autoguida assoluto	Input_2	Input_6	Input_10
Interruttore ingresso sonda a contatto	Input_3	Input_7	Input_11

Esempio di Programmazione a blocchi funzionali MC_TouchProbe



Esempio di Diagramma ladder MC_TouchProbe



Esempio di Testo strutturato MC_TouchProbe

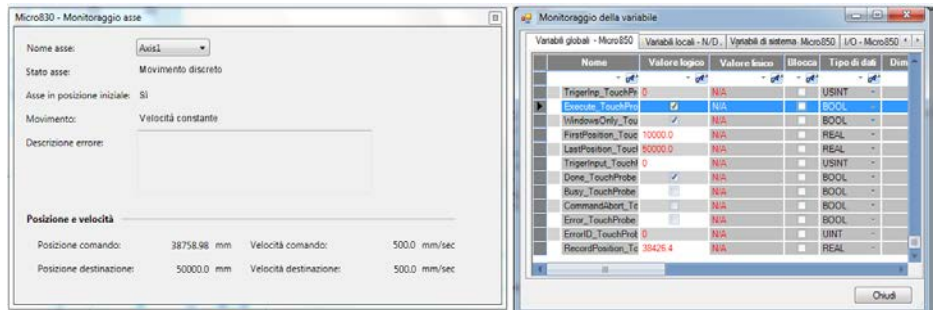
```

MC_TouchProbe_1(
void MC_TouchProbe_1(Axis1_REF AxisIn, USINT TriggerInp, BOOL Execute, BOOL WindowOnly, REAL FirstPosition, REAL LastPosition)
Type : MC_TouchProbe, Records an axis position at a trigger event

FirstPosition_TouchProbe := 10000.0;
LastPosition_TouchProbe := 50000.0;
MC_TouchProbe_1(Axis1, TrigerInp_TouchProbe, Execute_TouchProbe, WindowsOnly_TouchProbe,
FirstPosition_TouchProbe, LastPosition_TouchProbe);
Done_TouchProbe := MC_TouchProbe_1.Done;
Busy_TouchProbe := MC_TouchProbe_1.Busy;
Error_TouchProbe := MC_TouchProbe_1.Error;
ErrorID_TouchProbe := MC_TouchProbe_1.ErrorID;
RecordPosition_TouchProbe := MC_TouchProbe_1.RecordedPosition;

```

Risultati



Vedere anche

[Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

[ID errori del blocco funzione di controllo movimento](#) a pagina 430

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[Istruzioni di controllo del movimento](#) a pagina 423

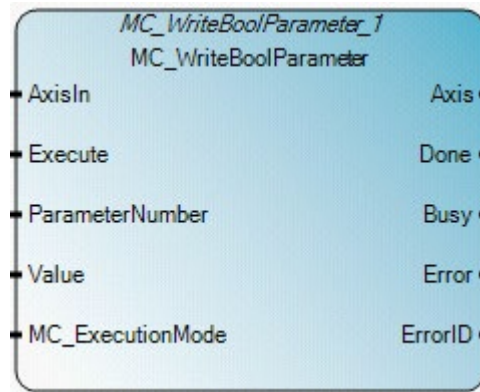
MC_WriteBoolParameter (parametro BOOL di scrittura del controllo movimento)

Modifica il valore di un parametro specifico del fornitore di tipo BOOL.

I parametri impostati dal blocco di funzione MC_WriteBoolParameter sono applicati solo temporaneamente nell'applicazione. Essi vengono sovrascritti dalle impostazioni permanenti, configurate dall'utente nella configurazione del controllo assi di Connected Components Workbench, quando il controllore viene commutato dalla modalità PRG a RUN, oppure quando attiva e disattiva l'alimentazione del controllore.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.

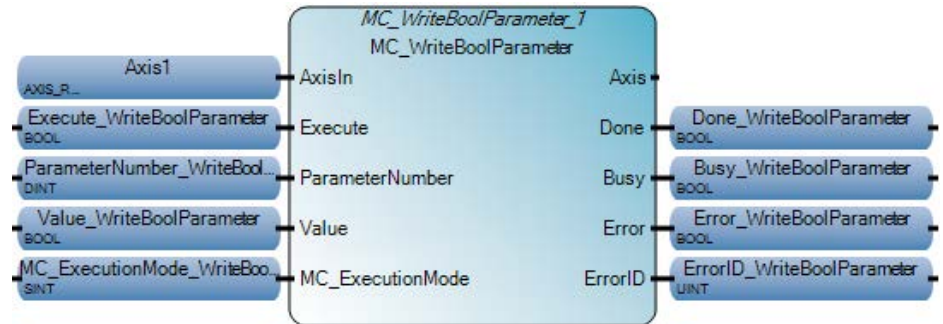


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

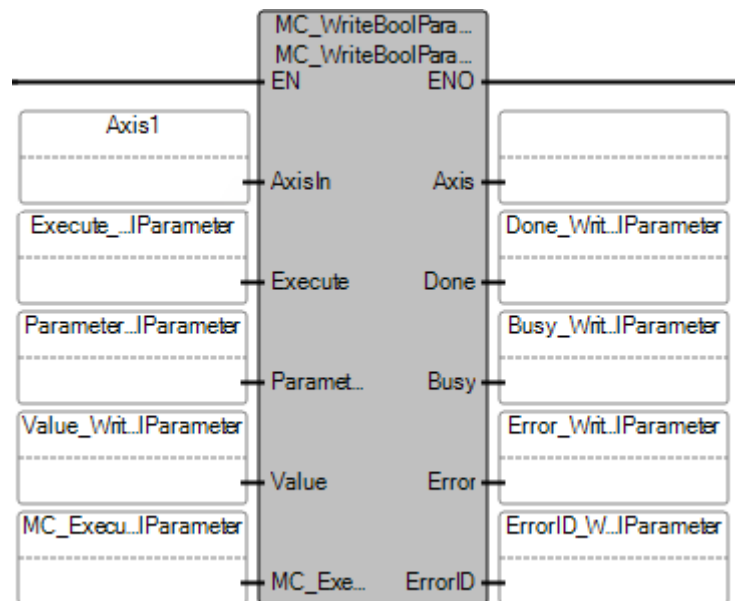
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_WriteBoolParameter corrente. FALSE: l'uscita del valore viene ripristinata su 0. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare il tipo di dati AXIS_REF per definire i parametri per AxisIn. Per un asse di feedback, utilizzare il tipo di dati FB_AXIS_REF per definire i parametri per AxisIn.
Execute	Ingresso	BOOL	TRUE: scrive il valore del parametro al fronte di salita.
ParameterNumber	Ingresso	DINT	Identificazione del parametro. Il numero del parametro definito nei dettagli relativi ai parametri del blocco funzione Controllo movimento.
Valore	Ingresso	BOOL	TRUE: il parametro specificato ha un nuovo valore.
MC_ExecutionMode	Ingresso	SINT	I valori sono: <ul style="list-style-type: none"> 0 (<i>mcImmediately</i>) - la funzionalità è immediatamente valida. 1 (<i>mcQueued</i>) - la nuova funzionalità diventa valida quando: <ul style="list-style-type: none"> tutti i comandi di movimento precedenti impostano uno dei seguenti parametri di uscita: Done, Aborted o Error l'asse non è in uno stato di movimento. quando (MC_ExecutionMode = 0) questo blocco di funzioni può essere completato con successo per tutti i parametri tranne Ciclo di lavoro (1005) solo quando lo stato dell'asse è Disattivato o Fermo. Quando (MC_ExecutionMode = 0), per Parametro ciclo di carico (1005), questo FB può essere completato correttamente tranne che l'asse si trova in uno stato Azzeramento o ErrorStop. Per (MC_ExecutionMode = 1), questo blocco di funzione può essere completato con successo solo quando lo stato dell'asse è Disabled, StandStill o il movimento in corso può essere completato terminando con lo stato StandStill Al momento gli altri valori di ingresso sono riservati e pertanto considerati come parametri non validi.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.
Finito	Uscita	BOOL	TRUE: il parametro è stato scritto correttamente.
Busy	Uscita	BOOL	TRUE il blocco funzione non è finito.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempio di Programmazione a blocchi funzionali MC_WriteBoolParameter



Esempio di Diagramma ladder MC_WriteBoolParameter



Esempio di Testo strutturato MC_WriteBoolParameter

```

MC_WriteBoolParameter_1(
void MC_WriteBoolParameter_1(AXIS_REF[FB_AXIS_REF] AxisIn, BOOL Execute, DINT ParameterNumber, BOOL Value, SINT MC_ExecutionMode)
Tipo: MC_WriteBoolParameter, Modifica il valore di un parametro BOOL specifico del movimento.

ParameterNumber_WriteBoolParameter := 5;
MC_WriteBoolParameter_1(Axis1, Execute_WriteBoolParameter, ParameterNumber_WriteBoolParameter,
Value_WriteBoolParameter, MC_ExecutionMode_WriteBoolParameter);
Done_WriteBoolParameter := MC_WriteBoolParameter_1.Done;
Busy_WriteBoolParameter := MC_WriteBoolParameter_1.Busy;
Error_WriteBoolParameter := MC_WriteBoolParameter_1.Error;
ErrorID_WriteBoolParameter := MC_WriteBoolParameter_1.ErrorID;
    
```

Risultati

The screenshot shows a window titled "Monitoraggio della variabile" with a table of variables. The table has columns for Name, Valore logico, Valore fisico, Blocca, and Tipo. The variables listed are:

Nome	Valore logico	Valore fisico	Blocca	Tipo
Execute_WriteBoolParameter	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
ParameterNumber_WriteBoolParameter	5	N/D	<input type="checkbox"/>	DINT
Value_WriteBoolParameter	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
MC_ExecutionMode_WriteBoolParameter	0	N/D	<input type="checkbox"/>	SINT
Done_WriteBoolParameter	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
Busy_WriteBoolParameter	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
Error_WriteBoolParameter	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
ErrorID_WriteBoolParameter	0	N/D	<input type="checkbox"/>	UINT

Vedere anche

- [Istruzioni di controllo del movimento](#) a pagina 423
- [Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426
- [ID errori del blocco funzione di controllo movimento](#) a pagina 430
- [Regole generali per i blocchi funzione di controllo movimento](#) a pagina 424

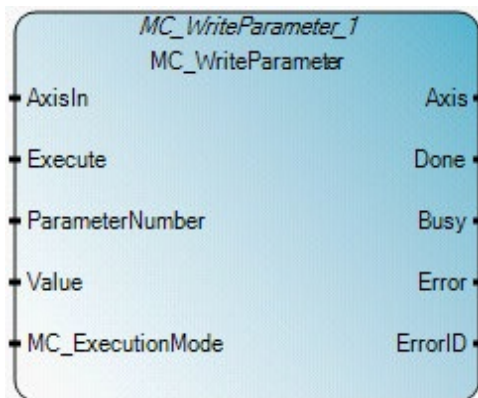
MC_WriteParameter (parametro di scrittura del controllo movimento)

Modifica del valore di un parametro specifico del fornitore di tipo REAL.

I parametri impostati dal blocco di funzione MC_WriteParameter sono applicati solo temporaneamente nell'applicazione. Essi vengono sovrascritti dalle impostazioni permanenti, configurate dall'utente nella configurazione del controllo assi di Connected Components Workbench, quando il controllore viene commutato da PRG a RUN, oppure quando attiva e disattiva l'alimentazione del controllore.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro830, Micro850 e Micro870 che supportano il controllo del movimento.

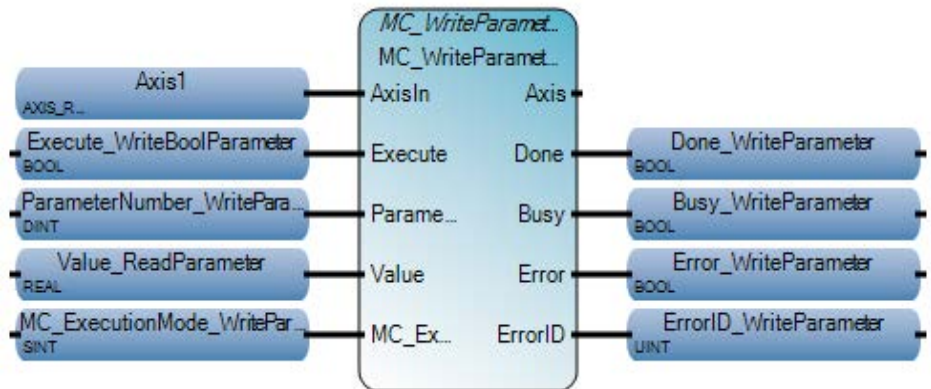


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

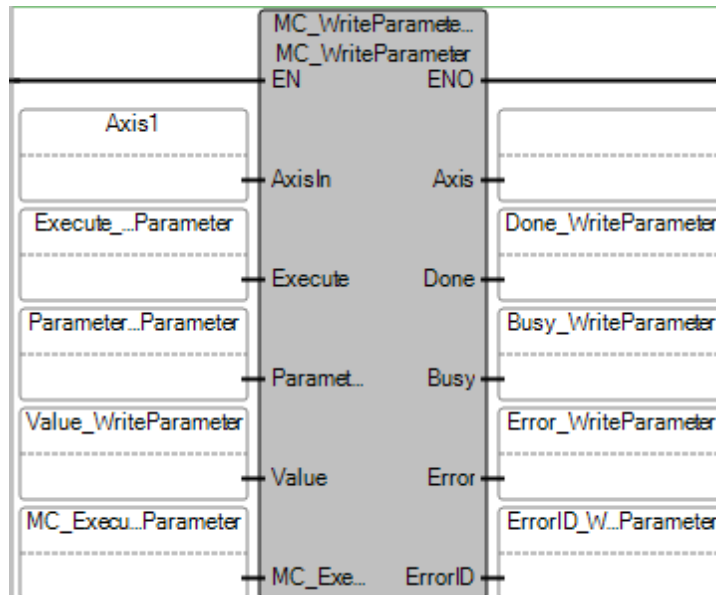
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il calcolo MC_WriteParameter corrente. FALSE: il calcolo non avviene. Applicabile solo alle programmazioni in Ladder Diagram.
AxisIn	Ingresso	AXIS_REF FB_AXIS_REF	Utilizzare il tipo di dati AXIS_REF per definire i parametri per AxisIn. Per un asse di feedback, utilizzare il tipo di dati FB_AXIS_REF per definire i parametri per AxisIn.
Execute	Ingresso	BOOL	TRUE: scrive il valore del parametro al fronte di salita.
ParameterNumber	Ingresso	DINT	Identificazione del parametro. Il numero del parametro definito nei dettagli relativi ai parametri del blocco funzione Controllo movimento.
Valore	Ingresso	REAL	Nuovo valore per il parametro specificato.
MC_ExecutionMode	Ingresso	SINT	I valori sono: <ul style="list-style-type: none"> • 0 (<i>mcImmediately</i>) - la funzionalità è immediatamente valida. • 1 (<i>mcQueued</i>) - la nuova funzionalità diventa valida quando: <ul style="list-style-type: none"> • tutti i comandi di movimento precedenti impostano uno dei seguenti parametri di uscita: Done, Aborted o Error • l'asse non è in uno stato di movimento. • implica che il parametro di uscita Busy è impostato su FALSE. Quando (MC_ExecutionMode = 0), questo blocco funzionale può essere completato con successo per tutti i parametri tranne Ciclo di lavoro (1005) solo quando lo stato dell'asse è Disattivato o Fermo. Quando (MC_ExecutionMode = 0), per Parametro ciclo di carico (1005), questo FB può essere completato correttamente tranne che l'asse si trova in uno stato Azzeramento o ErrorStop. Per (MC_ExecutionMode = 1), questo blocco di funzione può essere completato con successo solo quando lo stato dell'asse è Disabled, StandStill o il movimento in corso può essere completato terminando con lo stato StandStill Al momento gli altri valori di ingresso sono riservati e pertanto considerati come parametri non validi.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Asse	Uscita	AXIS_REF	L'uscita asse nelle programmazioni in Ladder Diagram è di sola lettura.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Done	Uscita	BOOL	TRUE: il parametro è stato scritto correttamente.
Busy	Uscita	BOOL	TRUE: indica che il blocco funzione ha il controllo dell'asse.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore.
ErrorID	Uscita	UINT	Numero univoco che identifica l'errore. Gli errori per questa istruzione sono definiti negli ID errori del blocco funzione Controllo del movimento.

Esempio di Programmazione a blocchi funzionali MC_WriteParameter



Esempio di Diagramma ladder MC_WriteParameter



Esempio di Testo strutturato MC_WriteParameter

```

MC_WriteParameter_1(
void MC_WriteParameter_1(AXIS_REF AxisIn, BOOL Execute, DINT ParameterNumber, REAL Value, SINT MC_ExecutionMode,
Type : MC_WriteParameter, Modifies the value of a motion specific REAL parameter

ParameterNumber_WriteParameter := 2;
Value_WriteParameter := 90000.0;
MC_WriteParameter_1(Axis1,Execute_WriteParameter,ParameterNumber_WriteParameter,
Value_WriteParameter,MC_ExecutionMode_WriteParameter);
Done_WriteParameter := MC_WriteParameter_1.Done;
Busy_WriteParameter := MC_WriteParameter_1.Busy;
Error_WriteParameter := MC_WriteParameter_1.Error;
ErrorID_WriteParameter := MC_WriteParameter_1.ErrorID;

```

Risultati

Nome	Valore logico	Valore fisico	Blocca	T
Execute_WriteParameter	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOO
ParameterNumber_WriteParameter	2	N/D	<input type="checkbox"/>	DINT
Value_WriteParameter	90000.0	N/D	<input type="checkbox"/>	REAL
MC_ExecutionMode_WriteParameter	0	N/D	<input type="checkbox"/>	SINT
Done_WriteParameter	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOO
Busy_WriteParameter	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOO
Error_WriteParameter	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOO
ErrorID_WriteParameter	0	N/D	<input type="checkbox"/>	UINT

Vedere anche

[Tipo di dati AXIS_REF](#) a pagina 432

[Tipo di dati FB_AXIS_REF](#) a pagina 433

[Istruzioni di controllo del movimento](#) a pagina 423

[Dettagli relativi ai parametri del blocco funzione per il controllo del movimento](#) a pagina 426

[Regole generali per i blocchi funzione Controllo movimento](#) a pagina 424

Istruzioni di controllo dei processi

Usare le istruzioni di controllo dei processi per monitorare e mantenere i cicli di processo per le quantità, quali pressione, temperatura, portata e livello dei fluidi. I controlli dei processi regolano il percorso inviando un segnale di uscita alla valvola di controllo.

Istruzione	Descrizione
DERIVATE a pagina 503	Differenzia un valore reale su una durata del ciclo definito.
FFL a pagina 505	Carica i dati a 8 bit, 16 bit, 32 bit o 64 bit in un vettore creato dall'utente denominato stack FIFO.
FFU a pagina 515	Scarica i dati a 8 bit, 16 bit, 32 bit o 64 bit da un vettore creato dall'utente denominato stack FIFO. I dati vengono scaricati nello stesso ordine in cui sono caricati usando l'istruzione FFL.
HYSTER a pagina 520	Esegue una isteresi booleana sulla differenza tra dati reali.
INTEGRAL a pagina 522	Integra un valore real durante la durata del ciclo definito.
LFL(LIFO load) a pagina 528	Carica i dati a 8 bit, 16 bit, 32 bit o 64 bit in un vettore creato dall'utente denominato stack LIFO.
LFU(LIFO unload) a pagina 530	Scarica i dati a 8 bit, 16 bit, 32 bit o 64 bit da un vettore creato dall'utente denominato stack LIFO. I dati vengono scaricati nello stesso ordine in cui sono caricati usando l'istruzione LFL.
PWM a pagina 533	Attiva o disattiva l'uscita della modulazione dell'ampiezza di un impulso (PWM) per un canale PWM configurato.
SCALER a pagina 535	Scala il valore di ingresso in base all'intervallo di uscita.
STACKINT a pagina 537	Gestisce uno stack di valori interi.
LIMIT a pagina 549	riduce i valori interi a un intervallo definito.
TND a pagina 547	Arresta il ciclo di scansione del programma utente in corso.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

DERIVATE

Differenziazione di un valore reale su una durata del ciclo definito.

Dettagli operazione:

- Se il valore del parametro CYCLE è inferiore alla temporizzazione del ciclo di esecuzione del dispositivo, il periodo di campionamento viene forzato a questa temporizzazione del ciclo.
- La derivata viene eseguita su una base di tempo espresso in millisecondi. Per esempio, la derivata di un ingresso di 1.000 che cambia in 2.000 in un intervallo di tempo di 1 secondo darà un valore pari a 1. Per convertire l'uscita dell'istruzione in secondi, moltiplicare il valore di uscita per 1000.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

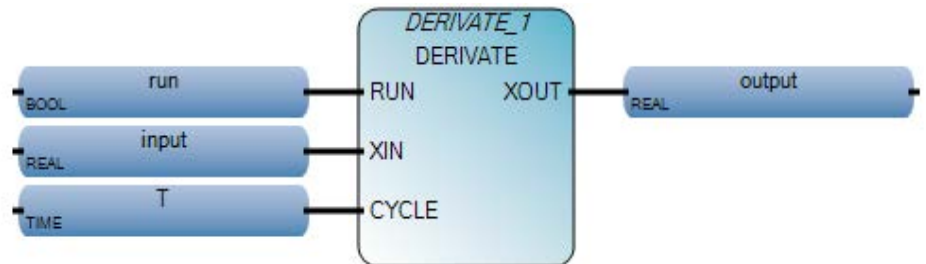
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



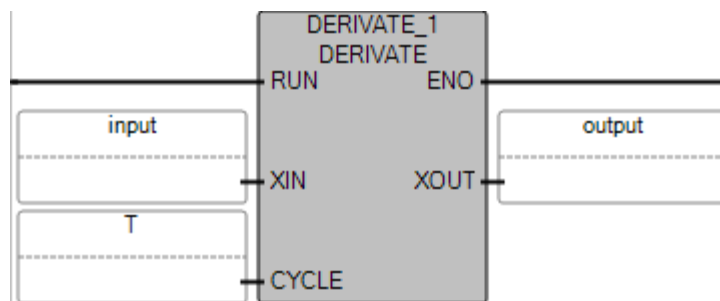
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
RUN	Ingresso	BOOL	Indica la modalità operativa dell'istruzione. TRUE: normale (esegue il calcolo) FALSE: reset
XIN	Ingresso	REAL	Definisce il valore su cui eseguire il calcolo di derivazione. Il valore deve essere un valore REAL.
CYCLE	Ingresso	TIME	Definisce il periodo di campionamento in cui raccogliere i valori. L'intervallo di valori del periodo di tempo possibile varia da 0 ms a 49 g 17 h 2 m 47 s 294 ms.
XOUT	Uscita	REAL	Uscita derivata.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempi di programmazione a blocchi funzionali DERIVATE



Esempio di diagramma ladder DERIVATE



Esempio di testo strutturato DERIVATE

```
DERIVATE_1 {
void DERIVATE_1(BOOL RUN, REAL XIN, TIME CYCLE)
Tipo: DERIVATE, Differenziazione in base alla durata
```

```
1 DERIVATE_1(run, input, T);
2 output := DERIVATE_1.XOUT;
```

(* Equivalenza ST: DERIVATE1 è un'istanza del blocco DERIVATE *)

```
DERIVATE1(manual_mode, sensor_value, t#100ms);
derivated_value := DERIVATE1.XOUT;
```

Vedere anche

[Istruzioni di controllo dei processi](#) a pagina 503

FFL (caricamento FIFO)

Carica i dati a 8 bit, 16 bit, 32 bit o 64 bit in un vettore creato dall'utente denominato stack FIFO.

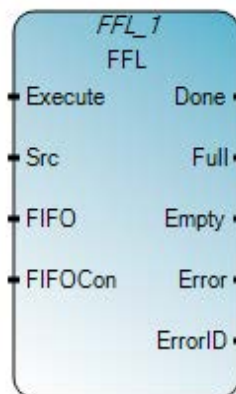
Dettagli operazione:

- Istruzione FFL: da modalità di non esecuzione in modalità di esecuzione
- Quando Esegue passa da FALSE a TRUE:
 - Le condizioni di errore vengono verificate.
 - I contenuti di Src vengono caricati all'interno dello stack FIFO nella posizione disponibile e Posizione incrementa di 1 se Posizione è minore o uguale a zero e minore della Lunghezza.
 - Il bit Pieno è impostato su TRUE se la Lunghezza è uguale alla Posizione. Il bit Pieno è FALSE se la Posizione è minore o uguale a zero e minore della Lunghezza.
 - Il bit Completato viene impostato quando l'istruzione viene eseguita correttamente.
- Quando Esegue passa da FALSE a TRUE:
 - I bit Errore, Completato e ErrorID sono impostati su FALSE.
 - Il bit Vuoto è impostato su TRUE se Posizione è uguale a zero.
 - Il bit Pieno è impostato su TRUE se la Lunghezza è uguale alla Posizione. Il bit Pieno è FALSE se la Posizione è minore o uguale a zero e minore della Lunghezza.
 - Le condizioni di errore FFL non sono verificate.
- Quando Esegue passa da TRUE a TRUE:
 - Non viene eseguita alcuna operazione di caricamento.
 - Il bit Vuoto è impostato su TRUE se Posizione è uguale a zero.
 - Il bit Pieno è impostato su FALSE se la Posizione è minore o uguale a zero e minore della Lunghezza. Il bit Pieno è

- impostato su TRUE se la Lunghezza è uguale alla Posizione.
- Le condizioni di errore FFL non sono verificate.
- Quando Esegue passa da FALSE a FALSE:
 - I bit Errore, Completato e ErrorID sono impostati su FALSE.
 - I bit Pieno e Vuoto bit conservano i valori dallo stato di esecuzione precedente.
- Il bit Vuoto è impostato su TRUE se Posizione è uguale a zero.
 - Il bit Pieno è impostato su TRUE se la Lunghezza è uguale alla Posizione. Il bit Pieno è FALSE se la Posizione è minore o uguale a zero e minore della Lunghezza.
 - Le condizioni di errore FLL non sono verificate.
- Istruzione FFL: dalla modalità in esecuzione alla modalità non in esecuzione:
 - I bit Errore, ErrorID, Completato, Vuoto e Pieno conservano lo stato modalità in esecuzione.
- Per creare un singolo elemento per il parametro FIFO:
 - Non vettore:
 - L'indirizzo basato sulla variabile come Fifo1 è consentito per FIFO.
 - La lunghezza deve essere configurata come 1.
 - Vettore
 - L'indirizzo basato sulla variabile come Fifo1 o Fifo1[0] è consentito per FIFO.
 - La lunghezza deve essere configurata come 1.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
-----------	-------------------	--------------	-------------

Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: se viene rilevato il fronte di salita, avviare l'operazione FFL. FALSE: fronte di salita non rilevato.
Src	Ingresso	ANY_ELEMENTARY	L'operando Src è l'indirizzo del valore utilizzato per riempire la posizione dello stack FIFO attualmente disponibile. Tipi di dati elementari supportati per Scr: <ul style="list-style-type: none"> • DWORD, REAL, TIME, DATE, LWORD, ULINT, LINT, LREAL, BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT. • La stringa non è supportata. • Gli elementi del vettore come Vettore[1] o Vettore[indice] sono supportati.
FIFO	Ingresso	ANY_ELEMENTARY	L'indirizzo iniziale dello stack. FIFO deve essere configurato come per le istruzioni FFL e FFU. Tipi di dati elementari supportati per FIFO: DWORD, REAL, TIME, DATE, LWORD, ULINT, LINT, LREAL, BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT. <ul style="list-style-type: none"> • La stringa non è supportata. • È supportato solo il vettore unidimensionale per FIFO.
FIFOCon	Ingresso	FF_LF_CON	Controllo e configurazione FIFO. La stessa configurazione deve essere configurata per le istruzioni FFL e FFU. Per configurare posizione e lunghezza, utilizzare il tipo di dati FF_LF_CON.
Done	Uscita	BOOL	Indica se l'operazione FFU è stata completata. TRUE: operazione completata correttamente. FALSE: l'operazione ha rilevato una condizione di errore o l'istruzione FFU non è in esecuzione.
Vuoto	Uscita	BOOL	Indica se lo stack FIFO è vuoto. TRUE: quando la posizione è uguale a 0. FALSE: quando la posizione non è uguale a 0.
Completo	Uscita	BOOL	Indica se lo stack FIFO è pieno. TRUE: quando lunghezza è uguale a posizione. FALSE: quando la posizione è maggiore o uguale a zero e minore della lunghezza.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE: l'operazione ha riscontrato un errore. FALSE: l'operazione è stata completata o l'istruzione FFU non è in esecuzione.
ErrorID	Uscita	USINT	Numero univoco che identifica l'errore. Gli errori sono definiti nei codici di errore FFL.

Tipo di dati FF_LF_CON

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati FF_LF_CON.

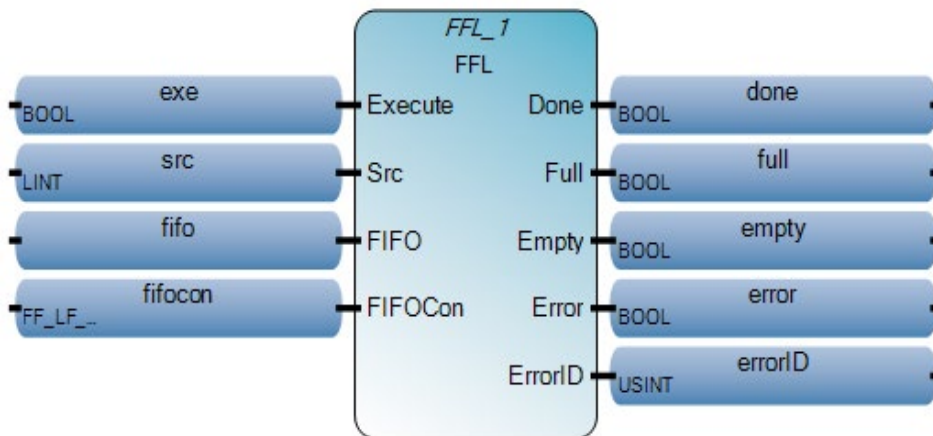
Parametri	Tipo di dati	Descrizione
Lunghezza	UINT	Numero di elementi utilizzati per l'operazione FIFO. Il limite massimo è 1024.
Position	USINT	Determina la posizione disponibile successiva in FIFO per la rimozione o l'immissione di Src. La posizione è l'offset del vettore. Esempio 1: <ul style="list-style-type: none"> • Vettore configurato dall'utente, arr[0..5]. La posizione iniziale è configurata come 1. I dati sono inseriti in arr[1] e la posizione aumenta di + 1. Esempio 2: <ul style="list-style-type: none"> • Vettore configurato dall'utente come arr[1..5]. La posizione iniziale è configurata come 1. I dati sono inseriti in arr[2] e la posizione aumenta di + 1.

Codici di errore FFL

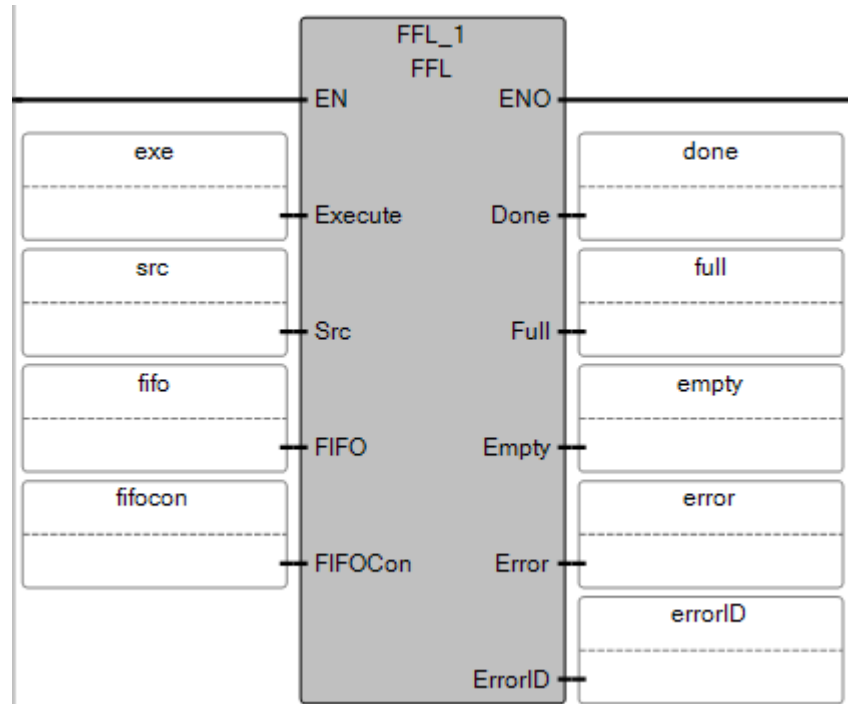
Utilizzare questa tabella per determinare i codici di errore FFL e FFU e le descrizioni.

Codice errore	Descrizione errore
0	Nessun errore.
1	Il tipo di dati Src FFL non è supportato.
2	Il tipo di dati Dest FFU non è supportato.
3	Il tipo di dati FIFO non è supportato.
4	I tipi di dati Src e Dest non corrispondono al tipo di dati FIFO. Azione correttiva: Il tipo di dati del parametro Src FFL e del parametro Dest FFU devono corrispondere al tipo di dati del vettore FIFO.
5	FIFO: la dimensione del vettore non è supportata. Azione correttiva: FIFO supporta solo vettori unidimensionali.
6	La lunghezza del controllo FIFOCon supera la dimensione del vettore FIFO. Azione correttiva: La lunghezza del controllo FIFOCon non supera la dimensione del vettore FIFO.
7	La lunghezza FIFOCon supera la lunghezza massima.
8	La lunghezza FIFOCon è zero.
9	La posizione FIFOCon supera la lunghezza FIFOCon.
10	La posizione e la lunghezza del controllo FFL sono uguali.
11	La posizione del controllo FFU è zero.
12	La dimensione del vettore FFL o FFU non è supportata. Azione correttiva: FFL e FFU supportano solo vettori unidimensionali.
13	FFL o FFU DestOffset supera la dimensione vettore Dest.

Esempio di Programmazione a blocchi funzionali FFL



Esempio di Diagramma ladder FFL



Esempio di Testo strutturato FFL

```

1 | FFL_1(exe, src, fifo, fifoon);
2 | done := FFL_1.Done;
3 | full := FFL_1.Full;
4 | empty := FFL_1.Empty;
5 | error := FFL_1.Error;
6 | errorID := FFL_1.ErrorID;

```

```

FFL_1(
void FFL_1(BOOL Execute, ANY_ELEMENTARY Src, ANY_ELEMENTARY[1..1] FIFO, FF_LF_CON FIFOCon)
Tipo: FFL, Carico FIFO

```

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	D
src	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
full	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifocon	<input type="checkbox"/>	FF_LF_CON	
fifocon.Length	10	N/D		<input type="checkbox"/>	UINT	
fifocon.Position	1	N/D		<input type="checkbox"/>	UINT	
fifo	<input type="checkbox"/>	BOOL	[1..
<input checked="" type="checkbox"/> fifo[1]	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[2]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[3]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[4]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[5]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[6]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[7]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[8]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[9]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[10]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
+ FFL_1	<input type="checkbox"/>	FFL	
exe	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
errorID	0	N/D		<input type="checkbox"/>	USINT	
error	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
empty	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
done	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	

Vedere anche

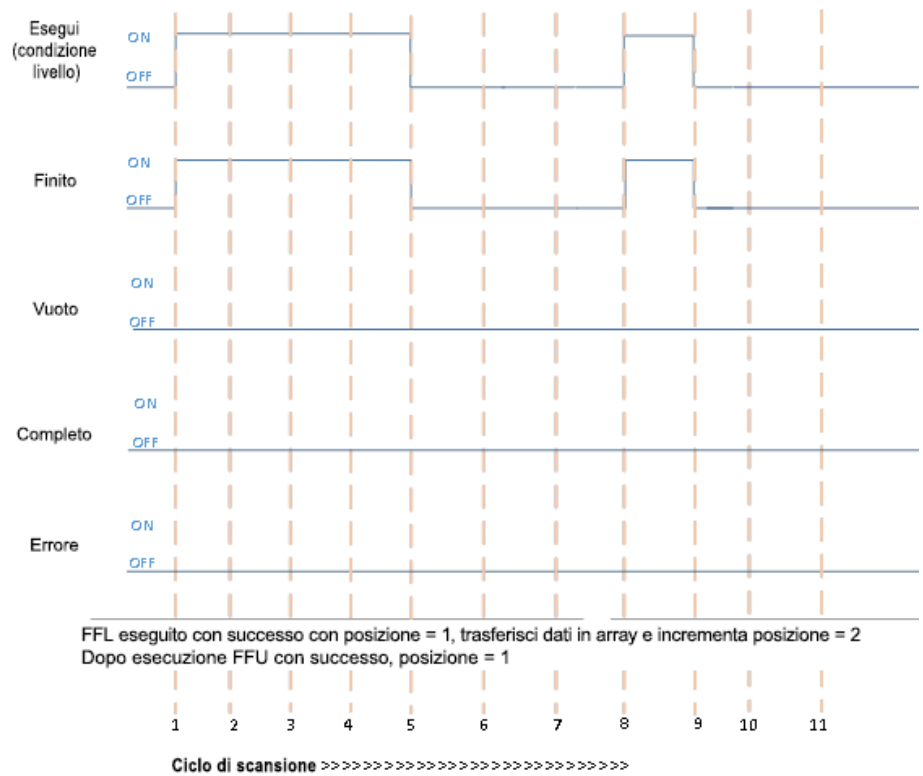
[Diagrammi di temporizzazione istruzione FFL e FFU](#) a pagina 510

[Istruzioni di controllo dei processi](#) a pagina 503

Diagrammi di temporizzazione istruzione FFL e FFU

Gli esempi di diagramma di temporizzazione seguenti descrivono gli scenari di esecuzione per le istruzioni FFL (carico FIFO) e FFU (scarica FIFO).

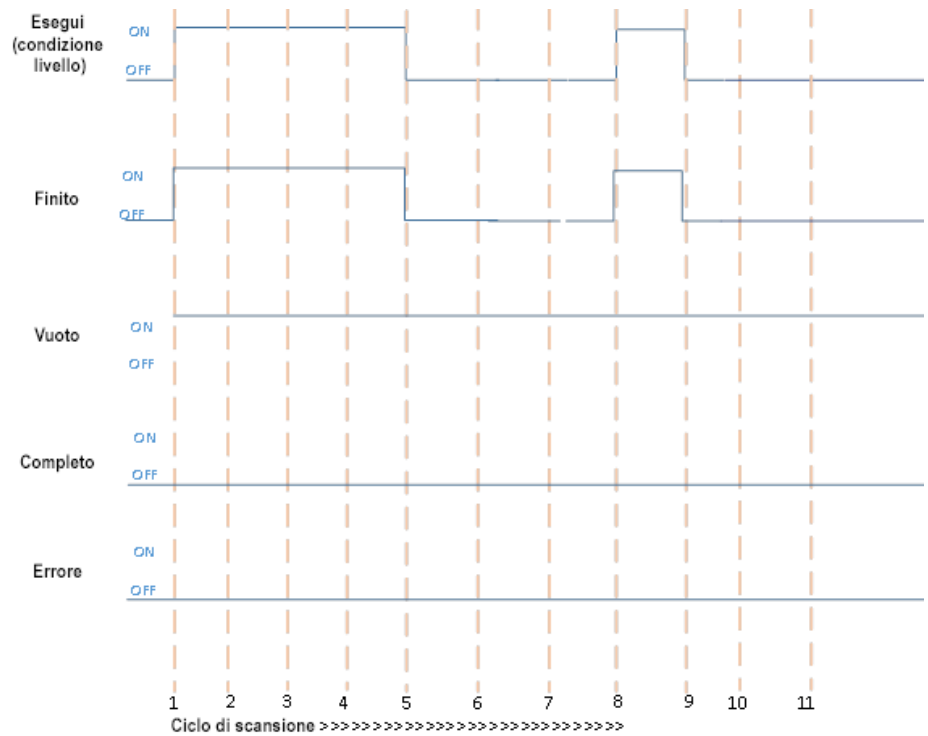
Il completamento dell'esecuzione FFL seguita dal completamento dell'esecuzione FFU



Utilizzare questa tabella per determinare i valori del parametro per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Esegui è TRUE. • Dati di carico (push) allo stack FIFO. • Il bit di uscita Completato è TRUE.
2, 3, 4	Nessuna modifica nella condizione ramo.
5	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Esegui è FALSE. • Il bit di uscita Completato è FALSE.
6, 7	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit Esegui è FALSE. • Il bit di uscita Completato è FALSE.
8	Il ramo passa a TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Esegui è TRUE. • Scaricare i dati dallo stack FIFO. • Il bit di uscita Completato è TRUE.
9	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit Esegui è FALSE. • Il bit di uscita Completato è FALSE.
10, 11	Nessuna modifica nella condizione ramo.

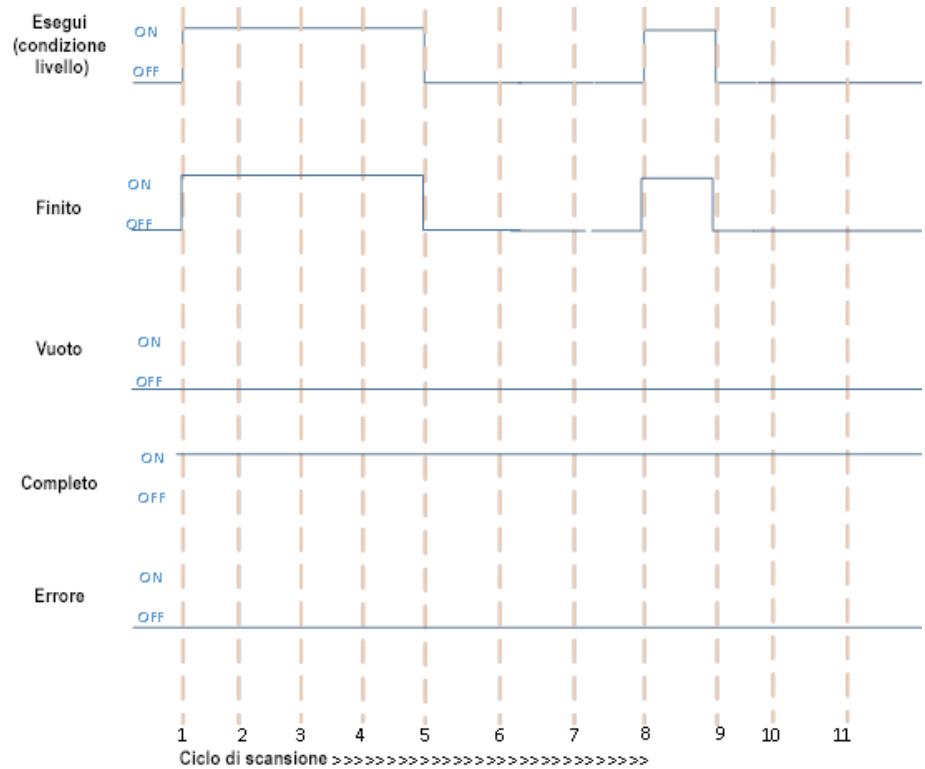
Completamento dell'esecuzione quando il bit vuoto è TRUE



Utilizzare questa tabella per determinare i valori del parametro per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Esegue è TRUE. Avvio esecuzione. • La Posizione è zero. Il bit Vuoto è TRUE. • Il bit di uscita Completato è TRUE.
2, 3, 4	Nessuna modifica nella condizione ramo.
5	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Esegue è FALSE. • Il bit Vuoto è TRUE. • Il bit di uscita Completato è FALSE.
6, 7	Nessuna modifica nella condizione ramo.
8	Il ramo passa a TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Esegue è TRUE. Avvio esecuzione. • Il bit Vuoto è TRUE. • Il bit di uscita Completato è TRUE.
9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Esegue è FALSE. • Il bit Vuoto è TRUE. • Il bit di uscita Completato è FALSE.
10, 11	Nessuna modifica nella condizione ramo.

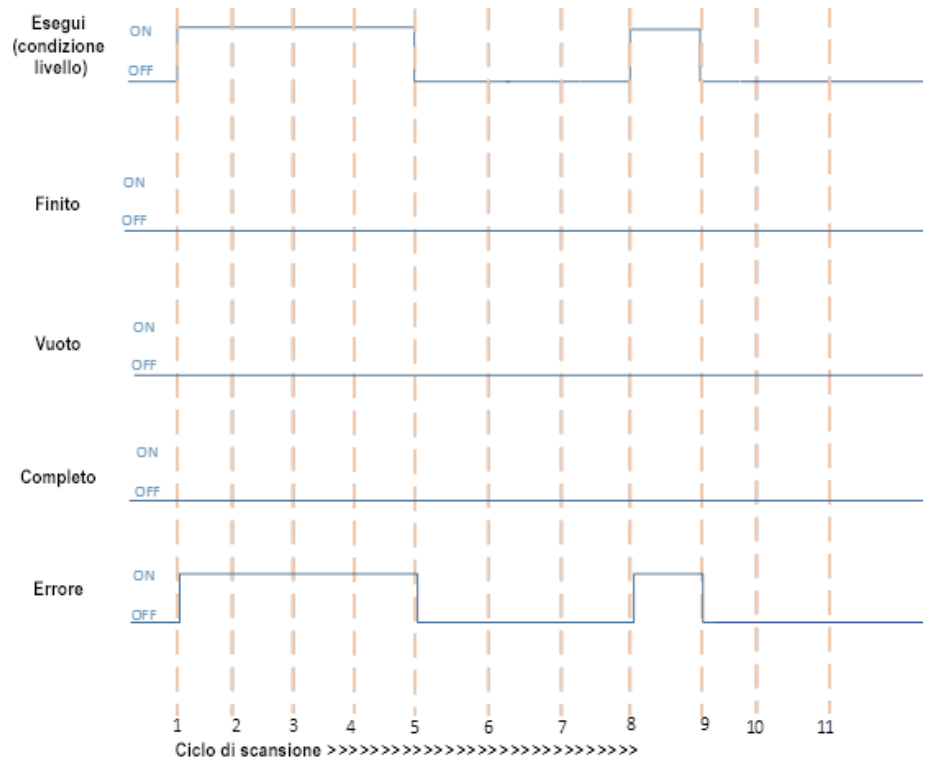
Completamento dell'esecuzione quando il bit vuoto è TRUE



Utilizzare questa tabella per determinare i valori del parametro per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Esegue è TRUE. Avvio esecuzione. • La Posizione è uguale alla Lunghezza, il bit Pieno è TRUE. • Il bit di uscita Completato è TRUE.
2, 3, 4	Nessuna modifica nella condizione ramo.
5	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Esegue è FALSE. • Il bit Pieno è TRUE. • Il bit di uscita Completato è FALSE.
6, 7	Nessuna modifica nella condizione ramo.
8	Il ramo passa a TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Esegue è TRUE. Avvio esecuzione. • Il bit Pieno è TRUE. • Il bit di uscita Completato è TRUE.
9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Esegue è FALSE. • Il bit Pieno è TRUE. • Il bit di uscita Completato è FALSE.
10, 11	Nessuna modifica nella condizione ramo.

Errore durante l'esecuzione di FFL e FFU



Utilizzare questa tabella per determinare i valori del parametro per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Esegui è TRUE. Avvio esecuzione. • Il bit Errore è TRUE.
2, 3, 4	Nessuna modifica nella condizione ramo.
5	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Esegui è FALSE. • I bit Error ed ErrorID sono FALSE.
6, 7	Nessuna modifica nella condizione ramo.
8	Il ramo passa a TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Esegui è TRUE. Avvio esecuzione. • Il bit Errore è TRUE.
9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Esegui è FALSE. • I bit Error ed ErrorID sono FALSE.
10, 11	Nessuna modifica nella condizione ramo.

Vedere anche

[FFL](#) a pagina 515

[FFU](#) a pagina 515

[Istruzioni di controllo dei processi](#) a pagina 503

FFU (Scarica FIFO)

Scarica i dati a 8, 16, 32 e 64 bit da un vettore creato dall'utente denominato stack FIFO (First In First Out, primo a entrare primo a uscire) nello stesso ordine in cui i dati sono stati caricati usando l'istruzione FFL.

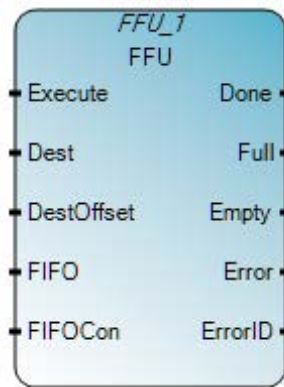
Dettagli operazione:

- Istruzione FFU: da modalità di non esecuzione in modalità di esecuzione:
 - Quando Esegue passa da FALSE a TRUE:
 - Le condizioni di errore FFU sono verificate.
 - Scarica i contenuti dello stack FIFO in corrispondenza della posizione zero se la Posizione è maggiore di zero e minore o uguale alla Lunghezza.
 - Gli elementi rimanenti si spostano di una posizione verso lo zero e l'elemento massimo dello stack FIFO viene impostato su zero, la Posizione diminuisce di 1.
 - Il bit Vuoto è impostato su TRUE se Posizione è uguale a zero.
 - Il bit Completato viene impostato quando l'istruzione viene eseguita correttamente.
 - Quando Esegue passa da FALSE a TRUE:
 - I bit Errore, Completato e ErrorID sono impostati su FALSE.
 - Il bit Vuoto è impostato su TRUE se Posizione è uguale a zero.
 - Il bit Pieno è impostato su TRUE se la Lunghezza è uguale alla Posizione. Il bit Pieno è impostato su FALSE se la Posizione è minore o uguale a zero e minore della Lunghezza.
 - Le condizioni di errore FFU non sono verificate.
 - Quando Esegue passa da TRUE a TRUE:
 - Non viene eseguita alcuna operazione di scaricamento.
 - Il bit Vuoto è impostato su TRUE se Posizione è uguale a zero.
 - Il bit Pieno è impostato su TRUE se la Lunghezza è uguale alla Posizione.
 - Le condizioni di errore FFU non sono verificate.
 - Quando Esegue passa da FALSE a FALSE:
 - I bit Errore, Completato e ErrorID sono impostati su FALSE.
 - Il bit Vuoto è impostato su TRUE se Posizione è uguale a zero.
 - Il bit Pieno è impostato su TRUE se la Lunghezza è uguale alla Posizione. Il bit Pieno è impostato su FALSE se la Posizione è minore o uguale a zero e minore della Lunghezza.
 - Le condizioni di errore FLU non sono verificate.
- Istruzione FFU: dalla modalità in esecuzione alla modalità non in esecuzione:

- I bit Errore, ErrorID, Completato, Vuoto e Pieno conservano lo stato modalità in esecuzione.
- Per creare un singolo elemento per il parametro FIFO:
 - Non vettore:
 - L'indirizzo basato sulla variabile come Fifo1 è consentito per FIFO.
 - La lunghezza deve essere configurata come 1.
 - Vettore
 - L'indirizzo basato sulla variabile come Fifo1 o Fifo1[0] è consentito per FIFO.
 - La lunghezza deve essere configurata come 1.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: se viene rilevato il fronte di salita rilevato, avviare l'operazione FFU. FALSE: fronte di salita non rilevato.
Dest	Ingresso	ANY_ELEMENTARY	Contiene il valore esistente nello stack FIFO. Tipi di dati elementari supportati per Dest: <ul style="list-style-type: none"> • DWORD, REAL, TIME, DATE, LWORD, ULINT, LINT, LREAL, BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT. • La stringa non è supportata. • Supporta solo vettori unidimensionali come Vettore[1] o Vettore[Indice].
DestOffset	Ingresso	UINT	Offset elemento di destinazione. Offset elemento se il tipo di destinazione è dati vettore, altrimenti impostare l'offset su 0. Per tipo di dati vettore, per scaricare nel primo elemento, impostare l'offset su 0.

FIFO	Ingresso	ANY_ELEMENTARY	L'indirizzo iniziale dello stack. FIFO deve essere configurato come nelle istruzioni FFL e FFU. <ul style="list-style-type: none"> • Tipi di dati elementari supportati per FIFO: DWORD, REAL, TIME, DATE, LWORD, ULINT, LINT, LREAL, BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT. • La stringa non è supportata. • È supportato solo la singola dimensione per FIFO.
FIFOCon	Ingresso	FF_LF_CON	Controllo e configurazione FIFO. La stessa configurazione deve essere configurata per le istruzioni FFL e FFU. Per configurare posizione e lunghezza, utilizzare il tipo di dati FF_LF_CON.
Done	Uscita	BOOL	Indica se l'operazione FFU è stata completata. TRUE: operazione completata correttamente. FALSE: l'operazione ha rilevato una condizione di errore o l'istruzione FFU non è in esecuzione.
Completo	Uscita	BOOL	Indica se lo stack FIFO è pieno. TRUE: quando lunghezza è uguale a posizione. FALSE: quando la posizione è maggiore di zero e minore della lunghezza.
Vuoto	Uscita	BOOL	Indica se lo stack FIFO è vuoto. TRUE: quando la posizione è uguale a 0. FALSE: quando la posizione non è uguale a 0.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE: l'operazione ha riscontrato un errore. FALSE: l'operazione è stata completata o l'istruzione FFU non è in esecuzione.
ErrorID	Uscita	USINT	Numero univoco che identifica l'errore. Gli errori sono definiti nei codici di errore FFU.

Tipo di dati FF_LF_CON

Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati FF_LF_CON.

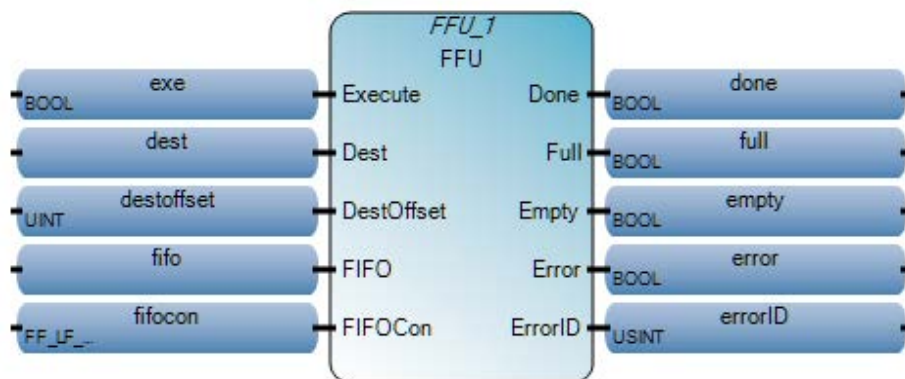
Parametri	Tipo di dati	Descrizione
Lunghezza	UINT	Numero di elementi utilizzati per l'operazione FIFO. Il limite massimo è 1024.
Position	USINT	Determina la posizione disponibile successiva in FIFO per la rimozione o l'immissione di Src. La posizione è l'offset del vettore. Esempio 1: <ul style="list-style-type: none"> • Vettore configurato dall'utente, arr[0..5]. La posizione iniziale è configurata come 1. I dati sono inseriti in arr[1] e la posizione aumenta di + 1. Esempio 2: <ul style="list-style-type: none"> • Vettore configurato dall'utente come arr[1..5]. La posizione iniziale è configurata come 1. I dati sono inseriti in arr[2] e la posizione aumenta di + 1.

Codici di errore FFU

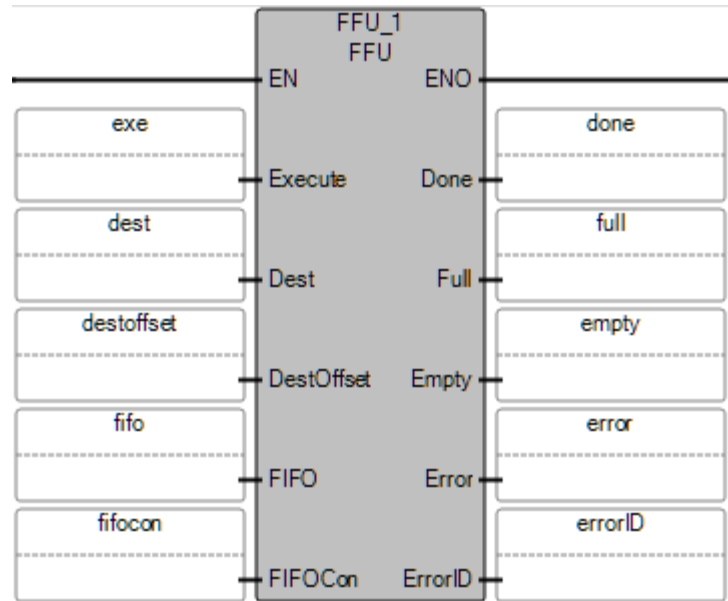
Utilizzare questa tabella per determinare i codici di errore FFL e FFU e le descrizioni.

Codice errore	Descrizione errore
0	Nessun errore.
1	Il tipo di dati Src FFL non è supportato.
2	Il tipo di dati Dest FFU non è supportato.
3	Il tipo di dati FIFO non è supportato.
4	I tipi di dati Src e Dest non corrispondono al tipo di dati FIFO. Azione correttiva: Il tipo di dati del parametro Src FFL e del parametro Dest FFU devono corrispondere al tipo di dati del vettore FIFO.
5	FIFO: la dimensione del vettore non è supportata. Azione correttiva: FIFO supporta solo vettori unidimensionali.
6	La lunghezza del controllo FIFOCon supera la dimensione del vettore FIFO. Azione correttiva: La lunghezza del controllo FIFOCon non supera la dimensione del vettore FIFO.
7	La lunghezza FIFOCon supera la lunghezza massima.
8	La lunghezza FIFOCon è zero.
9	La posizione FIFOCon supera la lunghezza FIFOCon.
10	La posizione e la lunghezza del controllo FFL sono uguali.
11	La posizione del controllo FFU è zero.
12	La dimensione del vettore FFL o FFU non è supportata. Azione correttiva: FFL e FFU supportano solo vettori unidimensionali.
13	FFL o FFU DestOffset supera la dimensione vettore Dest.

Esempio di Programmazione a blocchi funzionali FFU



Esempio di Diagramma ladder FFU



Esempio di Testo strutturato FFU

```

1 | FFU_1(exe, dest, destoffset, fifo, fifoon);
2 | done := FFU_1.Done;
3 | full := FFU_1.Full;
4 | empty := FFU_1.Empty;
5 | error := FFU_1.Error;
6 | errorID := FFU_1.ErrorID;

```

```

FFU_1(
void FFU_1(BOOL Execute, ANY_ELEMENTARY[1..1] Dest, UINT DestOffset, ANY_ELEMENTARY[1..1] FIFO, FF_LF_CON FIFOCOn)
Tipo: FFU, Scarico FIFO

```

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dime
dest	<input type="checkbox"/>	BOOL	[1..1]
destoffset	0	N/D		<input type="checkbox"/>	UINT	
done	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
empty	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
error	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
errorID	0	N/D		<input type="checkbox"/>	USINT	
exe	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
FFU_1	<input type="checkbox"/>	FFU	
fifo	<input type="checkbox"/>	BOOL	[1..10]
fifo[1]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[2]	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[3]	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[4]	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[5]	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[6]	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[7]	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[8]	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[9]	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifo[10]	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
fifocon	<input type="checkbox"/>	FF_LF_CON	
fifocon.Length	10	N/D		<input type="checkbox"/>	UINT	
fifocon.Position	0	N/D		<input type="checkbox"/>	UINT	
full	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	

Vedere anche

[Diagrammi di temporizzazione istruzione FFL e FFU](#) a pagina 510

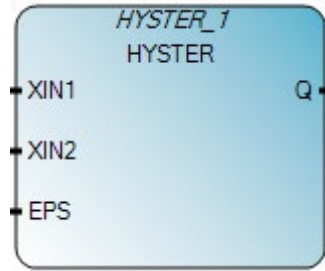
[Istruzioni di controllo dei processi](#) a pagina 503

HYSTER (isteresi)

Isteresi booleana sulla differenza tra numeri reali. Confronta il valore corrente di un ingresso con il limite massimo stabilito aggiungendo la quantità storica di ritardo come misurato dalla isteresi al valore previsto per un ingresso e valutando se il valore corrente supera tale limite.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

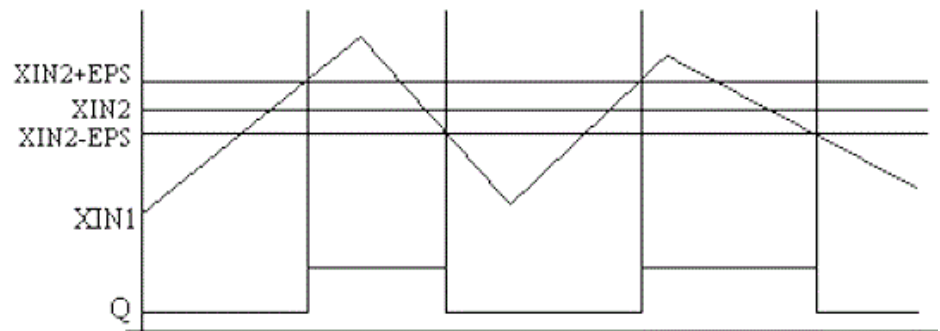


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

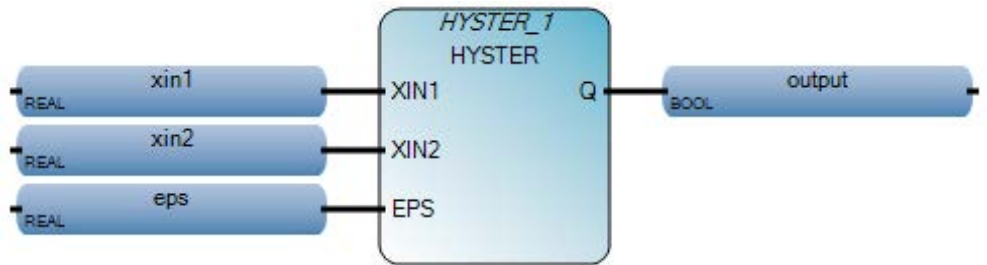
Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue il blocco istruzione. FALSE: non esegue il blocco istruzione. Applicabile solo alle programmazioni in Ladder Diagram.
XIN1	Ingresso	REAL	Qualsiasi valore reale.
XIN2	Ingresso	REAL	Per verificare se il valore di ingresso XIN1 ha superato il limite massimo definito per questo ingresso XIN2 + EPS.
EPS	Ingresso	REAL	Valore di isteresi (deve essere maggiore di zero).
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Q	Uscita	BOOL	Il risultato dell'istruzione HYSYTER. TRUE: l'ingresso supera il limite superiore, ma non è inferiore al limite inferiore. FALSE: l'ingresso non ha superato il limite superiore.

Esempio di diagramma di temporizzazione HYSYTER

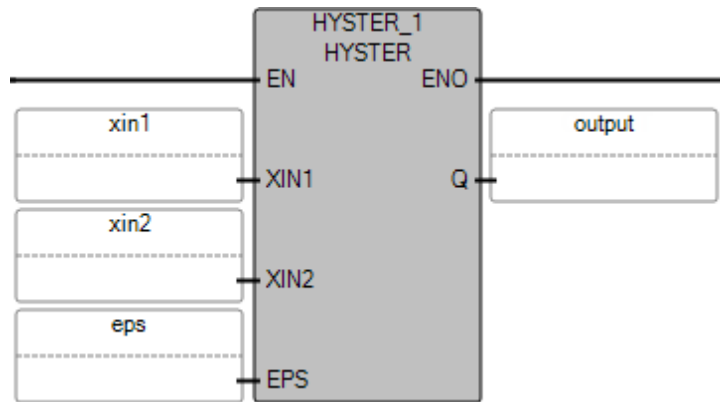
Nel seguente diagramma, HYSYTER viene utilizzato per verificare il ritardo di prestazioni a causa di attrito motore in un intervallo di tempo di 5 secondi. L'istruzione viene eseguita ogni 10 millisecondi. Durante la fase di avvio il motore ha funzionato in maniera efficiente.



Esempio di programmazione a blocchi funzionali HYSTER



Esempio di diagramma ladder HYSTER



Esempio di testo strutturato HYSTER

```

HYSTER_1 {
void HYSTER_1(REAL XIN1, REAL XIN2, REAL EPS)
Tipo: HYSTER, Isteresi booleana su differenza di dati di tipo reale

1: xin1 := 10.0;
2: xin2 := 1.0;
3: eps := 1.0;
4: HYSTER_1(xin1, xin2, eps);
5: output := HYSTER_1.Q;

```

Vedere anche

[Istruzioni di controllo dei processi](#) a pagina 503

INTEGRAL

Integra un valore real durante la durata del ciclo definito.

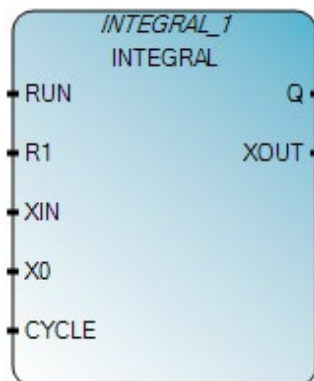
Dettagli operazione:

- Quando un blocco funzione INTEGRAL viene inizializzato per la prima volta, i valori iniziali non vengono considerati. Usare il parametro R1 per impostare i valori iniziali per il calcolo.

- Per evitare la perdita del valore incorporato, il valore di integrazione non viene cancellato automaticamente quando il controllore passa da PROGRAM a RUN o quando il parametro RUN passa da FALSE a TRUE. Usare il parametro R1 per cancellare il valore dell'integrale quando il controllore per la prima volta passa dalla modalità PROGRAM a RUN o quando si inizia una nuova integrazione.
- Si raccomanda di non usare i parametri EN o ENO facoltativi con questo blocco funzione, in quanto il calcolo della durata del ciclo sarà interrotto quando EN è FALSE, con conseguente errore di integrazione. Se si utilizzano i parametri EN o ENO, commutare il parametro R1 con EN uguale a TRUE per cancellare il risultato corrente e assicurare una corretta integrazione.
- L'integrazione viene eseguita su una base temporale in millisecondi (ovvero, integrando un input di 1 con un valore iniziale di 0 per 1 secondo il risultato sarà pari a 1000). Per convertire l'uscita dell'istruzione in secondi, il valore in uscita deve essere diviso per 1000.
- Se il valore del parametro CYCLE è inferiore all'intervallo di tempo del ciclo di esecuzione del dispositivo, il periodo di campionamento viene forzato all'intervallo di tempo del ciclo.
- L'esecuzione del campionamento XIN e del blocco funzione si verifica ad ogni durata del ciclo + Scan Time Jitter.
- Per un dato programma utente, Scan Time Jitter varia da controllore a controllore.
- La durata del ciclo determina la sensibilità del blocco funzione Integral. Le variazioni che si verificano in XIN tra due campionamenti (o all'interno della durata del ciclo) non vengono prese in considerazione quando viene calcolato il valore dell'integrale XOUT.
- La durata del ciclo e il valore di Scan Time Jitter influiscono entrambi sull'imprecisione complessiva dell'uscita Integral come mostrato nel XIN in sincronia con l'esempio di esecuzione del blocco funzione e nel XIN non in sincronia con l'esecuzione del blocco funzione.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

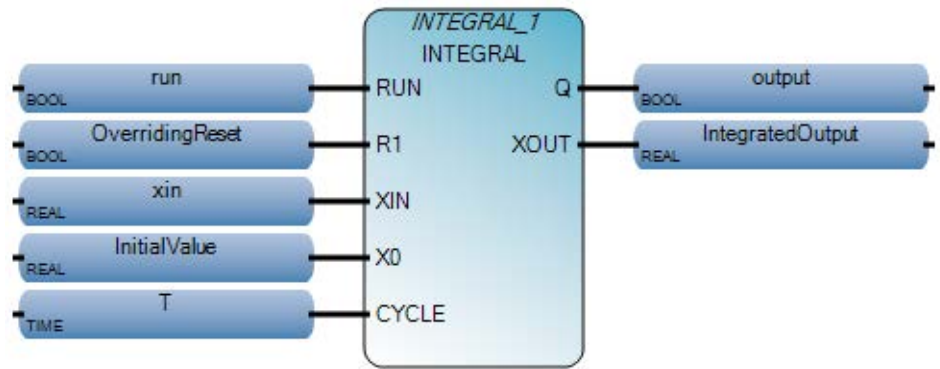
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



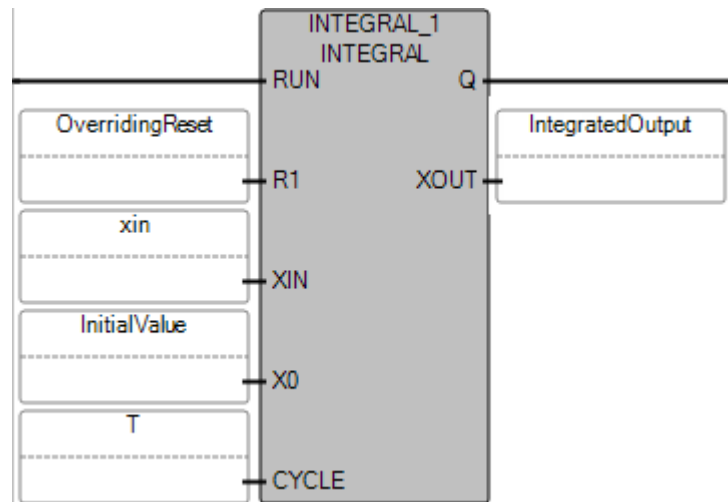
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
RUN	Ingresso	BOOL	Modalità: TRUE = integra / FALSE = attesa.
R1	Ingresso	BOOL	Ignora ripristino.
XIN	Ingresso	REAL	Ingresso: qualsiasi valore reale.
X0	Ingresso	REAL	Valore iniziale.
CYCLE	Ingresso	TIME	Periodo di camp. I valori possibili variano da 0ms a 49d17h2m47s294ms.
Q	Uscita	BOOL	Non R1.
XOUT	Uscita	REAL	Uscita integrata.

Esempio di programmazione a blocchi funzionali INTEGRAL



Esempio di diagramma ladder integrale



Esempio di testo strutturato integrale

```
INTEGRAL_1(
void INTEGRAL_1(BOOL RUN, BOOL R1, REAL XIN, REAL X0, TIME CYCLE)
Tipo: INTEGRAL, Integrazione nel tempo
```

```
1: run := TRUE;
2: xin := 10.0;
3: InitialValue := 5.0;
4: T := T#10s;
5: INTEGRAL_1(run, OverridingReset, xin, InitialValue, T);
6: output := NOT OverridingReset;
7: IntegratedOutput := INTEGRAL_1.XOUT;
```

(* Equivalenza ST: INTEGRAL1 è un'istanza di un blocco INTEGRAL *)

```
INTEGRAL1(manual_mode, NOT(manual_mode), sensor_value,
init_value, t#100ms);
controlled_value := INTEGRAL1.XOUT;
```

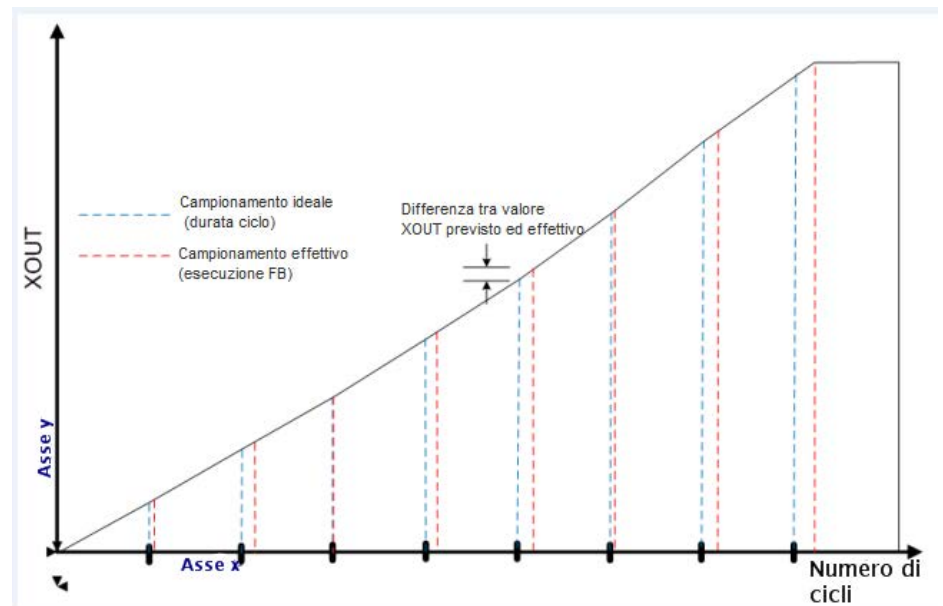
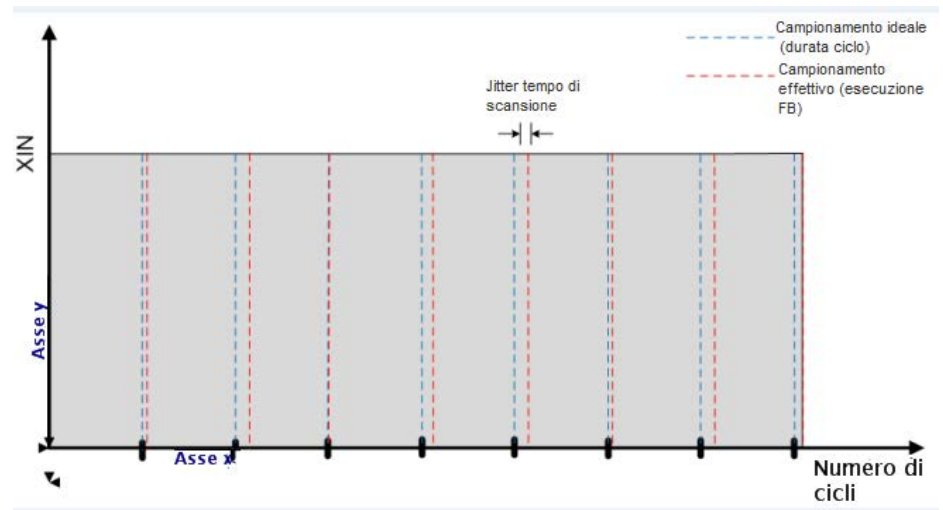
Risultati

The screenshot shows a window titled "Monitoraggio della variabile" with a table of variables. The table has columns for Name, Logical Value, Physical Value, Initial Value, Lock, and Data Type. The variables listed are InitialValue, INTEGRAL_1, IntegratedOutput, output, OverridingReset, run, T, and xin.

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
InitialValue	5.0	N/D		<input type="checkbox"/>	REAL
INTEGRAL_1	<input type="checkbox"/>	INTEGRAL
IntegratedOutput	6.404045E+07	N/D		<input type="checkbox"/>	REAL
output	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
OverridingReset	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
run	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
T	T#0s	N/D		<input type="checkbox"/>	TIME
xin	10.0	N/D		<input type="checkbox"/>	REAL

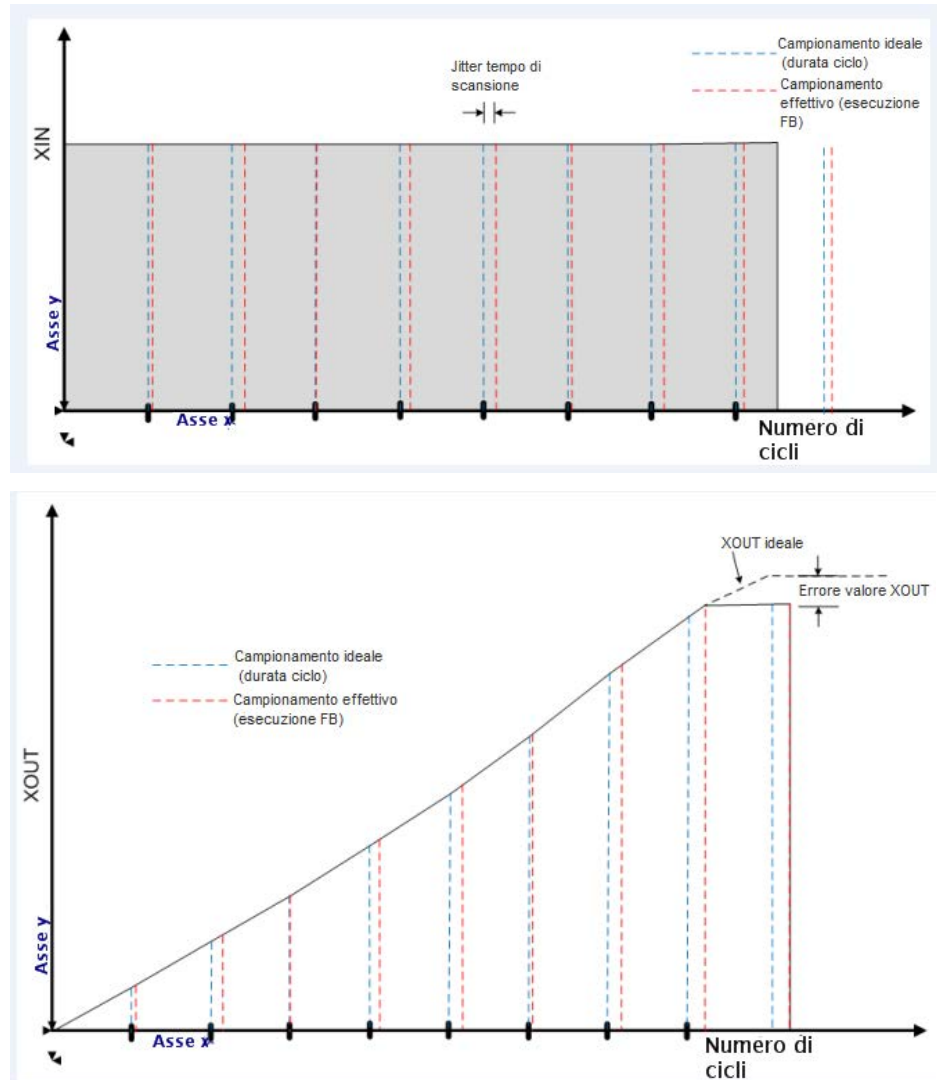
Esempio di XIN in sincronismo con l'esecuzione del blocco funzione

Le figure seguenti mostrano l'effetto di Scan Time Jitter sul valore XOUT:



Esempio di XIN non in sincronismo con l'esecuzione del blocco funzione

Le figure seguenti mostrano un esempio in cui viene introdotto un errore nel valore di XOUT per un blocco funzione Integral:



Vedere anche

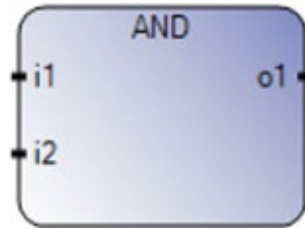
[Istruzioni di controllo dei processi](#) a pagina 503

AND

Esegue un'operazione booleana AND tra due o più valori.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
i1	Ingresso	BOOL	Valore in tipo di dati Boolean.
i2	Ingresso	BOOL	Valore in tipo di dati Boolean.
o1	Uscita	BOOL	Risultato dell'operazione booleana AND dei valori di ingresso.

Esempio di testo strutturato AND

(* Equivalenza ST: *)

```
bo10 := bi101 AND NOT (bi102);
bo5 := (bi51 AND bi52) AND bi53;
```

Vedere anche

[Istruzioni booleane](#) a pagina 159

LFL (carico LIFO)

L'istruzione LFL viene utilizzata per caricare i dati (8 bit, 16 bit, 32 bit o 64 bit) in un vettore creato dall'utente denominato stack LIFO. Le istruzioni LFL e LFU vengono utilizzate in coppia.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

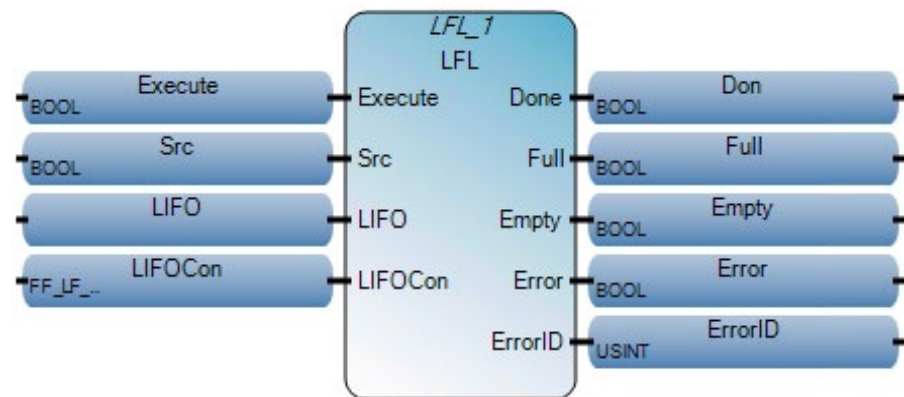
Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



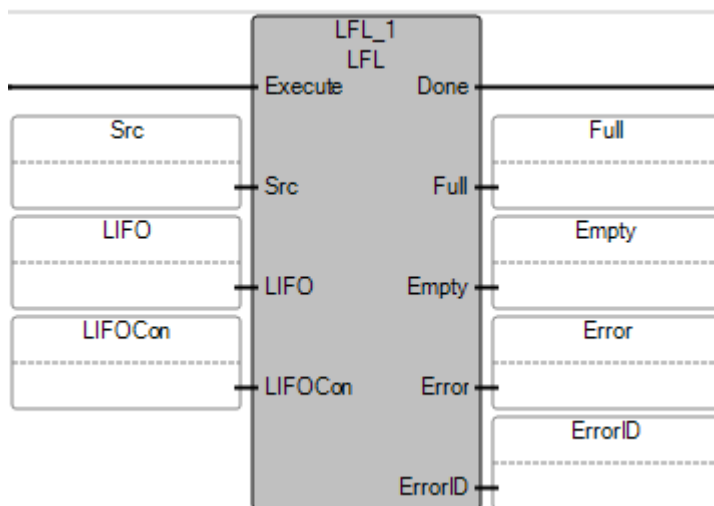
Utilizzare questa tabella per determinare il valore del parametro per queste istruzioni.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: se viene rilevato il fronte di salita, avviare l'operazione LFL. FALSE: fronte di salita non rilevato.
Src	Ingresso	ANY_ELEMENTARY	L'operando Src è l'indirizzo del valore usato per compilare la posizione attualmente disponibile nello stack LIFO. Tipi di dati elemento supportati: <ul style="list-style-type: none"> • DWORD, REAL, TIME, DWORD, REAL, TIME, DATE, LWORD, ULINT, LINT, LREAL, BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT.\ • Elemento del vettore come Vettore [1] o Vettore[Indice]. • La stringa non è supportata
LIFO	Ingresso	ANY_ELEMENTARY	L'indirizzo iniziale dello stack. LIFO deve essere configurato come per le istruzioni LFL e LFU. Tipi di dati elemento supportati: <ul style="list-style-type: none"> • DWORD, REAL, TIME, DWORD, REAL, TIME, DATE, LWORD, ULINT, LINT, LREAL, BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT. • È supportata solo la singola dimensione. • La stringa non è supportata.
LIFOCon	Ingresso	FF_LF_CON	Controllo e configurazione LIFO. La stessa configurazione deve essere definita per le istruzioni LFL e LFU. Per configurare posizione e lunghezza, utilizzare il tipo di dati FF_LF_CON.
Completo	Uscita	BOOL	Indica se lo stack LIFO è pieno. TRUE: quando lunghezza è uguale a posizione. FALSE: quando la posizione è maggiore o uguale a zero e minore della lunghezza.
Vuoto	Uscita	BOOL	Indica se lo stack LIFO è vuoto. TRUE: quando la posizione è uguale a zero. FALSE: quando la posizione non è uguale a zero.
Errore	Uscita	BOOL	Indicare la presenza di una condizione di errore. TRUE: l'operazione ha riscontrato un errore. FALSE: l'operazione è stata completata o l'istruzione non è in esecuzione.
ErrorID	Uscita	USINT	Numero univoco che identifica l'errore. Gli errori sono definiti nei codici di errore LFL.
Done	Uscita	BOOL	Indicare quando l'operazione è completata. TRUE: l'operazione è stata completata. FALSE: l'operazione ha rilevato una condizione di errore o l'istruzione FFU non è in esecuzione.

Esempio di programmazione a blocchi funzionali LFL



Esempio di Diagramma ladder LFL



Esempio di Testo strutturato LFL

```
LFL_1(
void LFL_1(BOOL Execute, ANY_ELEMENTARY Src, ANY_ELEMENTARY[1..1] LIFO, FF_LF_CON LIFOCon)
Tipo: LFL, Carico LIFO
```

```
1 LFL_1(exe, Src, fifo, fifocon);
2 Done:=LFL_1.Done;
3 Full:=LFL_1.Full;
4 Empty:=LFL_1.Empty;
5 Error:=LFL_1.Error;
6 ErrorID:=LFL_1.ErrorID;
```

Vedere anche

[Istruzioni di controllo dei processi](#) a pagina 503

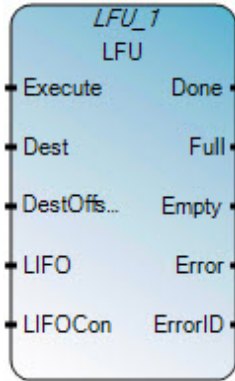
[LFU \(scarica LIFO\)](#) a pagina 530

LFU (scarica LIFO)

L'istruzione LFU scarica i dati (8 bit, 18 bit, 32 bit, 64 bit) da un vettore creato dall'utente denominato stack LIFO. Le istruzioni LFU e LFL vengono utilizzate in coppia.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

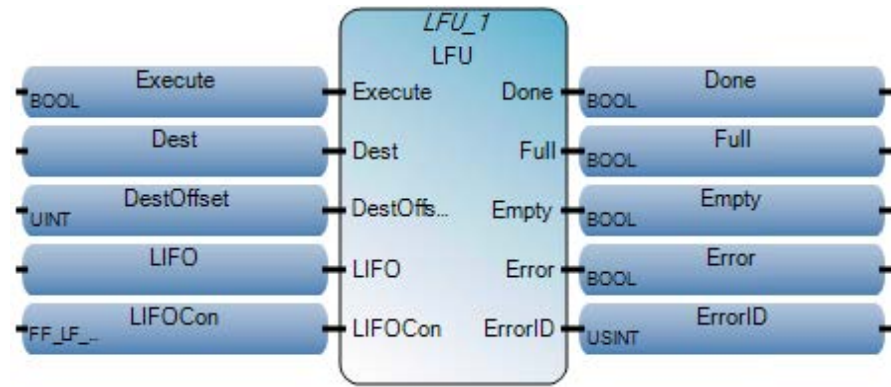
Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



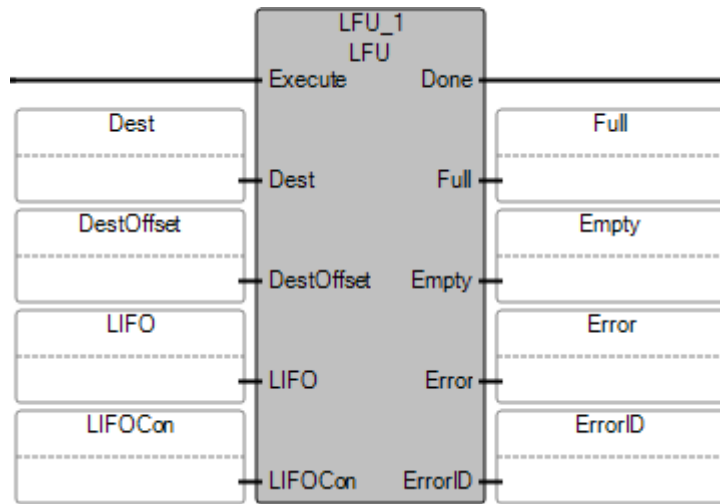
Utilizzare questa tabella per determinare il valore del parametro per queste istruzioni.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: se viene rilevato il fronte di salita, avvia l'operazione LFLU. FALSE: fronte di salita non rilevato.
Dest	Ingresso	ANY_ELEMENTARY	Contiene il valore esistente nello stack LIFO. Tipi di dati elementari supportati per Dest: <ul style="list-style-type: none"> • DWORD, REAL, TIME, DATE, LWORD, ULINT, LINT, LREAL, BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT. • La stringa non è supportata. • Supporta solo vettori unidimensionali come Vettore[1] o Vettore[Indice].
DestOffset	Ingresso	UINT	Offset elemento di destinazione. Offset elemento se il tipo di destinazione è dati vettore, altrimenti impostare l'offset su 0. Per tipo di dati vettore, per scaricare nel primo elemento, impostare l'offset su 0.
LIFO	Ingresso	ANY_ELEMENTARY	L'indirizzo iniziale dello stack. LIFO deve essere configurato come per le istruzioni LFL e LFU. Tipi di dati elemento supportati: <ul style="list-style-type: none"> • DWORD, REAL, TIME, DWORD, REAL, TIME, DATE, LWORD, ULINT, LINT, LREAL, BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT. • È supportata solo la singola dimensione. • La stringa non è supportata.
LIFOCon	Ingresso	FF_LF_CON	Controllo e configurazione LIFO. La stessa configurazione deve essere definita per le istruzioni LFL e LFU. Per configurare posizione e lunghezza, utilizzare il tipo di dati FF_LF_CON.
Completo	Uscita	BOOL	Indica se lo stack LIFO è pieno. TRUE: quando lunghezza è uguale a posizione. FALSE: quando la posizione è maggiore o uguale a zero e minore della lunghezza.
Errore	Uscita	BOOL	Indicare la presenza di una condizione di errore. TRUE: l'operazione ha riscontrato un errore. FALSE: l'operazione è stata completata o l'istruzione non è in esecuzione.
ErrorID	Uscita	USINT	Numero univoco che identifica l'errore. Gli errori sono definiti nei codici di errore LFLU.
Done	Uscita	BOOL	Indicare quando l'operazione è completata. TRUE: l'operazione è stata completata. FALSE: l'operazione ha rilevato una condizione di errore o l'istruzione LFLU non è in esecuzione.

Esempio di programmazione a blocchi funzionali LFU



Esempio di Diagramma ladder LFU



Esempio di Testo strutturato LFU

```
LFU_1(
void LFU_1(BOOL Execute, ANY_ELEMENTARY[1..1] Dest, UINT DestOffset, ANY_ELEMENTARY[1..1] LIFO, FF_LF_CON LIFOCon)
Tipo: LFU, Scarico LIFO
```

```
1 LFU_1(exe, Dest, destoffset, lifo, lifocon);
2 Done:=LFU_1.Done;
3 Full:=LFU_1.Full;
4 Empty:=LFU_1.Empty;
5 Error:=LFU_1.Error;
6 ErrorID:=LFU_1.ErrorID;
```


Vedere anche

[Istruzioni di controllo dei processi](#) a pagina 503

[LFL \(carico LIFO\)](#) a pagina 528

PWM (Pulse Width Modulation)

Attiva o disattiva l'uscita del PMW (Pulse Width Modulation) per un canale PWM configurato.

Questo blocco istruzione viene utilizzato con i controllori Micro820 2080-LC20-20QBB e supporta un canale PWM utilizzando il canale di uscita 6 integrato.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale solo per il controllore 2080-LC20-20QBB Micro820.

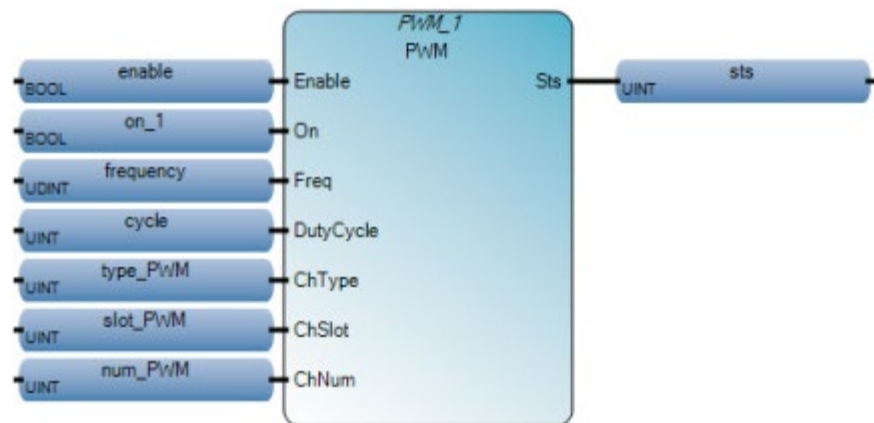


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

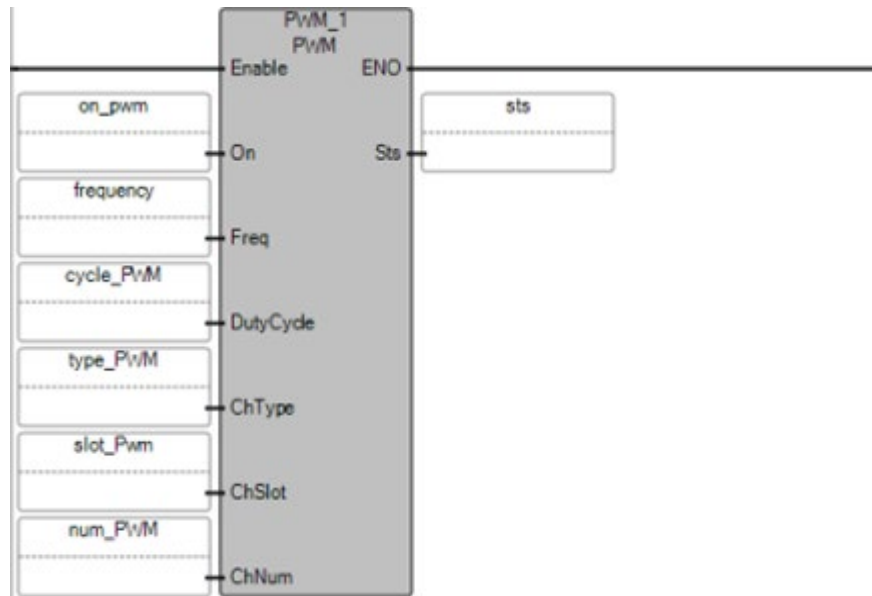
Parametri	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. Questo livello è attivato per il blocco istruzione. TRUE: aggiornare Sts. PWM viene attivato o disattivato a seconda del parametro di ingresso On e della configurazione valida. FALSE: Sts viene soltanto aggiornato. Lo stato (attivo o inattivo) di PWM non subisce variazioni.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
On	Ingresso	BOOL	Imposta l'uscita PWM su On/Attivo o Off/inattivo. TRUE: l'uscita PWM viene attivata o continua a essere attiva con la configurazione valida più recente. Il LED uscita è ON quando PWM è attivo, anche se il duty cycle è impostato su 0%. FALSE: l'uscita PWM viene disattivata se anche la configurazione è valida.
Freq	Ingresso	UDINT	Frequenza di impulso. • 1 - 100.000 Hz
DutyCycle	Ingresso	UINT	Ciclo di lavoro per impulso. • 0 - 1000 (0% - 100%)
ChType	Ingresso	UINT	Tipo di canale • 0: integrato • 1: plug-in • 2: espansione
ChSlot	Ingresso	UINT	Slot canale • 0: integrato
ChNum	Ingresso	UINT	Numero canale • 0 - PWM CHO
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.
Sts	Uscita	UINT	Codici di stato PWM: • 00: blocco funzione non abilitato (nessuna operazione). • 01: configurazione PWM eseguita con successo. • 02: ciclo di carico non valido. • 03: frequenza non valida. • 04: tipo canale non valido. • 05: slot canale non valido. • 06: numero canale non valido. • 07: catalogo non valido. Nel catalogo in uso la funzionalità PWM non è supportata.

Esempio di Programmazione a blocchi funzionali PWM



Esempio di Diagramma ladder PWM



Esempio di Testo strutturato PWM

```
PWM_1 (
void PWM_1(BOOL Enable, BOOL On, UDINT Freq, UINT DutyCycle, UINT ChType, UINT ChSlot, UINT ChNum)
Tipo: PWM, Abilitare l'uscita PWM.

1  PWM (EN, Enable, On, Freq, DutyCycle, ChType, ChSlot, ChNum);
2  output := PWM.ENO
3  sts := PWM.Sts
```

Vedere anche

[Istruzioni di controllo dei processi](#) a pagina 503

SCALER (scala)

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

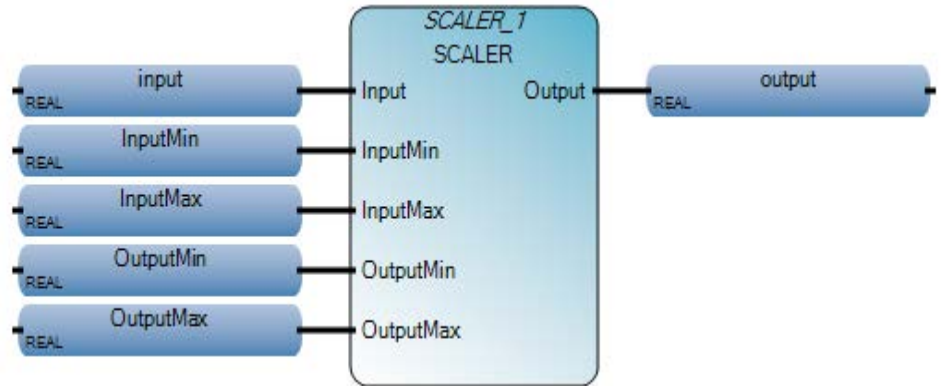
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.,

Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

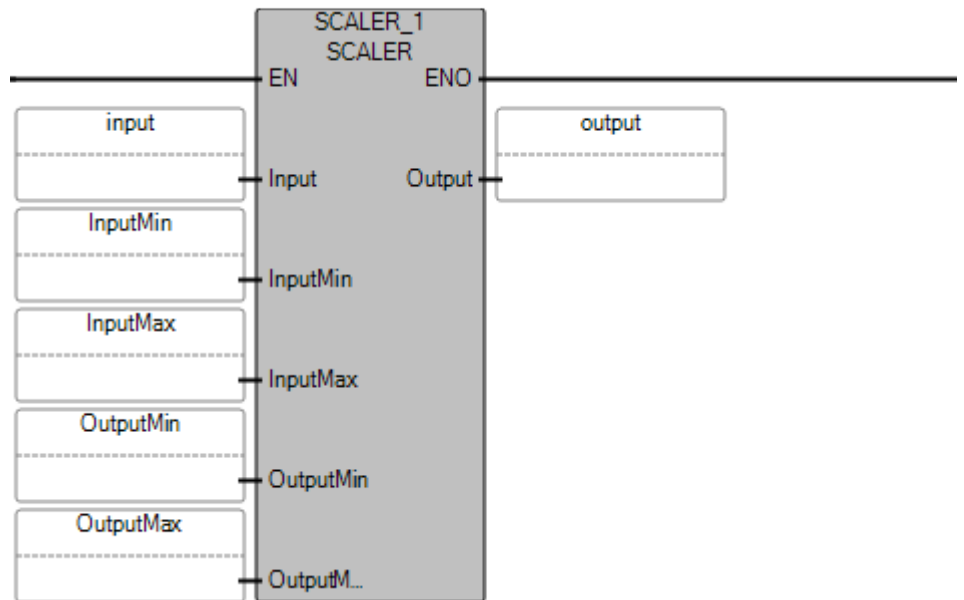
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue l'equazione per scalare. FALSE: nessuna equazione per scalare. Applicabile solo alle programmazioni in Ladder Diagram.
Ingresso	Ingresso	REAL	Ingresso segnale. L'ingresso non è limitato da InputMin e InputMax. Per limitare l'ingresso, è necessaria un'istruzione LIM_ALARM che definisca le condizioni dell'ingresso prima che venga inserito nell'istruzione SCALER.
InputMin	Ingresso	REAL	Determina la pendenza e il valore dell'offset.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
InputMax	Ingresso	REAL	Determina la pendenza e il valore dell'offset.
OutputMin	Ingresso	REAL	Determina la pendenza e il valore dell'offset.
OutputMax	Ingresso	REAL	Determina la pendenza e il valore dell'offset.
Uscita	Uscita	REAL	Uscita scalata. Uscita non limitata da OutputMin e OutputMax.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Programmazione a blocchi funzionali SCALER



Esempio di Diagramma ladder SCALER



Esempio di Testo strutturato SCALER

```

SCALER_1 (
void SCALER_1(REAL Input, REAL InputMin, REAL InputMax, REAL OutputMin, REAL OutputMax)
Tipo: SCALER, Ridimensionare il valore di ingresso in base all'intervallo di uscita.

1 input := 10.0;
2 InputMin := 5.0;
3 InputMax := 15.0;
4 OutputMin := 1.0;
5 OutputMax := 10.0;
6 SCALER_1(input, InputMin, InputMax, OutputMin, OutputMax);
7 output := SCALER_1.Output;

```

(* Equivalenza ST: SCALER1 è un'istanza del blocco SCALER *)

```

SCALER1(Signal_In, 4.0, 20.0 , 0.0 , 150.0 ) ;
Out_Temp := SCALER1.Output ;

```

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
SCALER_1	<input type="checkbox"/>	SCALER
OutputMin	1.0	N/D		<input checked="" type="checkbox"/>	REAL
OutputMax	10.0	N/D		<input type="checkbox"/>	REAL
output	5.5	N/D		<input type="checkbox"/>	REAL
InputMin	5.0	N/D		<input type="checkbox"/>	REAL
InputMax	15.0	N/D		<input type="checkbox"/>	REAL
input	10.0	N/D		<input type="checkbox"/>	REAL

Vedere anche

[Istruzioni di controllo dei processi](#) a pagina 503

STACKINT (valori interi dello stack)

Gestisce uno stack di valori interi.

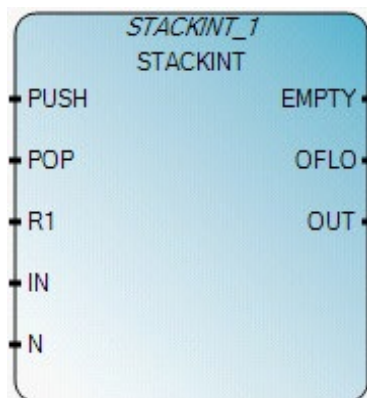
Dettagli operazione:

- STACKINT include un rilevamento di fronte di salita per entrambi i comandi PUSH e POP. La dimensione massima dello stack è 128. Il valore OFLO è valido solo dopo un ripristino (R1 è stato impostato su TRUE almeno una volta e quindi reimpostato su FALSE).
- La dimensione (N) dello stack definita dall'applicazione non può essere minore di 1 o maggiore di 128.

- Se $N < 1$, STACKINT assume una dimensione pari a 1.
- se $N > 128$, STACKINT assume una dimensione pari a 128.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

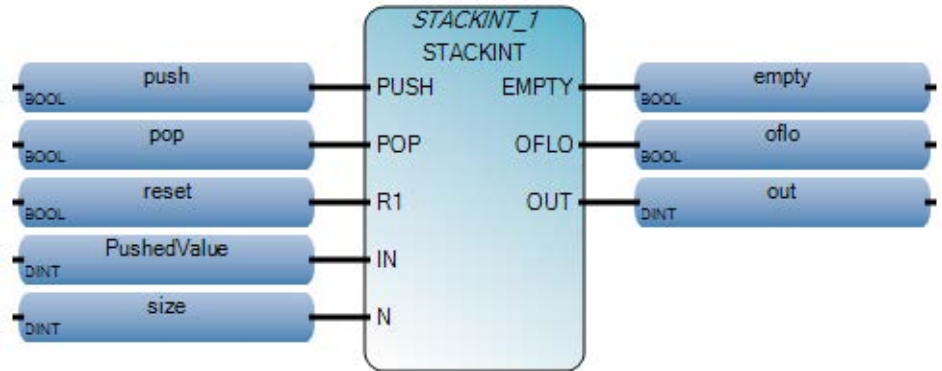
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



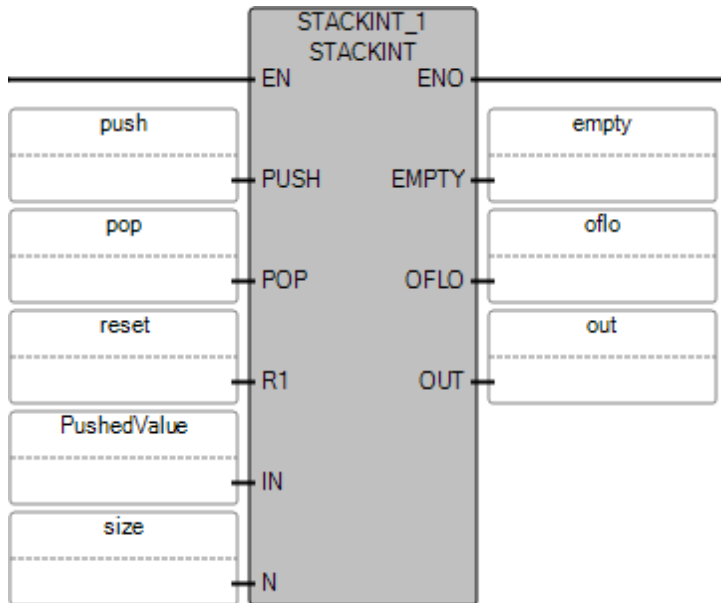
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
PUSH	Ingresso	BOOL	TRUE: fronte di salita rilevato, su comando PUSH. Aggiunge il valore IN all'inizio dello stack. FALSE: fronte di salita non rilevato su comando PUSH.
POP	Ingresso	BOOL	TRUE: fronte di salita rilevato, su comando PUSH. Elimina l'ultimo valore inviato tramite push all'inizio dello stack. FALSE: fronte di salita non rilevato su comando POP.
R1	Ingresso	BOOL	TRUE: esegue il reset dello stack allo stato vuoto. FALSE: nessun reset.
IN	Ingresso	DINT	Valore spinto.
N	Ingresso	DINT	Dimensione dello stack definita dall'applicazione. La dimensione massima dello stack è 128.
EMPTY	Uscita	BOOL	TRUE: se lo stack è vuoto. FALSE: lo stack contiene valori.
OFLO	Uscita	BOOL	TRUE: Overflow, lo stack è pieno e R1 è stato impostato su TRUE almeno una volta e di nuovo su FALSE. FALSE: la dimensione dello stack è 128 o meno. Nessun overflow.
OUT	Uscita	DINT	Valore in cima allo stack. OUT uguale a 0 quando OFLO è TRUE.

Esempio di Programmazione a blocchi funzionali STACKINT



Esempio di Diagramma ladder STACKINT



Esempio di Testo strutturato STACKINT

```
STACKINT_1 (
void STACKINT_1(BOOL PUSH, BOOL POP, BOOL R1, DINT IN, DINT N)
Tipo: STACKINT, Stack di analogici interi
```

```
1 PushedValue := 5;
2 size := 10;
3 STACKINT_1(push, pop, reset, PushedValue, size);
4 empty := STACKINT_1.EMPTY;
5 oflo := STACKINT_1.OFLO;
6 out := STACKINT_1.OUT;
```

(* Equivalenza ST: STACKINT1 è un'istanza del blocco STACKINT *)

```
STACKINT1(err_detect, acknowledge, manual_mode,
err_code, max_err);
```

```

appli_alarm := auto_mode AND NOT(STACKINT1.EMPTY);
err_alarm := STACKINT1.OFLO;
last_error := STACKINT1.OUT;
    
```

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dim
empty	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
oflo	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
out	5	N/D		<input type="checkbox"/>	DINT	
pop	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
push	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
PushedValue	5	N/D		<input type="checkbox"/>	DINT	
reset	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
size	10	N/D		<input type="checkbox"/>	DINT	
STACKINT_1	<input type="checkbox"/>	STACKINT	

Vedere anche

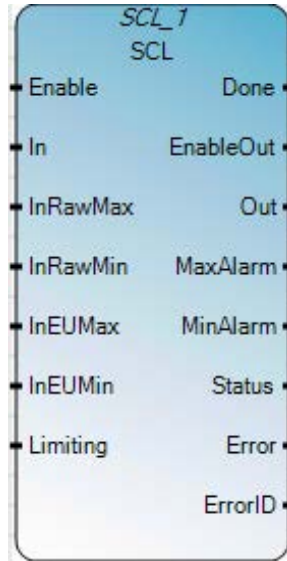
[Istruzioni di controllo dei processi](#) a pagina 503

SCL

Converte un valore di ingresso REAL non scalato in un valore in virgola mobile REAL espresso in unità ingegneristiche e include allarmi e limiti dell'uscita.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

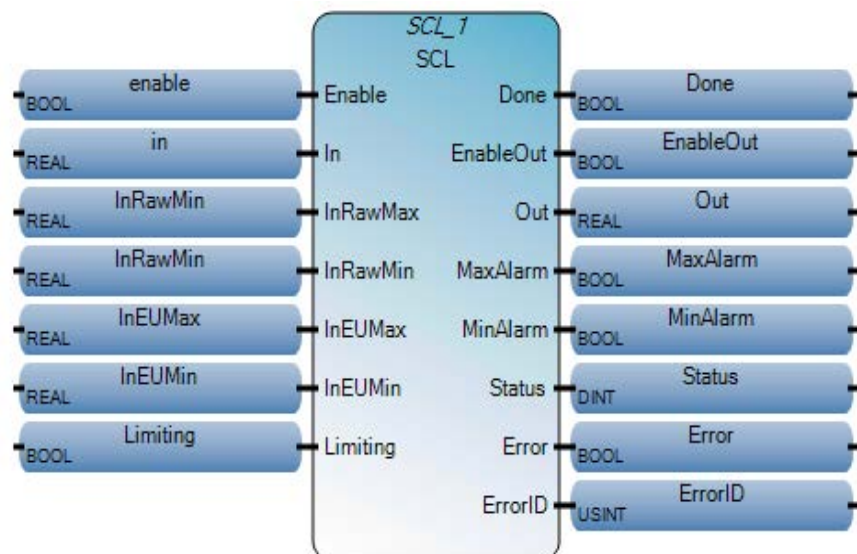
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	<p>TRUE: fronte di salita rilevato.</p> <ul style="list-style-type: none"> Se $InRawMin \geq InRawMax$, Status.0 e Status.1 i bit sono impostati su 1. Done è cancellato, MaxAlarm e MinAlarm sono cancellati. Errore è impostato su TRUE ed ErrorID è impostato su 1. Else Out è calcolato per primo. Vengono poi verificate le condizioni di allarme. Se MaxAlarm è impostato, MinAlarm è cancellato e viceversa. Quindi se la limitazione è impostata, Out sarà nell'intervallo di InEUMin e InEUMax. Quindi il bit Done è impostato su TRUE. I bit Stato sono impostati su 0. Errore ed ErrorID sono impostati su 0. Il valore Out calcolato viene confrontato con NAN (non un numero). Se NAN è il valore Out, EnableOut è cancellato. Done è cancellato se la condizione di errore è impostata. <p>FALSE: fronte di salita non rilevato.</p> <ul style="list-style-type: none"> L'istruzione non viene eseguita. Le uscite non sono aggiornate. ma Error, ErrorID, EnableOut e Done sono impostati su 0.
In	Ingresso	REAL	<p>Il segnale d'ingresso analogico. Valido = qualsiasi mobile Predefinito = 0,0</p>
InRawMax	Ingresso	REAL	<p>Il valore massimo raggiungibile dall'ingresso all'istruzione. Se $InRawMax \leq InRawMin$, l'istruzione imposta il bit appropriato in Stato; Error. ErrorID sono aggiornati. L'aggiornamento Out viene interrotto. Valido = $InRawMax > InRawMin$ Predefinito = 0,0</p>
InRawMin	Ingresso	REAL	<p>Il valore minimo raggiungibile dall'ingresso all'istruzione. Se $InRawMin \geq InRawMax$, l'istruzione imposta il bit appropriato in Stato ed Error. ErrorID sono aggiornati. L'aggiornamento Out viene interrotto. Valido = $InRawMin < InRawMax$ Predefinito = 0,0</p>
InEUMax	Ingresso	REAL	<p>Il valore scalato dell'ingresso corrispondente a InRawMax. Valido = qualsiasi valore reale Predefinito = 0,0</p>
InEUMin	Ingresso	REAL	<p>Il valore scalato dell'ingresso corrispondente a InRawMin. Valido = qualsiasi valore reale Predefinito = 0,0</p>

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Limitazione	Ingresso	BOOL	Selettore di limitazione. TRUE: Out limitato tra InEUMin e InEUMax.
Done	Uscita	BOOL	Indica quando l'operazione viene completata. TRUE: operazione completata correttamente. FALSE: si è verificata una condizione di errore per l'operazione o Enable è impostato su FALSE.
EnableOut	Uscita	BOOL	Indica che l'istruzione è abilitata. Imposta su False se Out è in overflow.
Out	Uscita	REAL	Rappresenta il valore scalato dell'ingresso analogico.
MaxAlarm	Uscita	BOOL	Indicatore di allarme ingresso massimo. Il valore è impostato su TRUE quando Ingresso > InRawMax.
MinAlarm	Uscita	BOOL	Indicatore di allarme ingresso minimo. Il valore è impostato su TRUE quando Ingresso < InRawMin.
Stato	Uscita	DINT	Stato del blocco funzione. InstructFault (Status.0) Verrà impostato dopo il rilevamento degli errori di esecuzione dell'istruzione. Non si tratta di un errore del controllore grave o di minore entità. Verificare i bit di stato rimanenti per determinare ciò che è accaduto. InRawRangeInv (Status.1) InRawMin >= InRawMax L'intervallo Status.3 - Status.31 è riservato a uso futuro e il relativo valore è 0.
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE: l'operazione ha riscontrato un errore. FALSE: l'operazione è stata completata o l'istruzione non è in esecuzione.
ErrorID	Uscita	BOOL	Numero univoco che identifica l'errore. Gli errori sono definiti nei codici di errore.

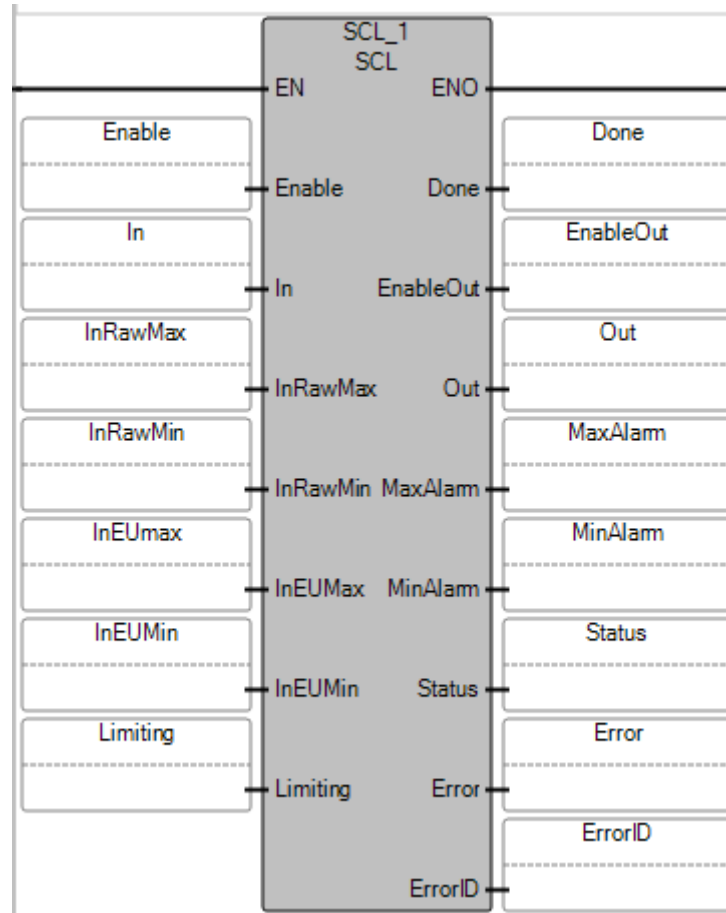
Codice errore

Codice ErrorID	Descrizione errore
1	InRawMax <= InRawMin

Esempio di programmazione a blocchi funzionali SCL



Esempio di Diagramma ladder SCL



Esempio di Testo strutturato SCL

```
SCL_1 |
void SCL_1(BOOL Enable, REAL In, REAL InRawMax, REAL InRawMin, REAL InEUMax, REAL InEUMin, BOOL Limiting)
Tipo: SCL, Un'istruzione SCL con allarme converte un valore di ingresso non scalato in un valore in virgola mobile espresso in unità ingegneristiche
```

```
1 SCL_1(Enable, In, InRawMax, InRawMin, InEUMax, InEUMin, Limiting);
2 Done :=SCL_1.Done;
3 EnableOut :=SCL_1.EnableOut;
4 Out :=SCL_1.Out;
5 MaxAlarm :=SCL_1.MaxAlarm;
6 MinAlarm :=SCL_1.MinAlarm;
7 Status :=SCL_1.Status;
8 Error :=SCL_1.Error;
9 ErrorID :=SCL_1.ErrorID;
```

Vedere anche

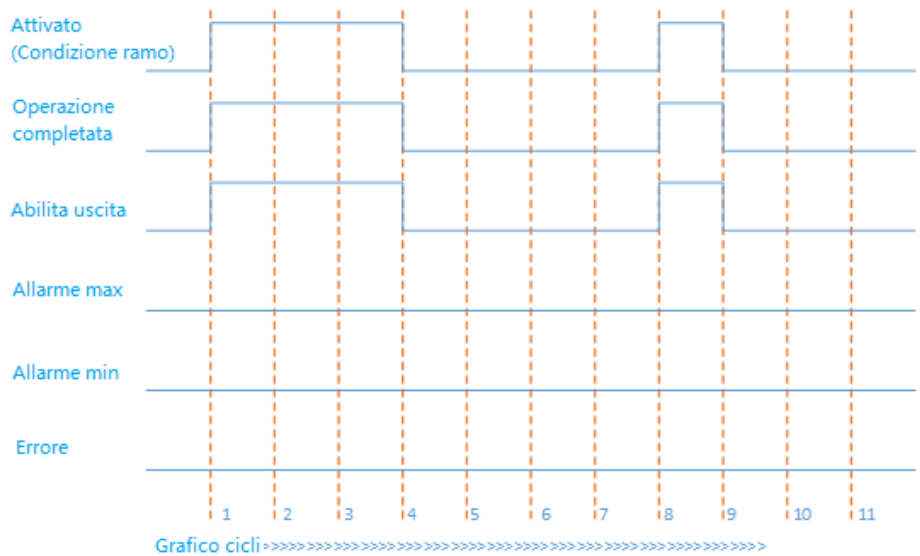
[Istruzioni di controllo dei processi](#) a pagina 503

[Diagrammi di temporizzazione dell'istruzione SCL](#) a pagina 544

Esempi di diagrammi di temporizzazione dell'istruzione SCL

Gli esempi di diagramma di temporizzazione seguenti descrivono scenari di esecuzione per l'istruzione SCL.

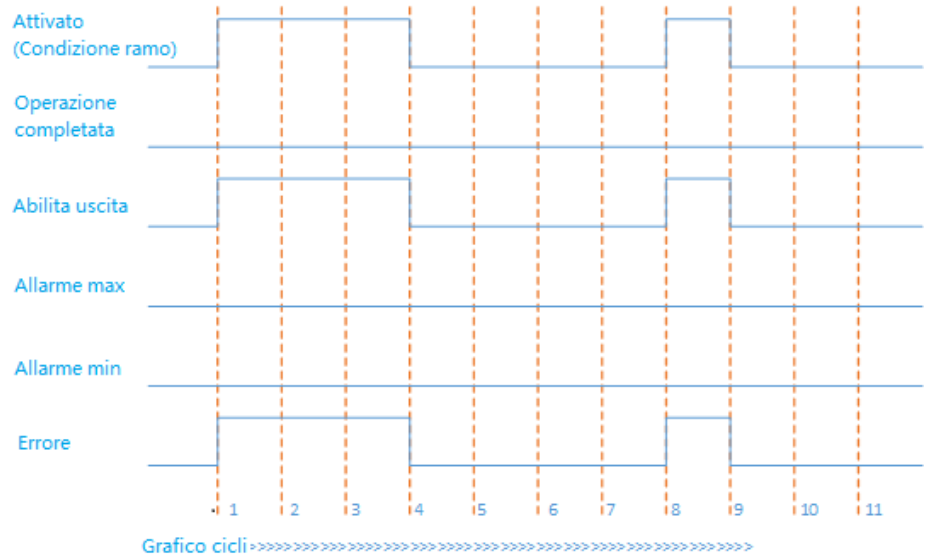
Completamento dell'esecuzione SCL



Ciclo di scansione	Descrizione
1	Se Enable è impostato su TRUE e i parametri di ingresso sono validi e compresi nell'intervallo, viene avviata l'esecuzione del blocco funzione. <ul style="list-style-type: none"> • Done ed EnableOut sono impostati su TRUE. • Out è calcolato in base agli ingressi forniti. • MaxAlarm, MinAlarm, ed Error sono impostati su FALSE. • Status ed ErrorID e Status sono impostati su 0 poiché non è stato generato alcun errore.
2, 3	Nessuna modifica nella condizione ramo.
4	Se Enable è impostato su FALSE, viene arrestata l'esecuzione del blocco funzione. <ul style="list-style-type: none"> • Done, EnableOut ed Error sono impostati su FALSE. • ErrorID è impostato su 0. • Out, MaxAlarm, MinAlarm e Status mantiene l'ultimo valore.
5, 6, 7	Nessuna modifica nella condizione ramo.
8	<ul style="list-style-type: none"> • Se Enable è impostato su TRUE e i parametri di ingresso sono validi e compresi nell'intervallo, viene avviata l'esecuzione del blocco funzione. • Done ed EnableOut sono impostati su TRUE. • Out è calcolato in base agli ingressi forniti. • MaxAlarm, MinAlarm, ed Error sono impostati su FALSE. • ErrorID e Status sono impostati su 0 poiché non è stato generato alcun errore.

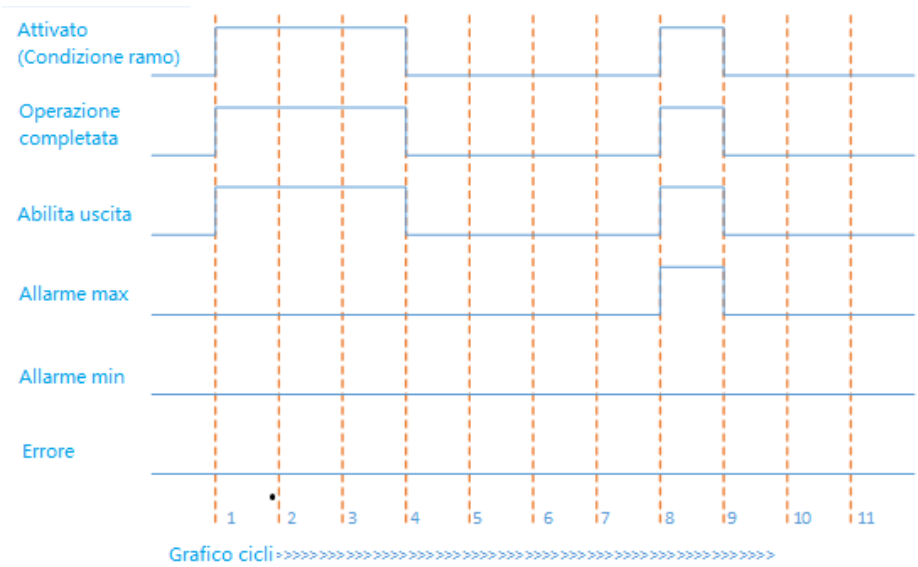
Ciclo di scansione	Descrizione
9	Se Enable è impostato su FALSE, viene arrestata l'esecuzione del blocco funzione. <ul style="list-style-type: none"> • Done, EnableOut ed Error sono impostati su FALSE. • ErrorID è impostato su 0. • Out, MaxAlarm, MinAlarm e Status mantiene l'ultimo valore.
10, 11	Nessuna modifica nella condizione ramo.

Esecuzione SCL non riuscita



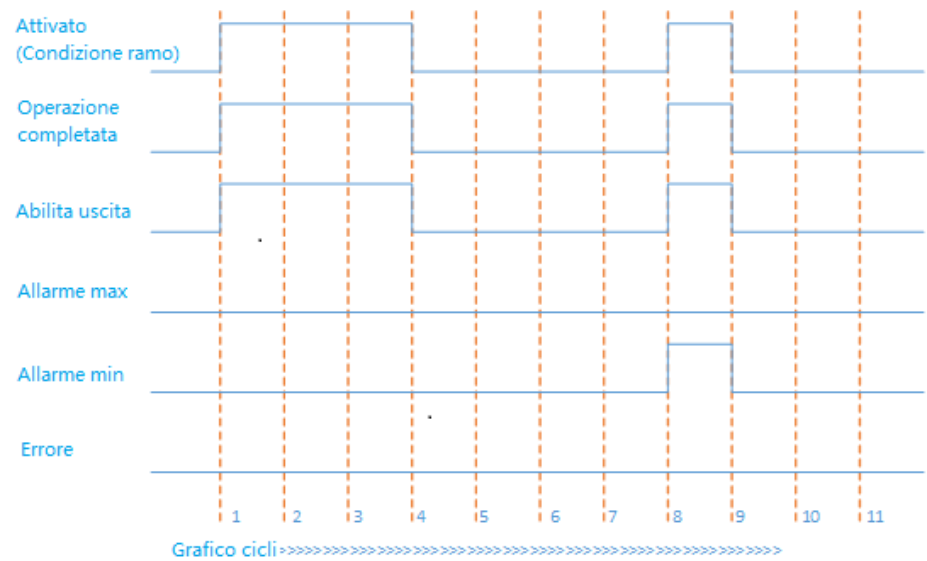
Nel seguente esempio, tutti i parametri di ingresso sono validi e compresi nell'intervallo, ma $InRawMin \geq InRawMax$. Nel ciclo di scansione 1 e 8, se Enable è impostato su TRUE e viene avviata l'esecuzione del blocco funzione, Error è impostato su True ed ErrorID è impostato su 1.

Generazione di MaxAlarm



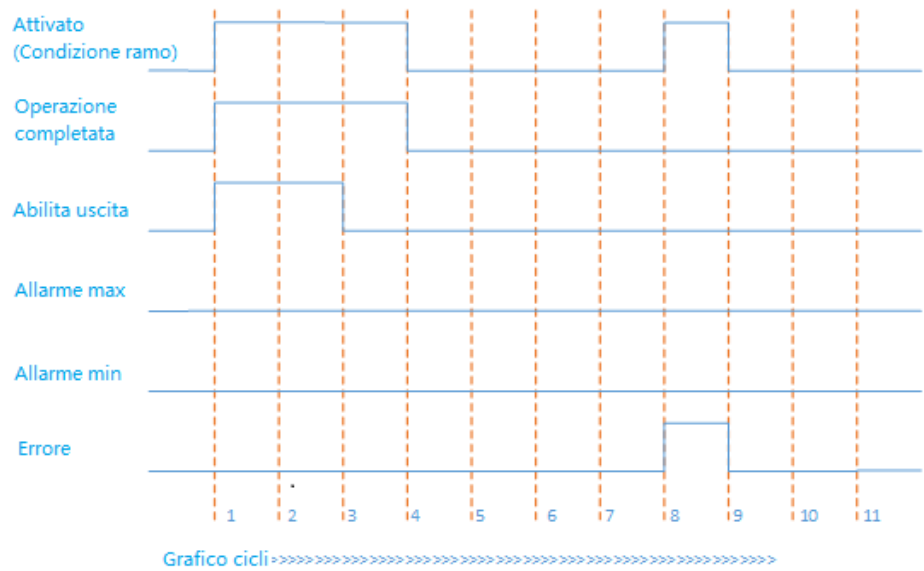
Nell'esempio del ciclo di scansione 8, tutti i parametri di ingresso sono validi e compresi nell'intervallo, ma $In > InRawMax$. Se Enable è impostato su TRUE e viene avviata l'esecuzione del blocco funzione, MaxAlarm è impostato su TRUE.

Generazione di MinAlarm



Nell'esempio del ciclo di scansione 8, tutti i parametri di ingresso sono validi e compresi nell'intervallo, ma $In < InRawMin$. Se Enable è impostato su TRUE e viene avviata l'esecuzione del blocco funzione, MinAlarm è impostato su TRUE.

Errore di configurazione ingresso e condizione di overflow uscita



Nell'esempio del ciclo di scansione 3, tutti i parametri di ingresso del blocco funzione sono validi e compresi nell'intervallo, ma si verifica un overflow dell'uscita a causa dei parametri di ingresso. EnableOut è impostato su FALSE. Il valore di Out non è valido.

Nell'esempio del ciclo di scansione 8, i parametri di ingresso del blocco funzione sono validi e rientrano nell'intervallo, ma si verifica un overflow dell'uscita a causa dei parametri di ingresso e di $InRawMin \geq InRawMax$, Error è impostato su TRUE. ErrorID è impostato su 1 e Status è impostato su 3.

Vedere anche

[SCL](#) a pagina 540

TND (interruzione programma corrente)

Arresta il ciclo di scansione del programma utente in corso Dopo la scansione dell'uscita e dell'ingresso e alcune operazioni di manutenzione, il programma dell'utente viene rieseguito dall'inizio della prima routine.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

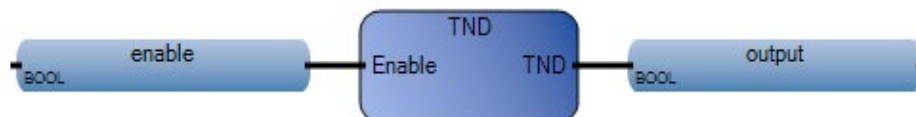
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



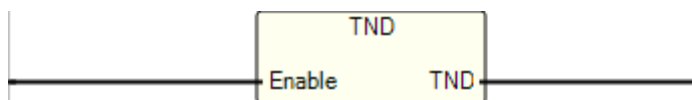
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitare funzione. Quando Enable = TRUE, esegue la funzione. Quando Enable = FALSE, non esegue la funzione.
TND	Uscita	BOOL	Se TRUE la funzione viene eseguita. se il monitoraggio delle variabili è attivo, il valore della variabile di monitoraggio viene assegnato all'uscita dell'istruzione. Se il monitoraggio delle variabili è inattivo, il valore della variabile di uscita viene assegnato all'uscita dell'istruzione.

Esempio di programmazione a blocchi funzionali TND



Esempio di diagramma ladder TND



Esempio di testo strutturato TND

```

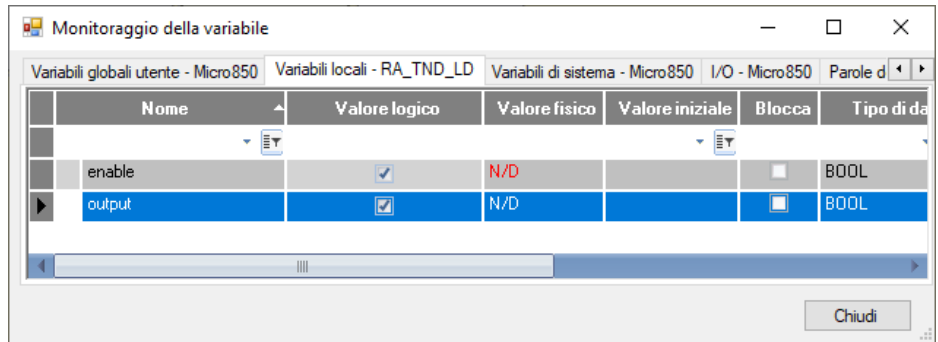
TND (
  BOOL TND(BOOL Enable)
  Interrompere la scansione del programma utente corrente.

1 | enable := TRUE;
2 | output := TND(enable);

(* Equivalenza ST: *)
TESTOUTPUT := TND(TESTENABLE);

```


Risultati



Vedere anche

[Istruzioni di controllo dei processi](#) a pagina 503

LIMIT (prova limite)

riduce i valori interi a un intervallo definito. I valori interi compresi tra il minimo e il massimo non sono modificati. I valori interi superiori al massimo sono sostituiti con il valore massimo. I valori interi inferiori al minimo sono sostituiti con il valore minimo.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

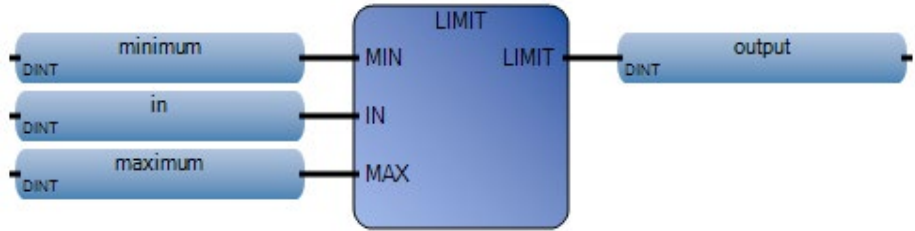
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



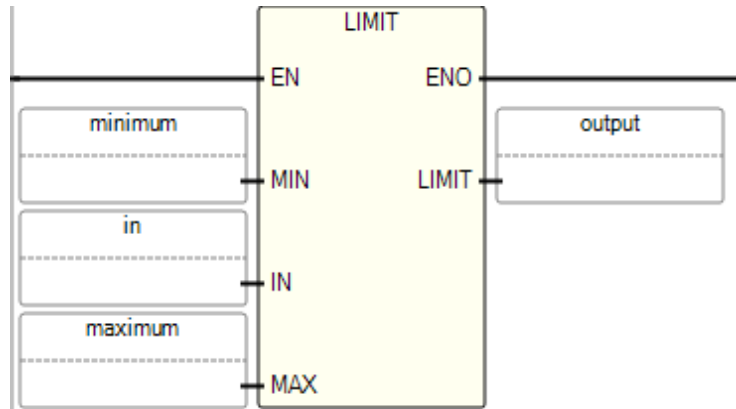
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitare funzione. TRUE: esegue il calcolo corrente LIMIT. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
MIN	Ingresso	DINT	Valore minimo supportato.
IN	Ingresso	DINT	Qualunque valore intero con segno.
MAX	Ingresso	DINT	Valore massimo supportato.
LIMIT	Uscita	DINT	Valore di ingresso vincolato all'intervallo supportato.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali LIMIT



Esempio di diagramma ladder LIMIT



Esempio di testo strutturato LIMIT

```
LIMIT (
  DINT LIMIT(DINT MIN, DINT IN, DINT MAX)
  Limite
```

```
1 minimum := 2;
2 in := 5;
3 maximum := 10;
4 output := LIMIT(minimum, in, maximum);
```

(* Equivalenza ST: *)

nuovo_valore := LIMIT (valore_min, valore, valore_max);

(* vincola il valore al gruppo [valore_min..valore_max] *)

Risultati

Monitoraggio della variabile

Variabili globali utente - Micro850 Variabili locali - RA_LIMIT_LD Variabili di sistema - Micro850 I/O - Micro850 Paro

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
in	5	N/D		<input type="checkbox"/>	DINT
maximum	10	N/D		<input type="checkbox"/>	DINT
minimum	2	N/D		<input type="checkbox"/>	DINT
output	5	N/D		<input type="checkbox"/>	DINT

Chiudi

Monitoraggio della variabile

Variabili globali utente - Micro870 Variabili locali - RA_LIMIT_LD Variabili di sistema - Micro870 I/O - Micro870

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
in		11	N/D		<input type="checkbox"/>	DINT
maximum		10	N/D		<input type="checkbox"/>	DINT
minimum		2	N/D		<input type="checkbox"/>	DINT
<input type="checkbox"/> output		10	N/D		<input type="checkbox"/>	DINT

Chiudi

Monitoraggio della variabile

Variabili globali utente - Micro870 Variabili locali - RA_LIMIT_LD Variabili di sistema - Micro870 I/O - Micro870

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
in		11	N/D		<input type="checkbox"/>	DINT
maximum		10	N/D		<input type="checkbox"/>	DINT
minimum		2	N/D		<input type="checkbox"/>	DINT
<input type="checkbox"/> output		10	N/D		<input type="checkbox"/>	DINT

Chiudi

Vedere anche

[Istruzioni di controllo dei processi](#) a pagina 503

Istruzione per il controllo del programma

Usare l'istruzione per il controllo del programma per controllare le istruzioni contemporaneamente da un programma utente e da un dispositivo dell'interfaccia operatore.

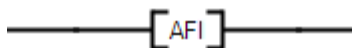
Istruzione	Descrizione
AFI a pagina 553	Disabilita un ramo.
NOP a pagina 553	Funziona come segnaposto.
SUS a pagina 554	Sospende l'esecuzione del controllore Micro800.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

AFI (sempre false)

All'inizio di un ramo, utilizzare le istruzioni AFI per disabilitare temporaneamente un ramo quando si esegue il debug senza necessità di cancellarlo dal programma. L'uscita di questa istruzione è sempre FALSE.



Linguaggi supportati: Diagramma ladder.

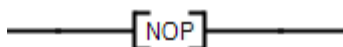
Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

Vedere anche

[Istruzione per il controllo del programma](#) a pagina 553

NOP (nessuna operazione)

L'istruzione NOP funziona come segnaposto. È possibile posizionare l'istruzione NOP in qualsiasi punto del ramo.



Linguaggi supportati: Diagramma ladder.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

Vedere anche

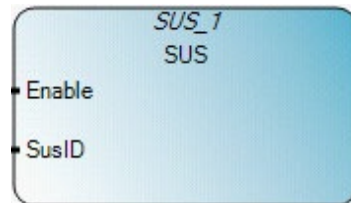
[Istruzione per il controllo del programma](#) a pagina 553

SUS (sospensione)

Sospende l'esecuzione del <controllore M800>. Il controllore resta in modalità RUN ma l'esecuzione viene sospesa indefinitamente. La sospensione serve a trovare gli errori del programma utente e aiuta nel monitoraggio del programma utente. Mettere l'istruzione SUS nelle sezioni del programma utente in cui bloccare condizioni inusuali. In modalità sospensione, RUN LED è impostato su OFF per indicare che lo scan del programma è Idle.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

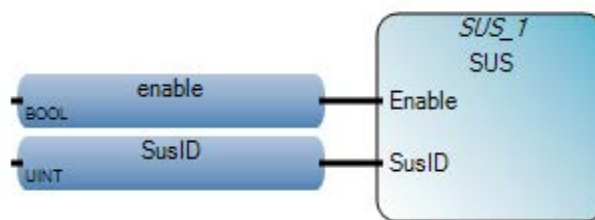
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



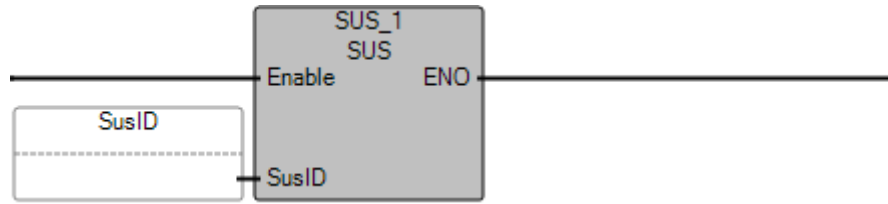
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue la funzione. FALSE: non esegue la funzione.
SusID	Ingresso	UINT	ID sospensione.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Esempio di Programmazione a blocchi funzionali SUS



Esempio di Diagramma ladder SUS



Esempio di Testo strutturato SUS

```
SUS_1 (
void SUS_1(BOOL Enable, UINT SusID)
Tipo: SUS, Sospendere l'esecuzione dell'applicazione.

1| SusID := 1;
2| SUS_1(enable, SusID);
```

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
SUS_1	<input type="checkbox"/>	SUS
enable	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
SUSID	1	N/D		<input type="checkbox"/>	UINT

Vedere anche

[Istruzione per il controllo del programma](#) a pagina 553

Istruzione proporzionale integrale derivativa (PID)

Utilizzare le istruzioni PID (Proportional-Integral-Derivative) per controllare il processo con maggiore precisione tramite la funzionalità PID.

Istruzione	Descrizione
IPIDCONTROLLER a pagina 557	Configurare e controllare gli ingressi e le uscite utilizzate per la logica proporzionale-integrale-derivativa (PID).
PID a pagina 586	Configura e controlla le uscite che controllano le proprietà fisiche come la temperatura, la pressione, il livello dei liquidi o la portata tramite cicli di processo.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

Che cos'è il controllo proporzionale integrale derivativo (PID)?

Il controllo proporzionale integrale derivativo (PID) permette al controllo di processo di mantenere in modo accurato il setpoint regolando le uscite del controllo. Un blocco funzione PID combina tutta la logica necessaria all'esecuzione del controllo proporzionale integrale derivativo (PID).

Vedere anche

[Istruzione proporzionale integrale derivativa \(PID\)](#) a pagina 557

[Operazione del blocco funzione IPIDController](#) a pagina 564

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

IPIDCONTROLLER (controllore proporzionale-integrale-derivativo)

Configurare e controllare gli ingressi e le uscite utilizzate per la logica proporzionale-integrale-derivativa (PID). La logica PID serve per controllare le proprietà fisiche quali temperatura, pressione, liquido, livello o portata tramite cicli di processo che calcolano un valore di errore come la differenza tra un setpoint desiderato e una variabile di processo misurata. Il controllore tenta di ridurre al minimo l'errore nel tempo attraverso la regolazione di una variabile di controllo. Il calcolo

include termini proporzionali (P), integrali (I) e derivativi (D) che vengono utilizzati come segue:

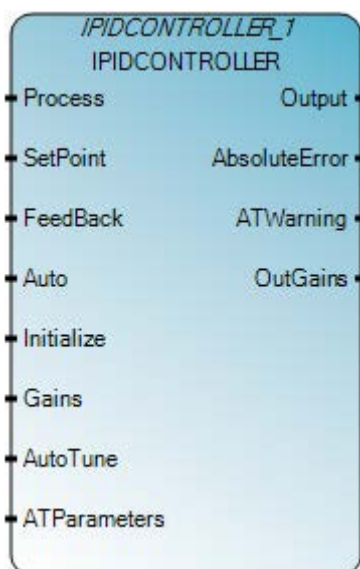
P: valori di errore presenti.

I: valori di errore passati.

D: possibili valori di errori futuri in base alla velocità corrente di modifica che controlla proprietà fisiche quali temperatura, pressione, livello dei liquidi o portata tramite cicli di processo.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

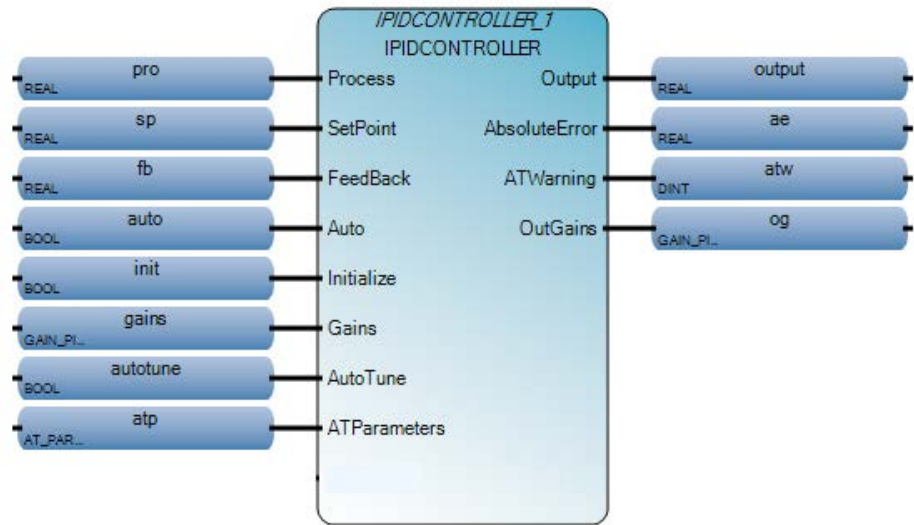


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

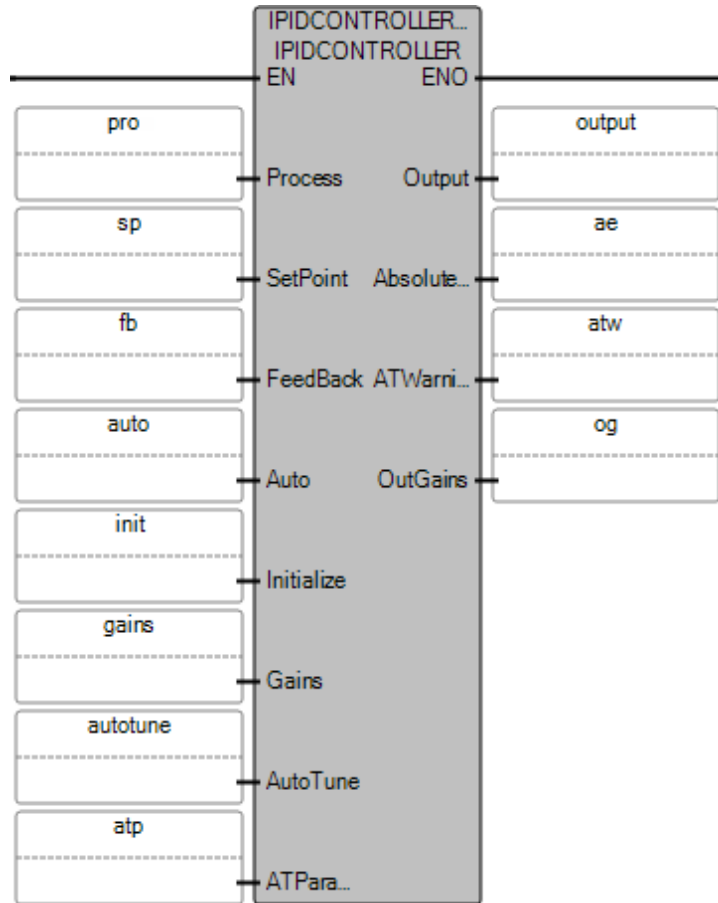
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Se TRUE, abilita il blocco istruzione. TRUE: esegue il calcolo PID. FALSE: il blocco istruzione non è attivo. Applicabile alle programmazioni in Ladder Diagram.
Process	Ingresso	REAL	Valore di processo, ovvero il valore misurato all'uscita del processo.
SetPoint	Ingresso	REAL	Set point.
Feedback	Ingresso	REAL	Segnale di feedback, ovvero il valore della variabile di controllo applicato al processo. Ad esempio, il feedback può essere l'uscita IPIDCONTROLLER.
Auto	Ingresso	BOOL	La modalità operativa del controllore PID: <ul style="list-style-type: none"> • TRUE: il controllore funziona in modalità normale. • FALSE: il controllore provoca la reimpostazione R alla traccia (F-GE).
Inizializza	Ingresso	BOOL	Una modifica del valore (da TRUE a FALSE o da FALSE a TRUE) causa l'eliminazione di qualsiasi guadagno proporzionale da parte del controllore durante quel ciclo. Inoltre, inizializza le sequenze AutoTune.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Gains	Ingresso	GAIN_PID	Guadagni PID per IPIDController. Utilizzare il tipo di dati GAIN_PID per definire i parametri per l'ingresso Gains.
AutoTune	Ingresso	BOOL	TRUE: se Auto è TRUE e Auto e Initialize sono su FALSE, la sequenza AutoTune è stata avviata. FALSE: Autotune non avviato.
ATParameters	Ingresso	AT_Param	Parametri Auto Tune. Utilizzare il tipo di dati AT_Param per definire i parametri per l'ingresso ATParameters.
Uscita	Uscita	REAL	Valore di uscita dal controllore.
AbsoluteError	Uscita	REAL	Errore assoluto (Process - SetPoint) dal controllore.
ATWarnings	Uscita	DINT	(ATWarning) Avviso per la sequenza Auto Tune. I possibili valori sono: <ul style="list-style-type: none"> • 0: nessuna sintonizzazione automatica eseguita. • 1 - In modalità auto tuning. • 2 - Auto tuning eseguito. • -1 - ERROR 1, ingresso impostato automaticamente su TRUE, nessun auto tuning possibile. • -2 - ERROR 2, errore auto tune, ATDynaSet scaduto.
OutGains	Uscita	GAIN_PID	Guadagni calcolati dopo le sequenze AutoTune. Utilizzare il tipo di dati GAIN_PID per definire l'uscita OutGains.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di Programmazione a blocchi funzionali IPIDCONTROLLER



Esempio di diagramma ladder IPIDCONTROLLER



Esempio di Testo strutturato IPIDCONTROLLER

```

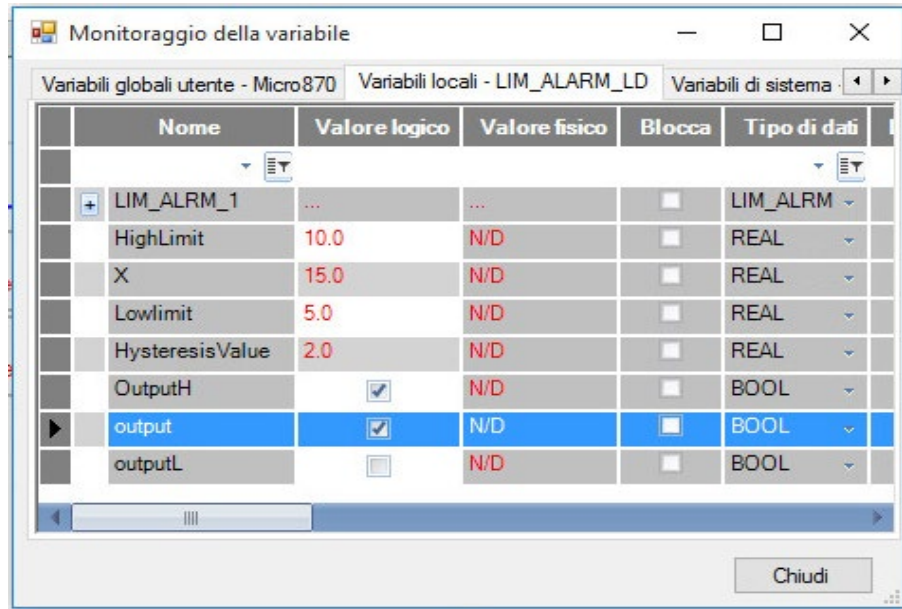
IPIDCONTROLLER_1(
void IPIDCONTROLLER_1(REAL Process, REAL SetPoint, REAL FeedBack, BOOL Auto, BOOL Initialize, GAIN_PID Gains, BOOL AutoTune, AT_PARAM ATParameters)
Tipo: IPIDCONTROLLER, Derivata integrale proporzionale.

1 IPIDCONTROLLER_1(pro, sp, fb, auto, init, gains, autotune, atp, em);
2 output := IPIDCONTROLLER_1.Output;
3 ae := IPIDCONTROLLER_1.AbsoluteError;
4 atw := IPIDCONTROLLER_1.ATWarning;
5 og := IPIDCONTROLLER_1.OutGains;

(* Equivalenza ST: IPIDController1 è un'istanza del
blocco IPIDController *)

IPIDController1(Proc,
SP,
FBK,
Auto,
Init,
G_In,
A_Tune,
A_TunePar,
Err );
Out_process := IPIDController1.Output ;
A_Tune_Warn := IPIDController1.ATWarning ;
Gain_Out := IPIDController1.OutGains ;
    
```

Risultati



Vedere anche

[Tipo di dati AT_Param](#) a pagina 562

[Tipo di dati GAIN_PID](#) a pagina 561

[Operazione del blocco funzione IPIDController](#) a pagina 564

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

[Che cos'è il controllo proporzionale integrale derivativo \(PID\)?](#) a pagina 557

Tipo di dati GAIN_PID

La tabella seguente descrive il tipo di dati GAIN_PID per l'istruzione IPIDCONTROLLER.

Parametro	Tipo di dati	Descrizione
DirectActing	BOOL	Tipo di azionamento: <ul style="list-style-type: none"> • TRUE: azionamento diretto, l'uscita si sposta nella stessa direzione dell'errore. Ovvero, il valore attuale del processo è maggiore del SetPoint e l'azione appropriata del controllore è di incrementare l'uscita, ad esempio: raffreddamento. • FALSE: azionamento inverso, l'uscita si sposta nella direzione opposta all'errore. Ovvero, il valore attuale del processo è maggiore del SetPoint e l'azione appropriata del controllore è di diminuire l'uscita, ad esempio: riscaldamento.

Parametro	Tipo di dati	Descrizione
ProportionalGain	REAL	<p>Guadagno proporzionale per PID ($\geq 0,0001$).</p> <p>Quando ProportionalGain è ($< 0,0001$), ProportionalGain = 0,0001</p> <p>Guadagno proporzionale per PID (P_Gain)</p> <p>Un guadagno proporzionale maggiore provoca una modifica maggiore nell'uscita in base alla differenza tra il PV (valore di processo misurato) e SV (valore punto preimpostato). Maggiore è il guadagno, più rapidamente l'errore diminuisce, ma ciò potrebbe comportare instabilità come oscillazioni. Minore è il guadagno, più lentamente l'errore diminuisce, ma il sistema è più stabile e meno sensibili a grandi errori. Il P_Gain è in genere il guadagno più importante per regolare e il primo guadagno da regolare durante la regolazione.</p>
TimeIntegral	REAL	<p>Valore integrale di tempo per PID in secondi ($\geq 0,0001$).</p> <p>Quando TimeIntegral è ($< 0,0001$), TimeIntegral = 0,0001</p> <p>Valore integrale di tempo per PID</p> <p>Una costante di tempo integrale minore provoca una modifica più rapida nell'uscita in base alla differenza tra il PV (valore di processo misurato) e SV (valore punto preimpostato) integrato questo periodo di tempo. Una costante di tempo integrale minore diminuisce l'errore stabile (errore quando SV non viene modificato) ma aumenta le possibilità di instabilità come oscillazioni. Una costante di tempo integrale maggiore rallenta la risposta del sistema e lo rende più stabile, ma PV si avvicina a SV a una velocità inferiore.</p>
TimeDerivative	REAL	<p>Valore derivativo di tempo per PID in secondi ($> 0,0$).</p> <p>Quando TimeDerivative è ($\leq 0,0$), TimeDerivative = 0,0</p> <p>Quando TimeDerivative è = 0, IPID agisce come PI.</p> <p>Valore derivativo di tempo per PID (Td)</p> <p>Una costante di tempo derivativa minore provoca una modifica più rapida nell'uscita basata sulla velocità di modifica della differenza tra PV (valore di processo misurato) e SV (valore punto preimpostato). Una costante di tempo derivativa minore rende un sistema più efficiente nel reagire a modifiche improvvise in caso di errore (SV viene modificato) ma aumenta le possibilità di instabilità, come oscillazioni. Una costante di tempo maggiore rende un sistema meno efficiente nel rispondere alle modifiche improvvise in caso di errore, ma il sistema è meno soggetto a modifiche nel PV dovute a rumore e passaggi. TimeDerivative (Td) è correlato al guadagno derivativo, ma consente di regolare il contributo derivativo al PID mediante l'ora in modo che il tempo di campionamento venga preso in considerazione.</p>
DerivativeGain	REAL	<p>Guadagno derivativo per PID ($> 0,0$).</p> <p>Quando DerivativeGain è ($< 0,0$), DerivativeGain = 0,1</p> <p>Guadagno derivativo per PID (D_Gain)</p> <p>Un guadagno derivativo maggiore provoca una modifica maggiore nell'uscita basata sulla velocità di modifica della differenza tra PV (valore di processo misurato) e SV (valore punto preimpostato). Un guadagno maggiore rende un sistema più efficiente nel rispondere a modifiche improvvise in caso di errore, ma aumenta le possibilità di instabilità, come oscillazioni. Un guadagno minore rende un sistema meno efficiente nel rispondere alle modifiche improvvise in caso di errore, ma rende il sistema meno soggetto a modifiche nel PV dovute a rumore e passaggi.</p>

Vedere anche

[IPIDCONTROLLER](#) a pagina 557

Tipo di dati AT_Param

Nella tabella seguente sono descritti i parametri del tipo di dati AT_Param.

Parametri	Tipo di dati	Descrizione
Load	REAL	Parametro di carico per auto tuning. Valore di uscita quando si avvia AutoTune.
Deviation	REAL	Deviazione per auto tuning. Deviazione standard usata per calcolare la noise band necessaria per AutoTune.
Passo	REAL	Valore del passo per AutoTune. Deve essere maggiore della noise band e minore di $\frac{1}{2}$ Load.
ATDynamSet	REAL	Tempo di attesa in secondi prima di abbandonare auto tune.

Parametri	Tipo di dati	Descrizione
ATReset	BOOL	Indica se il valore di uscita viene ripristinato a zero dopo una sequenza di AutoTune: <ul style="list-style-type: none"> • TRUE - Ripristina l'uscita a zero. • FALSE: lascia l'uscita sul valore Load.

Vedere anche

[IPIDCONTROLLER](#) a pagina 557

[Creazione di un ciclo di feedback per il valore manipolato](#) a pagina 579

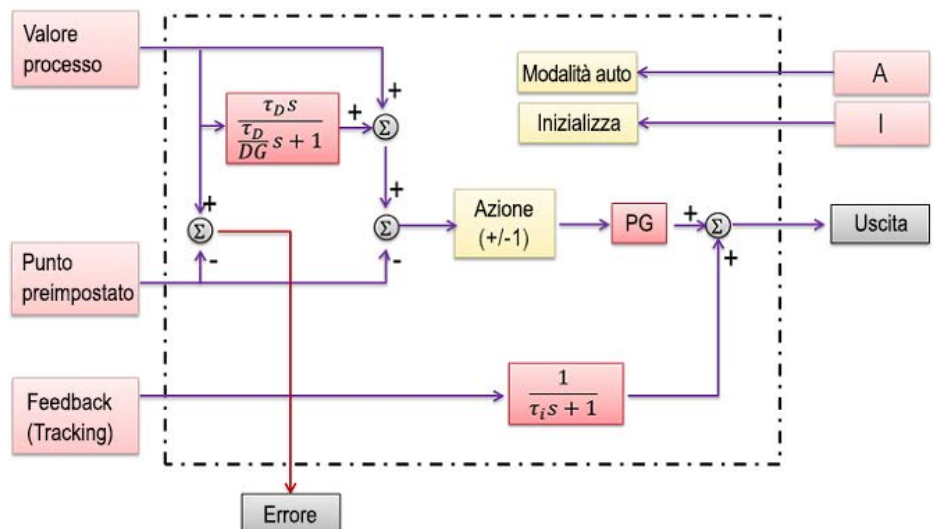
Modalità di implementazione del controllo PID da parte del blocco funzione IPIDController

Il blocco funzione IPIDController, disponibile nel set di istruzioni di Connected Components Workbench, si basa sulla teoria del controllo PID e combina tutta la logica necessaria all'esecuzione dell'elaborazione del canale di ingresso analogico e del controllo proporzionale integrale derivativo (PID). In HMI, il faceplate IPID è disponibile per l'uso con il blocco di funzione IPIDController.

Descrizione del blocco di funzione IPIDController

Il blocco di funzione IPIDController utilizza i seguenti componenti per il blocco di funzione:

- R: Azionamento (+/- 1)
- PG: Guadagno proporzionale
- DG: Guadagno filtro derivato
- td: τ_D
- ti: τ_I



Prevenzione della carica integrale

Se la differenza tra il valore del setpoint e del processo è ampia, il valore dell'uscita aumenterà in modo significativo e durante il tempo che impiega a diminuire il processo non sarà sotto controllo. Il blocco di funzione IPIDController traccia il feedback in modo interattivo e previene la carica integrale. Quando l'uscita è saturata, il termine integrale nel controllore viene rielaborato, in modo che il nuovo valore fornisca un'uscita al limite della saturazione.

Vedere anche

[Istruzione proporzionale integrale derivativa \(PID\)](#) a pagina 557

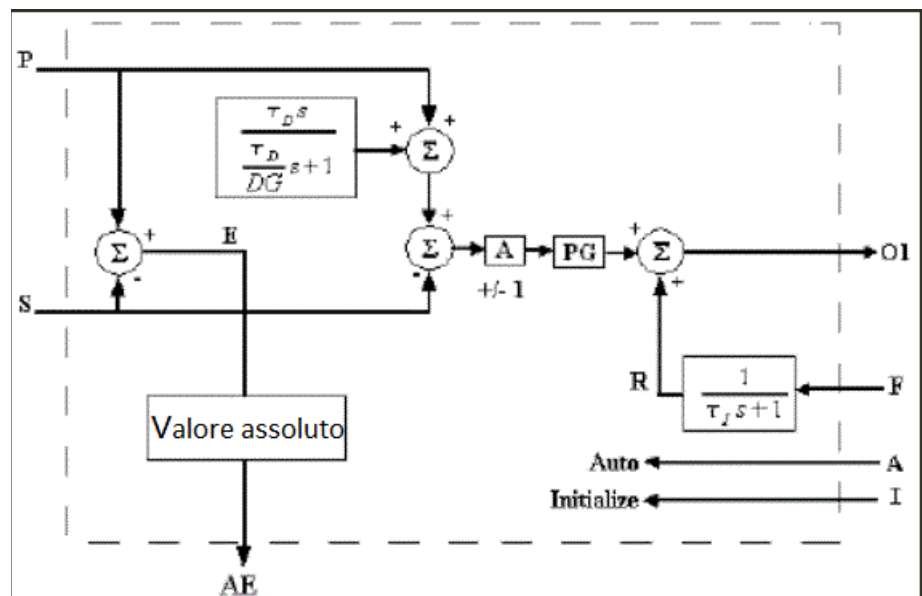
[Operazione del blocco funzione IPIDController](#) a pagina 564

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

Operazione del blocco funzione IPIDController

Quando **Input Auto** è **TRUE**, IPIDController funziona in modalità normale automatica.

Quando **Input Auto** è **FALSE**, causa il ripristino di R per il tracciamento (F-GE), forzando l'uscita IPIDController a tracciare il feedback entro i limiti di IPIDController. A questo punto il controllore ritorna in modalità automatica senza l'incremento dell'uscita.



Per il parametro di ingresso Initialize, il passaggio da FALSE a TRUE o da TRUE a FALSE quando AutoTune è FALSE, comporta che IPIDController elimini qualsiasi azione di guadagno proporzionale durante il dato ciclo (ad esempio, Initialize). Utilizzare questo processo per prevenire il bumping dell'uscita quando si effettuano modifiche al SetPoint mediante un blocco funzione dell'interruttore.

Per eseguire una sequenza AutoTune:

Per eseguire una sequenza AutoTune, ATPParameters di ingresso devono essere completati. I parametri Input Gain e DirectActing devono essere impostati in base al processo e all'impostazione di DerivativeGain (di solito 0,1). La sequenza AutoTune viene avviata con la seguente sequenza:

1. Impostare il parametro di ingresso Initialize su TRUE.
2. Impostare il parametro di ingresso Autotune su TRUE.
3. Cambiare il parametro di ingresso Initialize su FALSE.
4. Attendere finché il parametro di uscita ATWarning non passa a 2.
5. Trasferire i valori del parametro di uscita OutGains al parametro di ingresso Gains.

Per finalizzare la regolazione, potrebbe essere necessario eseguire una regolazione fine in funzione dei requisiti e dei processi. Quando si imposta TimeDerivative a 0,0, IPIDController forza DerivativeGain a 1,0 e quindi funziona come un controllore PI.

Vedere anche

[Istruzione proporzionale integrale derivativa \(PID\)](#) a pagina 557

[Che cos'è il controllo proporzionale integrale derivativo?](#) a pagina 557

[Modalità di implementazione del controllo PID da parte del blocco funzione IPIDController](#) a pagina 563

[Operazione del blocco funzione IPIDController](#) a pagina 564

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

Utilizzo dell'istruzione proporzionale integrale derivativa

Questa sezione fornisce dettagli ed esempi specifici per l'utilizzo dell'istruzione proporzionale integrale derivativa, inclusi i seguenti argomenti:

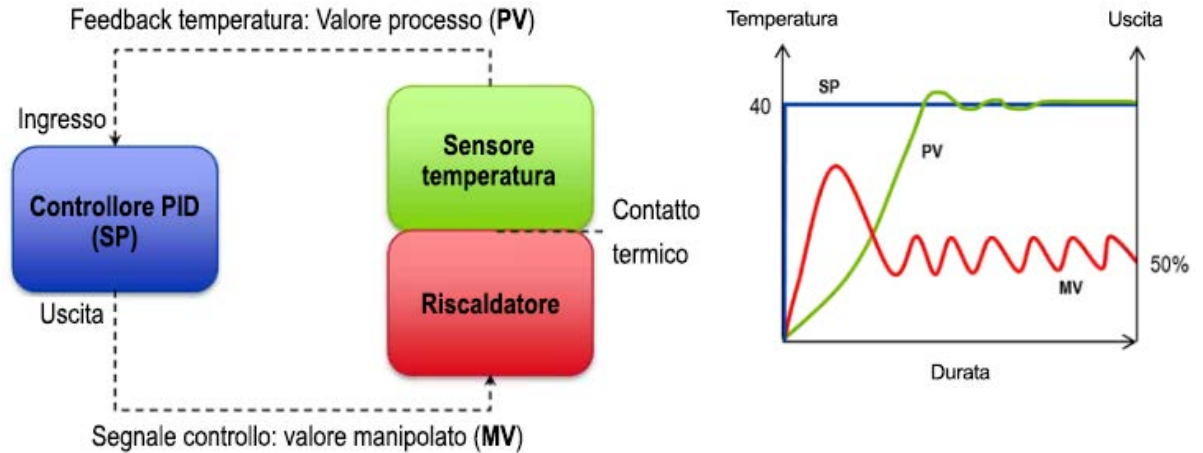
Esempio: come creare un ciclo di feedback per il valore manipolato

L'aggiunta di un ciclo di feedback per il valore manipolato previene la sovraelongazione eccessiva fornendo un valore minimo e massimo per MV.

Esempio ciclo di feedback sulla temperatura

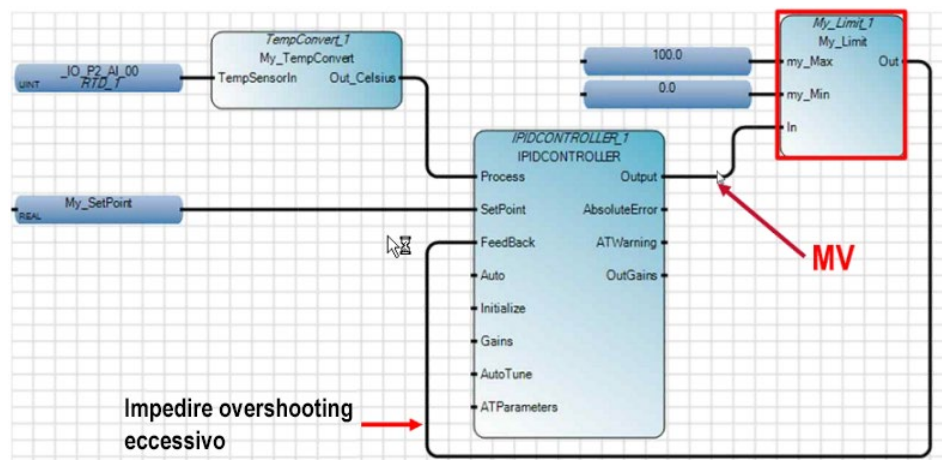
All'inizio del processo di controllo della temperatura la differenza tra il valore di processo (PV) e il valore setpoint (SP) è ampia, come mostrato nel seguente grafico. In questo esempio di ciclo di feedback sulla temperatura, PV inizia con 0 gradi Celsius e si sposta verso il

valore SP di 40 gradi Celsius. Tenere anche presente che la fluttuazione tra il valore alto e basso manipolato (MV) scende e si stabilizza nel tempo. Il comportamento di MV dipende dai valori usati in ciascuno dei parametri P, I e D.



IPIDController con ciclo di feedback

Il seguente diagramma a blocchi funzionali include un ciclo di feedback per il valore manipolato che previene la sovralongazione eccessiva fornendo un valore minimo e massimo per MV.



Esempio: come implementare l'auto tuning in un blocco funzione IPIDController

Utilizzare il parametro AutoTune del blocco funzione IPIDController per implementare l'auto tuning nel programma di controllo.

Requisiti e raccomandazioni per l'auto tuning

Di seguito è riportato un riepilogo dei requisiti e delle raccomandazioni per la corretta implementazione dell'auto tuning.

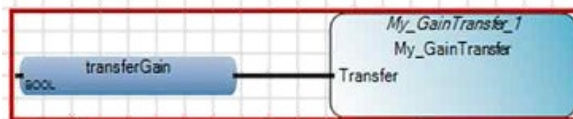
- L'auto tuning deve causare l'oscillazione dell'uscita del ciclo di controllo, vale a dire che l'IPIDController deve essere richiamato con una certa frequenza per una campionatura adeguata dell'oscillazione.
- Il blocco funzione IPIDController deve essere eseguito con un intervallo relativamente costante.
- Configurare il tempo di scansione del programma alla metà dell'intervallo di oscillazione.
- Considerare l'utilizzo di un blocco di istruzioni Interruzione testo strutturato (STI) per controllare il blocco di istruzioni IPIDController.

Esempio: come aggiungere un UDFB a un programma PID

Aggiungere UDFB esternamente al programma principale per eseguire funzioni specializzate, come conversione di unità o trasferimento di valori.

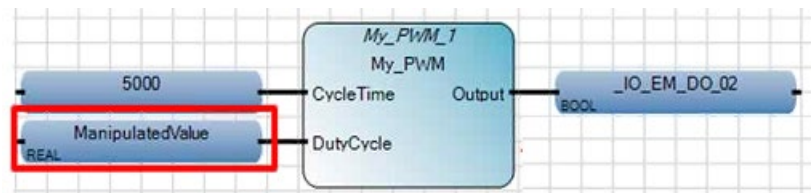
Trasferimento del valore di guadagno con auto tuning

Questo UDFB trasferisce il valore del guadagno AutoTune a My_GainTransfer per l'utilizzo da parte del controllore.



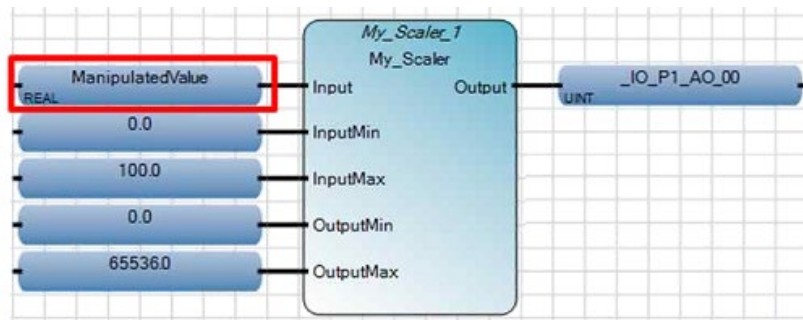
Conversione di un valore manipolato in un'uscita digitale

Questo UDFB converte un valore manipolato (MV) in un'uscita digitale (DO) in modo che sia possibile usarlo per controllare un ingresso digitale n(DI).



Conversione di un valore manipolato in un'uscita analogica

Questo UDFB converte un valore manipolato (MV) in un'uscita analogica (AO) in modo che sia possibile usarlo per controllare un ingresso analogico (AI).



Vedere anche

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

[Istruzione proporzionale integrale derivativa \(PID\)](#) a pagina 557

[Che cos'è il controllo proporzionale integrale derivativo?](#) a pagina 557

[Operazione del blocco funzione IPIDController](#) a pagina 564

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

Utilizzo dell'auto tuning con il blocco funzione IPIDController

Utilizzare il parametro AutoTune del blocco funzione IPIDController per implementare l'auto tuning nel programma di controllo.

Requisiti e raccomandazioni per l'auto tuning

Di seguito è riportato un riepilogo dei requisiti e delle raccomandazioni per la corretta implementazione dell'auto tuning.

- L'auto tuning deve causare l'oscillazione dell'uscita del ciclo di controllo, vale a dire che l'IPIDController deve essere richiamato con una certa frequenza per una campionatura adeguata dell'oscillazione.
- Il blocco funzione IPIDController deve essere eseguito con un intervallo relativamente costante.
- Configurare il tempo di scansione del programma alla metà dell'intervallo di oscillazione.
- Considerare l'utilizzo di un blocco istruzione STI (Selectable Timed Interrupt) per controllare il blocco istruzione IPIDController.

Vedere anche

[Istruzione proporzionale integrale derivativa \(PID\)](#) a pagina 557

[Che cos'è il controllo proporzionale integrale derivativo?](#) a pagina 557

[Operazione del blocco funzione IPIDController](#) a pagina 564

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

Auto tuning in sistemi di primo e secondo ordine

Utilizzare l'auto tuning in un sistema di primo ordine in cui è utilizzato un solo elemento oppure in un sistema di secondo ordine in cui sono utilizzati due elementi indipendenti.

Un sistema di primo ordine prevede un solo elemento indipendente di accumulazione di energia. Gli esempi includono:

- Raffreddamento di un serbatoio per fluidi, con accumulazione di energia termica.
- Flusso o fluido da un serbatoio, con accumulazione di energia potenziale.
- Un motore a coppia costante che controlla un volano, con accumulazione di energia cinetica rotazionale.
- Una rete elettrica RC ad anticipo di fase, con accumulazione di energia capacitiva.

In un sistema di primo ordine la funzione può essere scritta in forma standard, come $f(t) = \tau \frac{dy}{dt} + y(t)$

Dove:

Variabile	Descrizione	Esempio: Raffreddamento di un serbatoio per fluidi utilizzando l'energia termica come elemento di accumulazione
t	Costante di tempo del sistema	È uguale a RC Dove R = resistenza termica delle pareti del serbatoio C = capacità termica del fluido
f	Forzatura della funzione	È la temperatura ambiente
y	Variabile di stato del sistema	È la temperatura del fluido

Un sistema di secondo ordine prevede due elementi per l'accumulazione di energia che scambiano l'energia accumulata. Gli esempi includono:

- Un motore che comanda un volano disco con motore accoppiato al volano mediante un albero con rigidità torsionale; energia cinetica rotazionale ed energia della molla torsionale sono le unità di memorizzazione.
- Un circuito elettrico composto da un generatore di corrente che alimenta un induttore e un resistore in serie (LR) con un condensatore (C) in derivazione. Vengono accumulate l'energia induttiva e l'energia capacitiva.

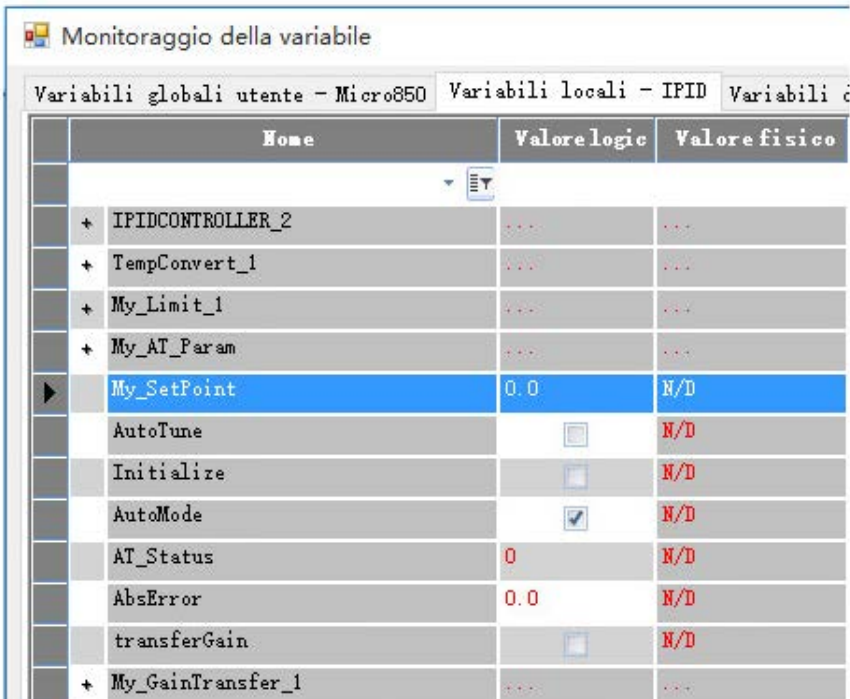
I sistemi azionati a motore e i sistemi di riscaldamento di norma possono essere modellati in base a un circuito elettrico LR e C.

Vedere anche

[Utilizzo dell'auto tuning con il blocco funzione IPIDController](#) a pagina 568

Configurazione dell'auto tuning

Seguire questa procedura generale per l'implementazione dell'auto tuning utilizzando la funzione IPIDController.

N.	Passo	Esempio																																							
1	Ripristinare il setpoint su zero.	 <table border="1"> <thead> <tr> <th>Nome</th> <th>Valore logico</th> <th>Valore fisico</th> </tr> </thead> <tbody> <tr> <td>IPIDCONTROLLER_2</td> <td>...</td> <td>...</td> </tr> <tr> <td>TempConvert_1</td> <td>...</td> <td>...</td> </tr> <tr> <td>My_Limit_1</td> <td>...</td> <td>...</td> </tr> <tr> <td>My_AT_Param</td> <td>...</td> <td>...</td> </tr> <tr> <td>My_SetPoint</td> <td>0.0</td> <td>N/D</td> </tr> <tr> <td>AutoTune</td> <td><input type="checkbox"/></td> <td>N/D</td> </tr> <tr> <td>Initialize</td> <td><input type="checkbox"/></td> <td>N/D</td> </tr> <tr> <td>AutoMode</td> <td><input checked="" type="checkbox"/></td> <td>N/D</td> </tr> <tr> <td>AT_Status</td> <td>0</td> <td>N/D</td> </tr> <tr> <td>AbsError</td> <td>0.0</td> <td>N/D</td> </tr> <tr> <td>transferGain</td> <td><input type="checkbox"/></td> <td>N/D</td> </tr> <tr> <td>My_GainTransfer_1</td> <td>...</td> <td>...</td> </tr> </tbody> </table>	Nome	Valore logico	Valore fisico	IPIDCONTROLLER_2	TempConvert_1	My_Limit_1	My_AT_Param	My_SetPoint	0.0	N/D	AutoTune	<input type="checkbox"/>	N/D	Initialize	<input type="checkbox"/>	N/D	AutoMode	<input checked="" type="checkbox"/>	N/D	AT_Status	0	N/D	AbsError	0.0	N/D	transferGain	<input type="checkbox"/>	N/D	My_GainTransfer_1
Nome	Valore logico	Valore fisico																																							
IPIDCONTROLLER_2																																							
TempConvert_1																																							
My_Limit_1																																							
My_AT_Param																																							
My_SetPoint	0.0	N/D																																							
AutoTune	<input type="checkbox"/>	N/D																																							
Initialize	<input type="checkbox"/>	N/D																																							
AutoMode	<input checked="" type="checkbox"/>	N/D																																							
AT_Status	0	N/D																																							
AbsError	0.0	N/D																																							
transferGain	<input type="checkbox"/>	N/D																																							
My_GainTransfer_1																																							

N. 2 Passo
Impostare la modalità automatica su False.

Esempio

Monitoraggio della variabile

Variabili globali utente - Micro850 Variabili locali - IPID Variabili di sistema

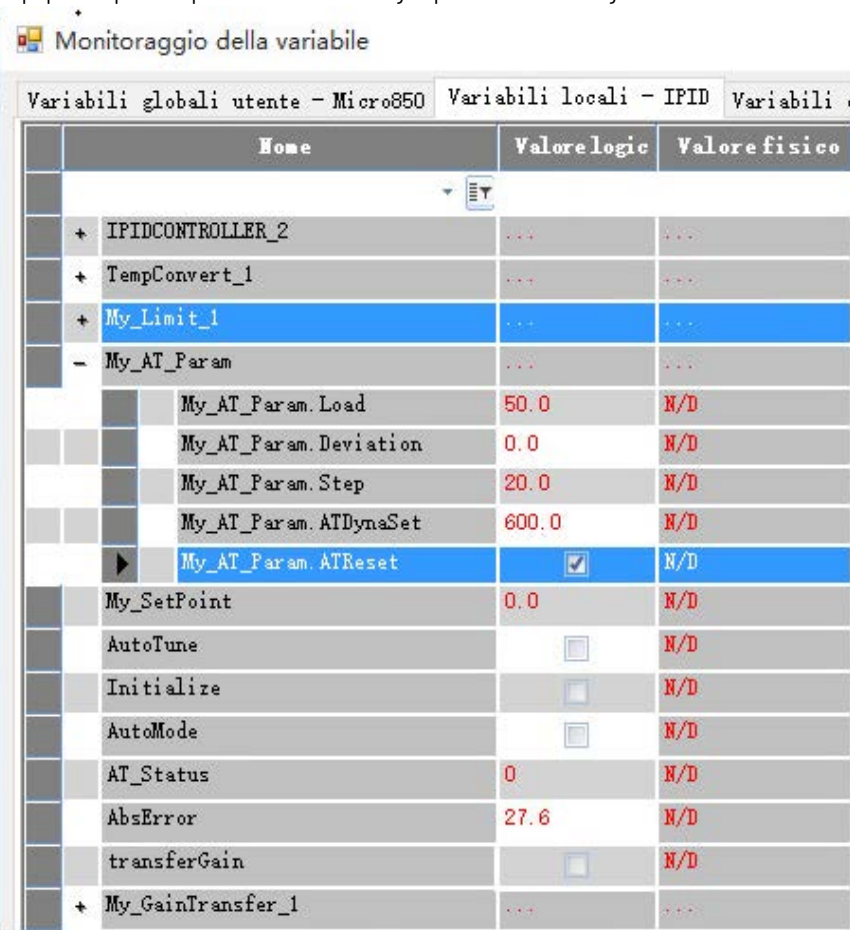
Nome	Valore logico	Valore fisico
+ IPIDCONTROLLER_2
+ TempConvert_1
+ My_Limit_1
+ My_AT_Param
▶ My_SetPoint	0.0	N/D
AutoTune	<input type="checkbox"/>	N/D
Initialize	<input type="checkbox"/>	N/D
AutoMode	<input type="checkbox"/>	N/D
AT_Status	0	N/D
AbsError	29.8	N/D
transferGain	<input type="checkbox"/>	N/D
+ My_GainTransfer_1

3 Impostare i parametri Gains.

Monitoraggio della variabile

Variabili globali utente - Micro850 Variabili locali - N/D Variabili di sistema

Nome	Valore logico	Valore fisico
- My_Gains
My_Gains.DirectActing	<input type="checkbox"/>	N/D
My_Gains.ProportionalGain	20.0	N/D
My_Gains.TimeIntegral	0.0001	N/D
My_Gains.TimeDerivative	0.0	N/D
My_Gains.DerivativeGain	0.1	N/D
ManipulatedValue	0.0	N/D
▶ - OutGains
OutGains.DirectActing	<input type="checkbox"/>	N/D
OutGains.ProportionalGain	0.0	N/D
OutGains.TimeIntegral	0.0	N/D
OutGains.TimeDerivative	0.0	N/D
OutGains.DerivativeGain	0.0	N/D

N.	Passo	Esempio																																																																																																																		
4	Impostare i parametri di auto tuning.	<p>Impostare i parametri di auto tuning, compresi il valore di carico iniziale, la variazione del passo dell'uscita, il tempo previsto per il completamento dell'auto tuning e ripristino dell'auto tuning.</p>  <p>Monitoraggio della variabile</p> <table border="1"> <thead> <tr> <th colspan="3">Variabili globali utente - Micro850</th> <th colspan="2">Variabili locali - IPID</th> <th>Variabili c</th> </tr> <tr> <th></th> <th>Nome</th> <th></th> <th>Valore logico</th> <th>Valore fisico</th> <th></th> </tr> </thead> <tbody> <tr> <td>+</td> <td>IPIDCONTROLLER_2</td> <td></td> <td>...</td> <td>...</td> <td></td> </tr> <tr> <td>+</td> <td>TempConvert_1</td> <td></td> <td>...</td> <td>...</td> <td></td> </tr> <tr> <td>+</td> <td>My_Limit_1</td> <td></td> <td>...</td> <td>...</td> <td></td> </tr> <tr> <td>-</td> <td>My_AT_Param</td> <td></td> <td>...</td> <td>...</td> <td></td> </tr> <tr> <td></td> <td>My_AT_Param.Load</td> <td></td> <td>50.0</td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>My_AT_Param.Deviation</td> <td></td> <td>0.0</td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>My_AT_Param.Step</td> <td></td> <td>20.0</td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>My_AT_Param.ATDynaSet</td> <td></td> <td>600.0</td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>My_AT_Param.ATReset</td> <td><input checked="" type="checkbox"/></td> <td></td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>My_SetPoint</td> <td></td> <td>0.0</td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>AutoTune</td> <td><input type="checkbox"/></td> <td></td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>Initialize</td> <td><input type="checkbox"/></td> <td></td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>AutoMode</td> <td><input type="checkbox"/></td> <td></td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>AT_Status</td> <td></td> <td>0</td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>AbsError</td> <td></td> <td>27.6</td> <td>N/D</td> <td></td> </tr> <tr> <td></td> <td>transferGain</td> <td><input type="checkbox"/></td> <td></td> <td>N/D</td> <td></td> </tr> <tr> <td>+</td> <td>My_GainTransfer_1</td> <td></td> <td>...</td> <td>...</td> <td></td> </tr> </tbody> </table>	Variabili globali utente - Micro850			Variabili locali - IPID		Variabili c		Nome		Valore logico	Valore fisico		+	IPIDCONTROLLER_2			+	TempConvert_1			+	My_Limit_1			-	My_AT_Param				My_AT_Param.Load		50.0	N/D			My_AT_Param.Deviation		0.0	N/D			My_AT_Param.Step		20.0	N/D			My_AT_Param.ATDynaSet		600.0	N/D			My_AT_Param.ATReset	<input checked="" type="checkbox"/>		N/D			My_SetPoint		0.0	N/D			AutoTune	<input type="checkbox"/>		N/D			Initialize	<input type="checkbox"/>		N/D			AutoMode	<input type="checkbox"/>		N/D			AT_Status		0	N/D			AbsError		27.6	N/D			transferGain	<input type="checkbox"/>		N/D		+	My_GainTransfer_1		
Variabili globali utente - Micro850			Variabili locali - IPID		Variabili c																																																																																																															
	Nome		Valore logico	Valore fisico																																																																																																																
+	IPIDCONTROLLER_2																																																																																																																	
+	TempConvert_1																																																																																																																	
+	My_Limit_1																																																																																																																	
-	My_AT_Param																																																																																																																	
	My_AT_Param.Load		50.0	N/D																																																																																																																
	My_AT_Param.Deviation		0.0	N/D																																																																																																																
	My_AT_Param.Step		20.0	N/D																																																																																																																
	My_AT_Param.ATDynaSet		600.0	N/D																																																																																																																
	My_AT_Param.ATReset	<input checked="" type="checkbox"/>		N/D																																																																																																																
	My_SetPoint		0.0	N/D																																																																																																																
	AutoTune	<input type="checkbox"/>		N/D																																																																																																																
	Initialize	<input type="checkbox"/>		N/D																																																																																																																
	AutoMode	<input type="checkbox"/>		N/D																																																																																																																
	AT_Status		0	N/D																																																																																																																
	AbsError		27.6	N/D																																																																																																																
	transferGain	<input type="checkbox"/>		N/D																																																																																																																
+	My_GainTransfer_1																																																																																																																	

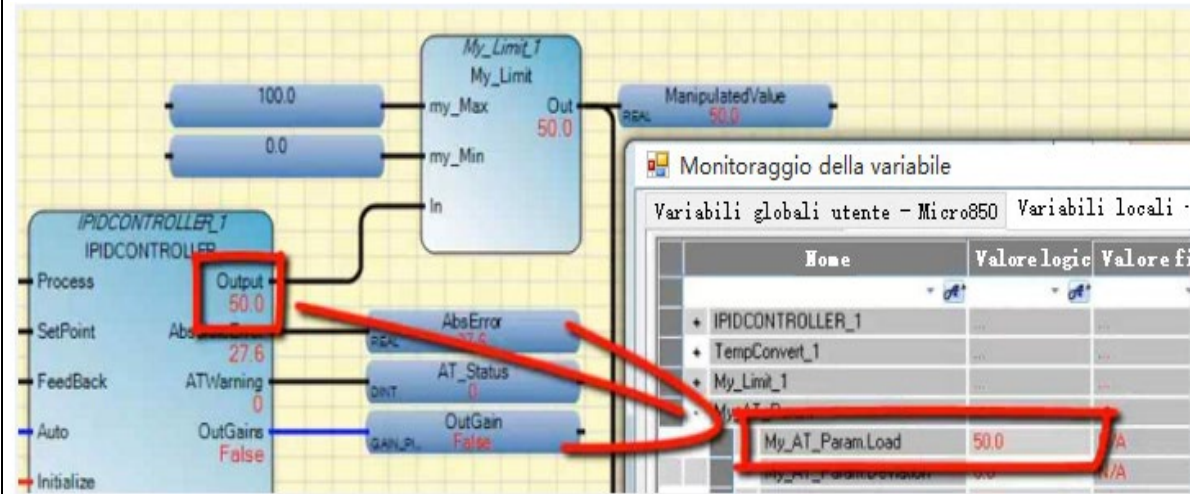
N. 5 **Passo**
 Impostare i parametri Initialize e Auto-Tune su True.

Esempio

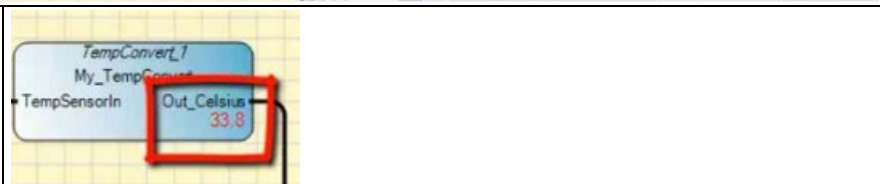
Monitoraggio della variabile

Home	Valore logico	Valore fisico
+ IPIDCONTROLLER_2
+ TempConvert_1
+ My_Limit_1
- My_AT_Param
My_AT_Param.Load	50.0	N/D
My_AT_Param.Deviation	0.0	N/D
My_AT_Param.Step	20.0	N/D
My_AT_Param.ATDynaSet	600.0	N/D
My_AT_Param.ATReset	<input checked="" type="checkbox"/>	N/D
My_SetPoint	0.0	N/D
▶ AutoTune	<input checked="" type="checkbox"/>	N/D
Initialize	<input checked="" type="checkbox"/>	N/D
AutoMode	<input type="checkbox"/>	N/D
AT_Status	0	N/D
AbsError	27.6	N/D
transferGain	<input type="checkbox"/>	N/D
+ My_GainTransfer_1

6 Notare che l'uscita viene impostata sul valore di carico quando si imposta il parametro Auto-Tune su True.



7 Osservare l'incremento rapido del valore di processo, finché non si avvicina al punto di saturazione.



8 Osservare la stabilizzazione del valore di processo e la sua fluttuazione.



N. 9 Passo
Impostare la deviazione.

Esempio

Monitoraggio della variabile

Variabili globali utente - Micro850 Variabili locali - IPID Variabili c

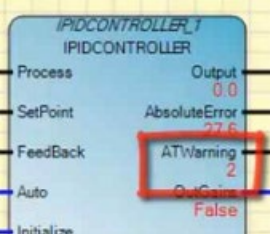
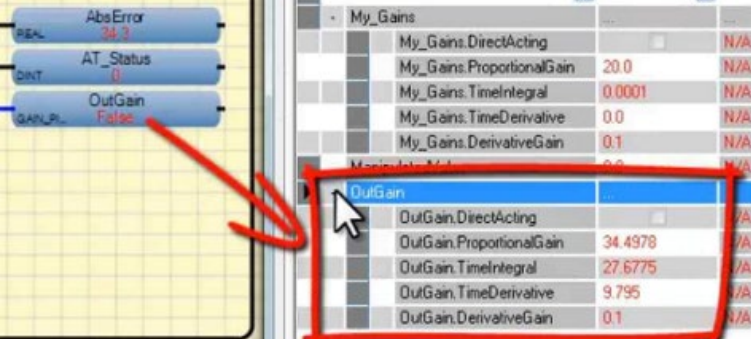
Nome	Valore logico	Valore fisico
+ IPIDCONTROLLER_2
+ TempConvert_1
+ My_Limit_1
- My_AT_Param
My_AT_Param.Load	50.0	N/D
My_AT_Param.Deviation	0.05	N/D
My_AT_Param.Step	20.0	N/D
My_AT_Param.ATDynaSet	600.0	N/D
My_AT_Param.ATReset	<input checked="" type="checkbox"/>	N/D
My_SetPoint	0.0	N/D
AutoTune	<input type="checkbox"/>	N/D
Initialize	<input type="checkbox"/>	N/D
AutoMode	<input type="checkbox"/>	N/D
AT_Status	0	N/D
AbsError	27.6	N/D
transferGain	<input type="checkbox"/>	N/D
+ My_GainTransfer_1

10 Impostare Initialize su False.

Monitoraggio della variabile

Variabili globali utente - Micro850 Variabili locali - IPID Variabili c

Nome	Valore logico	Valore fisico
+ IPIDCONTROLLER_2
+ TempConvert_1
+ My_Limit_1
- My_AT_Param
My_AT_Param.Load	50.0	N/D
My_AT_Param.Deviation	0.05	N/D
My_AT_Param.Step	20.0	N/D
My_AT_Param.ATDynaSet	600.0	N/D
My_AT_Param.ATReset	<input checked="" type="checkbox"/>	N/D
My_SetPoint	0.0	N/D
AutoTune	<input type="checkbox"/>	N/D
Initialize	<input checked="" type="checkbox"/>	N/D
AutoMode	<input type="checkbox"/>	N/D

<p>N. 11</p>	<p>Passo Il controllore avvia l'auto tuning. Attendere l'impostazione di ATWarning su 2.</p>	<p>Esempio</p> 																																										
<p>12</p>	<p>Impostare AutoTune su False.</p>	<p>Monitoraggio della variabile</p> <p>Variabili globali utente - Micro850 Variabili locali - IPID Variabili d</p> <table border="1"> <thead> <tr> <th>Nome</th> <th>Valore logico</th> <th>Valore fisico</th> </tr> </thead> <tbody> <tr> <td>+ IPIDCONTROLLER_2</td> <td>....</td> <td>....</td> </tr> <tr> <td>+ TempConvert_1</td> <td>....</td> <td>....</td> </tr> <tr> <td>+ My_Limit_1</td> <td>....</td> <td>....</td> </tr> <tr> <td>- My_AT_Param</td> <td>....</td> <td>....</td> </tr> <tr> <td> My_AT_Param.Load</td> <td>50.0</td> <td>N/D</td> </tr> <tr> <td> My_AT_Param.Deviation</td> <td>0.05</td> <td>N/D</td> </tr> <tr> <td> My_AT_Param.Step</td> <td>20.0</td> <td>N/D</td> </tr> <tr> <td> My_AT_Param.ATDynaSet</td> <td>600.0</td> <td>N/D</td> </tr> <tr> <td> My_AT_Param.ATReset</td> <td><input checked="" type="checkbox"/></td> <td>N/D</td> </tr> <tr> <td>My_SetPoint</td> <td>0.0</td> <td>N/D</td> </tr> <tr> <td>AutoTune</td> <td><input type="checkbox"/></td> <td>N/D</td> </tr> <tr> <td>Initialize</td> <td><input checked="" type="checkbox"/></td> <td>N/D</td> </tr> </tbody> </table>	Nome	Valore logico	Valore fisico	+ IPIDCONTROLLER_2	+ TempConvert_1	+ My_Limit_1	- My_AT_Param	My_AT_Param.Load	50.0	N/D	My_AT_Param.Deviation	0.05	N/D	My_AT_Param.Step	20.0	N/D	My_AT_Param.ATDynaSet	600.0	N/D	My_AT_Param.ATReset	<input checked="" type="checkbox"/>	N/D	My_SetPoint	0.0	N/D	AutoTune	<input type="checkbox"/>	N/D	Initialize	<input checked="" type="checkbox"/>	N/D			
Nome	Valore logico	Valore fisico																																										
+ IPIDCONTROLLER_2																																										
+ TempConvert_1																																										
+ My_Limit_1																																										
- My_AT_Param																																										
My_AT_Param.Load	50.0	N/D																																										
My_AT_Param.Deviation	0.05	N/D																																										
My_AT_Param.Step	20.0	N/D																																										
My_AT_Param.ATDynaSet	600.0	N/D																																										
My_AT_Param.ATReset	<input checked="" type="checkbox"/>	N/D																																										
My_SetPoint	0.0	N/D																																										
AutoTune	<input type="checkbox"/>	N/D																																										
Initialize	<input checked="" type="checkbox"/>	N/D																																										
<p>13</p>	<p>Osservare i valori presenti in OutGains dopo l'auto tuning.</p>	 <table border="1"> <thead> <tr> <th>Nome</th> <th>Valore logico</th> <th>Valore fisico</th> </tr> </thead> <tbody> <tr> <td>- My_Gains</td> <td>....</td> <td>....</td> </tr> <tr> <td> My_Gains.DirectActing</td> <td><input type="checkbox"/></td> <td>N/A</td> </tr> <tr> <td> My_Gains.ProportionalGain</td> <td>20.0</td> <td>N/A</td> </tr> <tr> <td> My_Gains.TimeIntegral</td> <td>0.0001</td> <td>N/A</td> </tr> <tr> <td> My_Gains.TimeDerivative</td> <td>0.0</td> <td>N/A</td> </tr> <tr> <td> My_Gains.DerivativeGain</td> <td>0.1</td> <td>N/A</td> </tr> <tr> <td> My_Gains.M...</td> <td>0.0</td> <td>N/A</td> </tr> <tr> <td>OutGain</td> <td>....</td> <td>....</td> </tr> <tr> <td> OutGain.DirectActing</td> <td><input type="checkbox"/></td> <td>N/A</td> </tr> <tr> <td> OutGain.ProportionalGain</td> <td>34.4978</td> <td>N/A</td> </tr> <tr> <td> OutGain.TimeIntegral</td> <td>27.6775</td> <td>N/A</td> </tr> <tr> <td> OutGain.TimeDerivative</td> <td>9.795</td> <td>N/A</td> </tr> <tr> <td> OutGain.DerivativeGain</td> <td>0.1</td> <td>N/A</td> </tr> </tbody> </table>	Nome	Valore logico	Valore fisico	- My_Gains	My_Gains.DirectActing	<input type="checkbox"/>	N/A	My_Gains.ProportionalGain	20.0	N/A	My_Gains.TimeIntegral	0.0001	N/A	My_Gains.TimeDerivative	0.0	N/A	My_Gains.DerivativeGain	0.1	N/A	My_Gains.M...	0.0	N/A	OutGain	OutGain.DirectActing	<input type="checkbox"/>	N/A	OutGain.ProportionalGain	34.4978	N/A	OutGain.TimeIntegral	27.6775	N/A	OutGain.TimeDerivative	9.795	N/A	OutGain.DerivativeGain	0.1	N/A
Nome	Valore logico	Valore fisico																																										
- My_Gains																																										
My_Gains.DirectActing	<input type="checkbox"/>	N/A																																										
My_Gains.ProportionalGain	20.0	N/A																																										
My_Gains.TimeIntegral	0.0001	N/A																																										
My_Gains.TimeDerivative	0.0	N/A																																										
My_Gains.DerivativeGain	0.1	N/A																																										
My_Gains.M...	0.0	N/A																																										
OutGain																																										
OutGain.DirectActing	<input type="checkbox"/>	N/A																																										
OutGain.ProportionalGain	34.4978	N/A																																										
OutGain.TimeIntegral	27.6775	N/A																																										
OutGain.TimeDerivative	9.795	N/A																																										
OutGain.DerivativeGain	0.1	N/A																																										

N. 14 **Passo**
 Trasferire il parametro da OutGain a My_Gains.

Esempio

Monitoraggio della variabile

Variabili globali utente - Micro850 Variabili locali - IPID Variabili

Nome	Valore logico	Valore fisico
+ IPIDCONTROLLER_2
+ TempConvert_1
+ My_Limit_1
- My_AT_Param
My_AT_Param.Load	50.0	N/D
My_AT_Param.Deviation	0.05	N/D
My_AT_Param.Step	20.0	N/D
My_AT_Param.ATDynaSet	600.0	N/D
My_AT_Param.ATReset	<input checked="" type="checkbox"/>	N/D
My_SetPoint	0.0	N/D
▶ AutoTune	<input type="checkbox"/>	N/D
Initialize	<input type="checkbox"/>	N/D
AutoMode	<input type="checkbox"/>	N/D
AT_Status	0	N/D
AbsError	0.0	N/D
▶ transferGain	<input checked="" type="checkbox"/>	N/D
+ My_GainTransfer_1

15 Osservare se il controllore è aggiornato con il parametro del guadagno ottenuto dopo l'auto tuning.

Monitoraggio della variabile

Variabili globali utente - Micro850 Variabili locali - N/D Variabili di

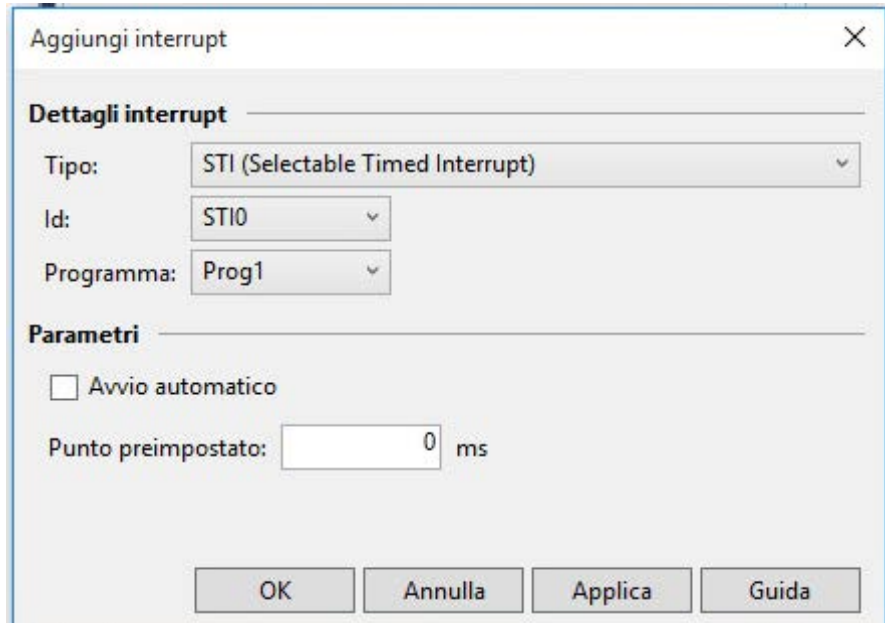
Nome	Valore logico	Valore fisico
- My_Gains
My_Gains.DirectActing	<input type="checkbox"/>	N/D
My_Gains.ProportionalGain	34.4978	N/D
My_Gains.TimeIntegral	27.6775	N/D
My_Gains.TimeDerivative	9.795	N/D
My_Gains.DerivativeGain	0.1	N/D
ManipulatedValue	0.0	N/D
▶ - OutGains
OutGains.DirectActing	<input type="checkbox"/>	N/D
OutGains.ProportionalGain	34.4978	N/D
OutGains.TimeIntegral	27.6775	N/D
OutGains.TimeDerivative	9.795	N/D
OutGains.DerivativeGain	0.0	N/D

Vedere anche

[Utilizzo dell'auto tuning con il blocco funzione IPIDController](#) a pagina 568

Utilizzo di un STI (Selectable Timed Interrupt) con auto tuning

Sebbene un'istruzione PID funzioni anche se non è controllata da STI (Selectable Timed Interrupt), l'utilizzo di un STI aumenta la percentuale di successo dell'auto tuning perché quest'ultimo funziona in base a un ciclo fisso.

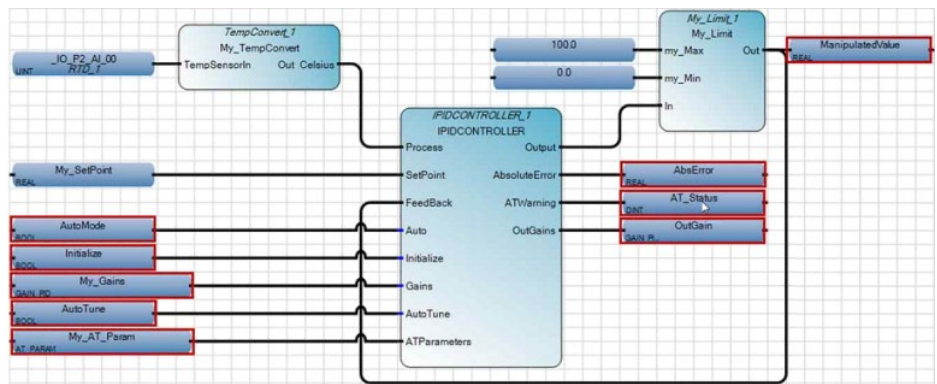


Vedere anche

[Utilizzo dell'auto tuning con il blocco funzione IPIDController](#) a pagina 568

Esempio: IPIDController con auto tuning

Nel programma dell'esempio seguente sono illustrate le variabili impiegate per configurare i parametri di auto tuning.



Parametri di auto tuning

Nella tabella seguente sono descritte le variabili utilizzate con ciascun parametro dell'esempio per configurare l'auto tuning.

Parametri di ingresso		
Variabile	Parametri	Descrizione
AutoMode	Auto	La modalità operativa del controllore PID: TRUE: il controllore funziona in modalità normale. FALSE: il termine derivativo è ignorato, forzando il tracciamento del feedback nei limiti del controllore a livello di uscita e consentendo al controllore di tornare alla modalità automatica senza il bumping dell'uscita.
Initialize	Initialize	Inizializza la sequenza AutoTune. In caso di modifica del valore da TRUE a FALSE o da FALSE a TRUE, il controllore elimina qualsiasi guadagno proporzionale durante il ciclo.
My_Gains	Gains	Stabilisce i guadagni PID per IPIDController.
My_Gains.DirectActing	DirectActing	Definisce il tipo di azionamento dell'uscita. TRUE - Azionamento diretto in cui l'uscita si sposta nella stessa direzione dell'errore. Ovvero, il valore attuale del processo è maggiore del SetPoint e l'azione appropriata del controllore è di incrementare l'uscita. Ad esempio, raffreddamento. FALSE - Azionamento inverso in cui l'uscita si sposta nella direzione opposta all'errore. Ovvero, il valore attuale del processo è maggiore del SetPoint e l'azione appropriata del controllore è di diminuire l'uscita. Ad esempio, riscaldamento.
My_Gains.ProportionalGain	ProportionalGain	Guadagno proporzionale per PID ($\geq 0,0001$).
My_Gains.TimeIntegral	TimeIntegral	Valore integrale di tempo per PID ($\geq 0,0001$). La tendenza all'oscillazione aumenta con la diminuzione dell'integrale del tempo.
My_Gains.TimeDerivative	TimeDerivative	Valore derivativo di tempo per PID ($> 0,0$). Lo smorzamento aumenta all'aumentare della derivata del tempo, ma si riduce se il valore della derivata del tempo è eccessivo.
My_Gains.DerivativeGain	Guadagno derivativo per PID ($> 0,0$).	
AutoTune	Se impostato su TRUE e Auto e Initialize sono su FALSE, la sequenza AutoTune è stata avviata.	
ATParameters		
Load	<ul style="list-style-type: none"> • Valore iniziale dell'uscita durante l'auto tuning. • Permette al valore di processo di stabilizzarsi sul valore di carico. 	
Deviation	<ul style="list-style-type: none"> • La deviazione standard per una serie di valori di processo stabilizzati. Ad esempio, se il valore di processo è stabilizzato tra 31,4 e 32,0, il valore di deviazione sarà $(32,0-31,4)/2 = 0,3$. • Alcuni valori di processo, tipo quello della temperatura, richiedono molto tempo per stabilizzarsi. 	
Passo	<ul style="list-style-type: none"> • Il processo di auto tuning deriva i parametri Gain in base a come il valore di processo cambia in seguito alla modifica del valore del passo. 	
ATDynaSet	<ul style="list-style-type: none"> • Tempo allocato per il completamento dell'auto tuning. Deve essere maggiore rispetto al tempo richiesto per il processo di auto tuning. • Un valore comune per molti sistemi è 600 secondi, ma per alcuni può essere necessario più tempo. 	
ATReset	<ul style="list-style-type: none"> • Se TRUE, l'uscita verrà ripristinata a "0" dopo il completamento dell'auto tuning. • Se FALSE, l'uscita resterà sul valore di carico una volta completato l'auto tuning. 	
Parametri di uscita		
Parametri	Descrizione	
AbsoluteError	Errore assoluto (Process - SetPoint) dal controllore.	
ATWarning	Avviso per la sequenza Auto Tune. I possibili valori sono: <ul style="list-style-type: none"> • 0: nessuna sintonizzazione automatica eseguita. • 1 - In modalità auto tuning. • 2 - Auto tuning eseguito. • -1 - ERROR 1, ingresso impostato automaticamente su TRUE, nessun auto tuning possibile. • -2 - ERROR 2, errore auto tuning, ATDynaSet scaduto 	
OutGains	Guadagni calcolati dopo le sequenze AutoTune.	

Vedere anche

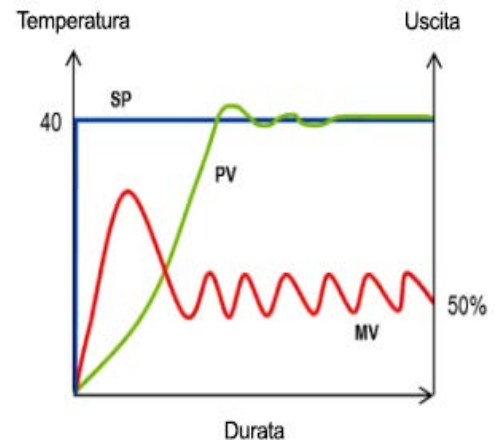
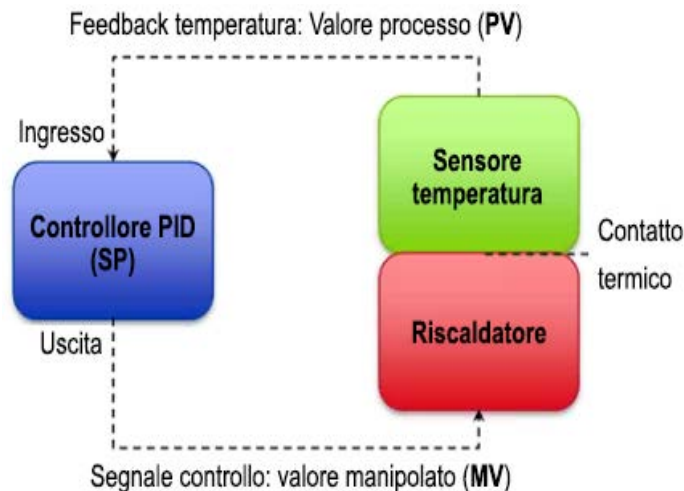
[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

Esempio: come creare un ciclo di feedback per il valore manipolato

L'aggiunta di un ciclo di feedback per il valore manipolato previene la sovraelongazione eccessiva fornendo un valore minimo e massimo per MV.

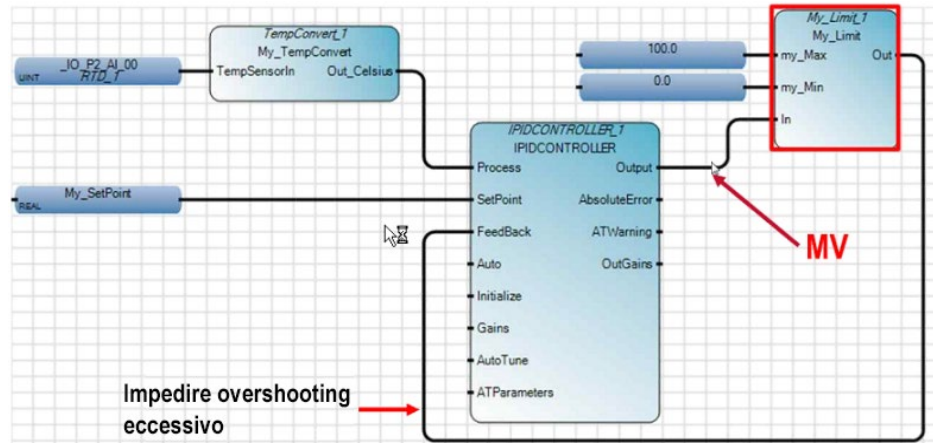
Esempio ciclo di feedback sulla temperatura

All'inizio del processo di controllo della temperatura la differenza tra il valore di processo (PV) e il valore setpoint (SP) è ampia, come mostrato nel seguente grafico. In questo esempio di ciclo di feedback sulla temperatura, PV inizia con 0 gradi Celsius e si sposta verso il valore SP di 40 gradi Celsius. Tenere anche presente che la fluttuazione tra il valore alto e basso manipolato (MV) scende e si stabilizza nel tempo. Il comportamento di MV dipende dai valori usati in ciascuno dei parametri P, I e D.



IPIDController con ciclo di feedback

Il seguente diagramma a blocchi funzionali include un ciclo di feedback per il valore manipolato che previene la sovralongazione eccessiva fornendo un valore minimo e massimo per MV.



Vedere anche

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

Esempio: come aggiungere un UDFB a un programma PID

Per i programmi PID, utilizzare i blocchi funzione definiti dall'utente (UDFB) esternamente al programma principale per eseguire funzioni specializzate, come conversione di unità o trasferimento di valori. Di seguito vengono forniti degli esempi di UDFB:

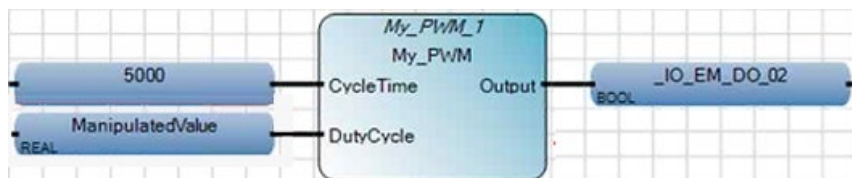
Trasferimento del valore di guadagno con auto tuning

Questo UDFB trasferisce il valore del guadagno AutoTune a My_GainTransfer per l'utilizzo da parte del controllore.



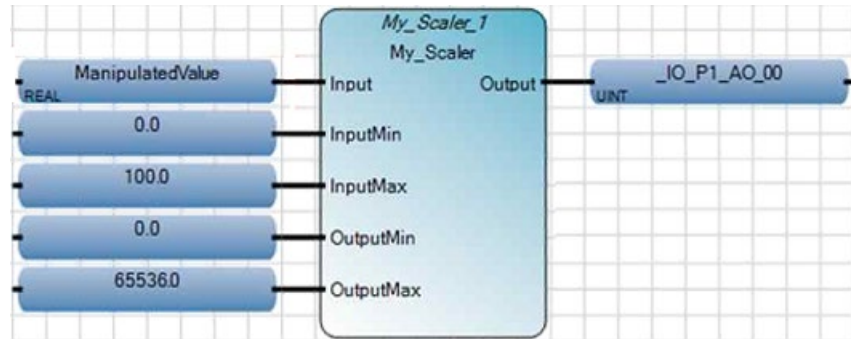
Conversione di un valore manipolato in un'uscita digitale

Questo UDFB converte un valore manipolato (MV) in un'uscita digitale (DO) in modo che sia possibile usarlo per controllare un ingresso digitale n(DI).



Conversione di un valore manipolato in un'uscita analogica

Questo UDFB converte un valore manipolato (MV) in un'uscita analogica (AO) in modo che sia possibile usarlo per controllare un ingresso analogico (AI).



Vedere anche

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

Esempio: come creare un programma IPIDController per controllare la temperatura

Il programma di controllo della temperatura mantiene la temperatura entro la zona di controllo.

Setpoint, processo e valori manipolati

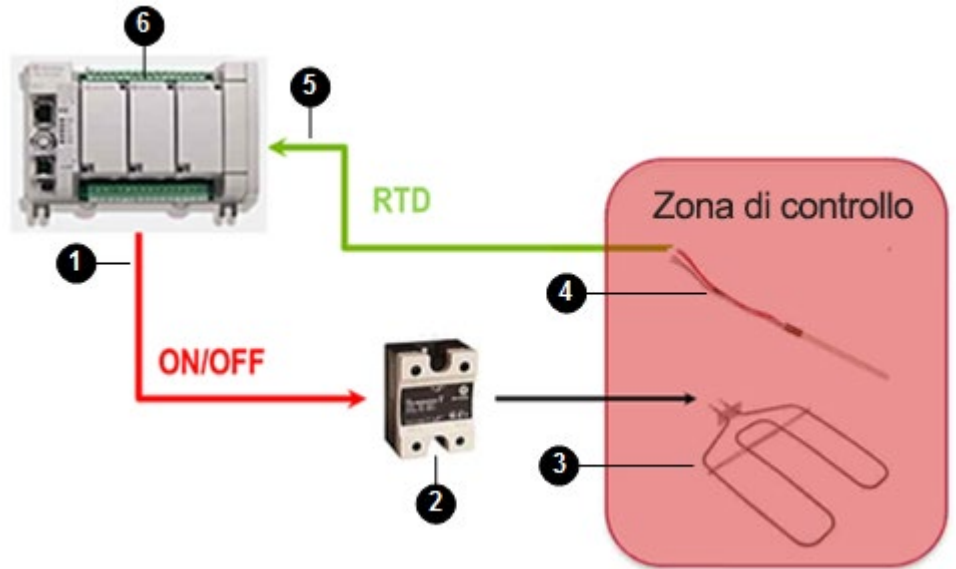
Nella tabella seguente viene definita la modalità di utilizzo dei valori SP, PV e MV nel programma di controllo della temperatura.

Elemento	Descrizione
Setpoint (SP)	Misurazione della temperatura in gradi Celsius; definisce la temperatura per la zona di controllo.
Valore di processo (PV)	È necessario convertirlo nella stessa unità di SP; la misura è in gradi Celsius.
Valore manipolato (MV)	È necessario convertirlo in un valore analogico, in modo da essere inviato alla PWM per il controllo dell'elemento riscaldatore.

Sistema di controllo della temperatura

Il diagramma e la tabella seguenti definiscono i componenti del sistema di controllo della temperatura che sono gestiti dal programma di

controllo della temperatura e descrivono gli eventi che si verificano durante l'esecuzione del programma di controllo.



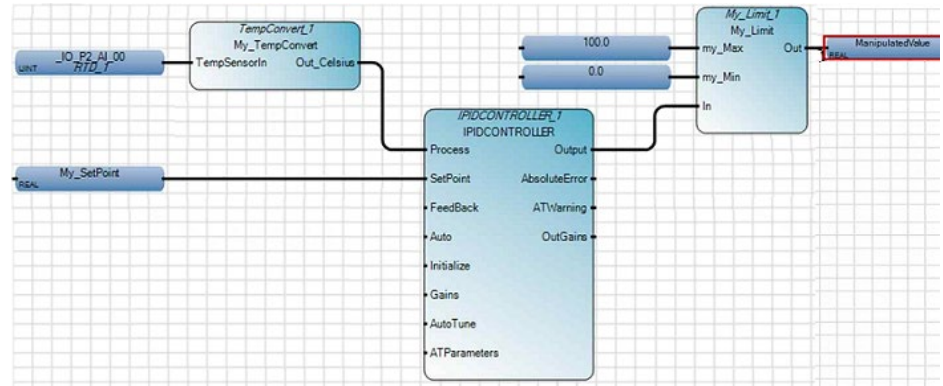
Sequenza di eventi per il programma di controllo della temperatura

Nella seguente tabella sono identificati i componenti del sistema di controllo della temperatura e sono descritti, in sequenza, gli eventi che si verificano durante l'esecuzione del programma di tale sistema.

No	Elemento	Descrizione
	Uscita controllore	Invia il MV alla PWM (On/Off).
	Modulazione di larghezza degli impulsi (controllore temperatura PWM)	Relè a stato solido che controlla l'elemento riscaldante.
	Elemento riscaldante	Aumenta la temperatura nella zona di controllo.
	Termometro a resistenza (RTD)	Misura la temperatura nella zona di controllo e invia il PV (segnale RTD) all'ingresso del controllore.
	Ingresso controllore	Riceve il PV (segnale RTD).
	Programma PLC	Converte il PV (segnale RTD) alla stessa unità del SP (gradi Celsius), determina la differenza tra il PV e il SP e regola il MV in base ai valori definiti nei parametri P, I e D.

Esempio: diagramma a blocchi funzionali per il controllo della temperatura

Questo diagramma a blocchi funzionali illustra i blocchi funzione predefiniti e definiti dall'utente e utilizzati nell'applicazione per il controllo della temperatura in una zona di controllo.



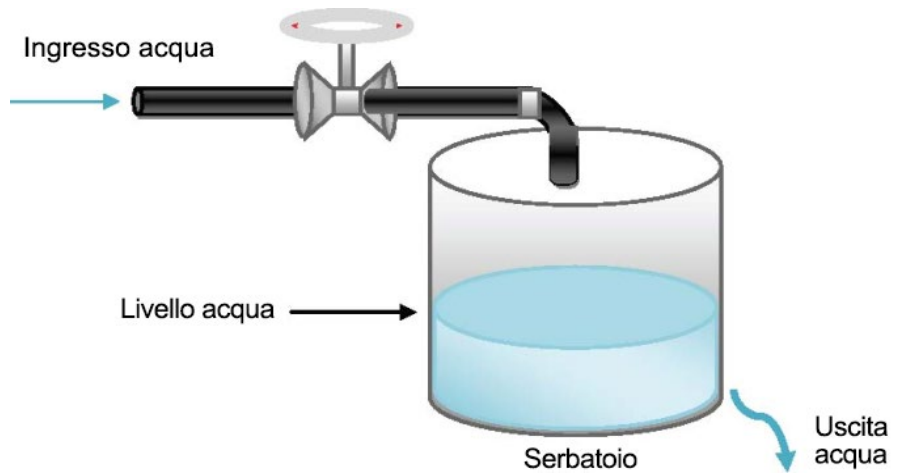
Vedere anche

[Esempio: come creare un programma IPIDController per controllare il livello di erogazione dell'acqua](#) a pagina 583

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

Esempio: come creare un programma IPIDController per controllare il livello di erogazione dell'acqua

Nell'esempio di programma di controllo per il livello di erogazione dell'acqua, viene mantenuta una quantità d'acqua sufficiente in un serbatoio di erogazione dell'acqua dotato di deflusso. Una valvola solenoide controlla l'acqua in ingresso riempiendo il serbatoio con una portata preimpostata; anche l'acqua che defluisce è controllata in base a una portata preimpostata.



Informazioni dell'esempio di programma

Il programma per il livello di erogazione dell'acqua include le seguenti informazioni.

- La sequenza degli eventi che si verificano nel processo di controllo
- Modalità di utilizzo nel programma di controllo del setpoint e dei valori manipolati e di processo
- Un diagramma a blocchi funzionali di esempio che illustra l'IPIDController e altri blocchi di istruzioni

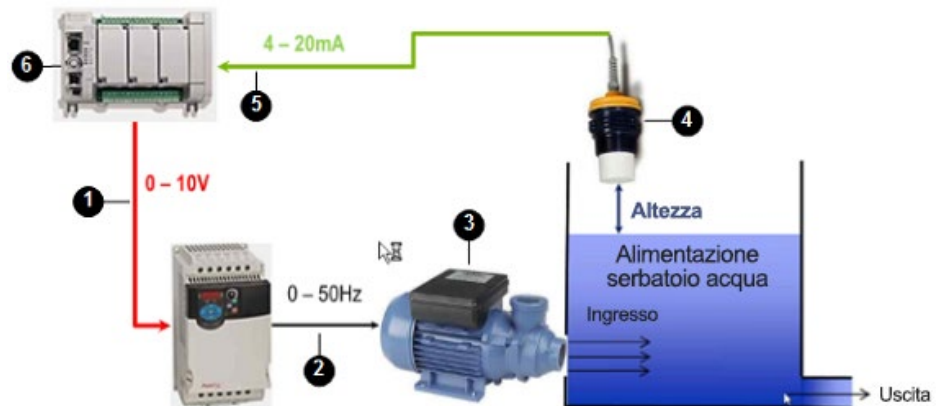
Setpoint, processo e valori manipolati

Nella tabella seguente è definita la modalità di utilizzo dei valori SP, PV e MV nel programma di controllo del livello di erogazione dell'acqua.

Elemento	Descrizione
Setpoint (SP)	Misurazione dell'altezza che definisce il livello di erogazione dell'acqua.
Valore di processo (PV)	Il valore 4 - 20 mA deve essere convertito nella stessa unità di SP, una misura di altezza.
Valore manipolato (MV)	È necessario convertirlo in un valore analogico, in modo da poter essere inviato all'unità per il controllo della pompa.

Sistema per il livello di erogazione dell'acqua

Nel seguente diagramma sono illustrati i componenti del sistema di controllo del livello di erogazione dell'acqua che sono comandati dal programma di tale sistema. Nella tabella riportata di seguito al diagramma sono descritti gli eventi che si verificano durante l'esecuzione del programma di controllo.



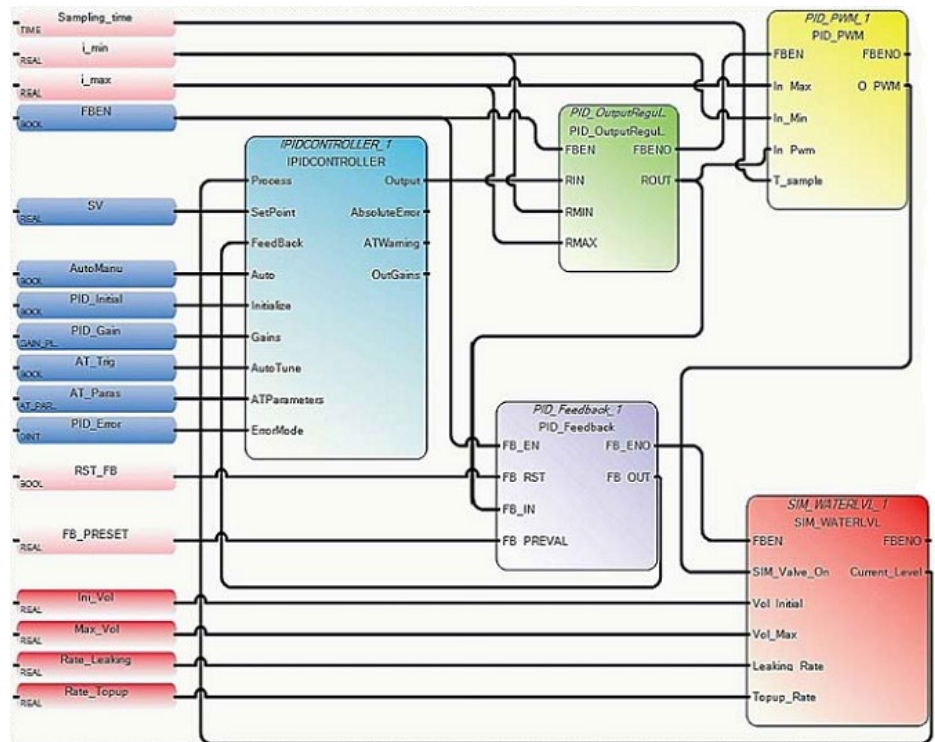
Sequenza degli eventi nel sistema per il livello di erogazione dell'acqua

Nella seguente tabella sono identificati i componenti del sistema di controllo del livello di erogazione dell'acqua e sono descritti, in sequenza, gli eventi che si verificano durante l'esecuzione del programma di tale sistema.

No	Elemento	Descrizione
1	Uscita controllore	Invia il MV all'unità PowerFlex (0 - 10 V).
2	Unità PowerFlex	Controlla la pompa dell'acqua (0 - 50 Hz).
3	Pompa dell'acqua	Controlla il livello dell'acqua nel serbatoio di erogazione.
4	Dispositivo di trasferimento delle uscite	Misura l'altezza del livello dell'erogazione dell'acqua (4 - 20 mA) e invia il PV al controllore.
5	Ingresso controllore	Riceve il PV (livello di erogazione dell'acqua 4 - 20 mA).
6	Programma PLC	Converte il PV alla stessa unità del SP (misura dell'altezza), determina la differenza tra il PV e il SP e regola il MV in base ai valori definiti nei parametri P, I e D.

Esempio: diagramma a blocchi funzionali per il controllo del livello di erogazione dell'acqua

Nel seguente diagramma a blocchi funzionali sono illustrati i blocchi funzione del programma predefiniti e definiti dall'utente per il controllo del livello di erogazione dell'acqua.



Blocchi funzione e UDFB usati nell'FBD del livello dell'acqua

Questa applicazione, sviluppata in linguaggio Diagramma a blocchi funzionali, utilizza i blocchi istruzione descritti nella seguente tabella.

Function block	Descrizione
Blocco funzione IPIDController	Fornisce il controllo del processo PID.
UDFB PID_OutputRegulator	Regola l'uscita di IPIDCONTROLLER in un intervallo sicuro per garantire che l'hardware impiegato nel processo non sia danneggiato. Codice campione: IF $RMIN \leq RIN \leq RMAX$, then $ROUT = RIN$, IF $RIN < RMIN$, then $ROUT = RMIN$, IF $RIN > RMAX$, then $ROUT = RMAX$
UDFB PID_Feedback	Agisce come un multiplexer. Codice campione: IF "FB_RST" is false, $FB_OUT=FB_IN$; If "FB_RST" is true, then $FB_OUT=FB_PREVAL$.
UDFB PID_PWM	Fornisce una funzione PWM convertendo un valore reale in un'uscita ON/OFF dipendente dal tempo.
UDFB SIM_WATERLVL	Simula il processo nell'esempio dell'applicazione.

Vedere anche

[Esempio: come creare un programma IPIDController per controllare la temperatura](#) a pagina 581

[Utilizzo dell'istruzione proporzionale integrale derivativa](#) a pagina 565

PID (proporzionale integrale derivativa)

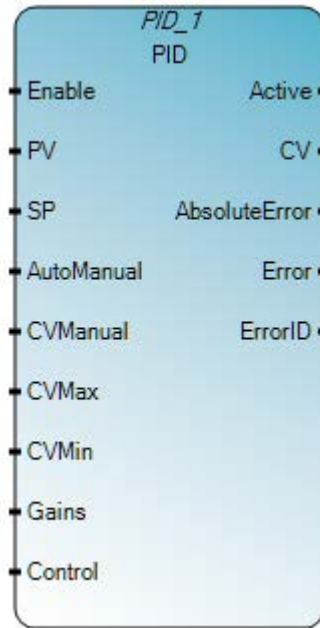
Un'istruzione di uscita che controlla le proprietà fisiche come la temperatura, la pressione, il livello dei liquidi o la portata tramite cicli di processo.

Dettagli operazione:

- Quando abilitato, PID controlla il processo utilizzando i parametri di ingresso inclusi SP e guadagni del controllore PID.
- Esegue la transizione dalla modalità Run alla modalità Programmazione, l'istruzione PID è disabilitato, i valori dei parametri vengono memorizzati.
- Esegue la transizione dalla modalità Programmazione alla modalità Run, istruzione PID resta disabilitata fino a quando un utente ripristina da Abilita a True.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Istruzione Abilita. TRUE: avvia l'esecuzione con i parametri di ingresso correnti. FALSE: PID non viene eseguito. Impostare CV su 0 e calcolare AbsoluteError.
PV	Ingresso	REAL	Valore processo. Questo valore è in genere letto da un modulo di ingresso analogico. L'unità SI deve essere la stessa di Setpoint.
SP	Ingresso	REAL	Il valore del punto preimpostato per il processo.
AutoManual	Ingresso	BOOL	Selezione della modalità automatica o manuale. TRUE: CV è controllato da PID. FALSE: esegue PID e CV è controllato dall'ingresso CVManual.
CVManual	Ingresso	REAL	Ingresso del valore di controllo definito per l'operazione modalità manuale. L'intervallo valido per CVManual è: CVMIn < CVManual < CVMMax
CVMIn	Ingresso	REAL	Limite minimo del valore di controllo. Se CV < CVMIn, quindi CV = CVMIn. Se CVMIn > CVMMax, si verifica un errore.
CVMMax	Ingresso	REAL	Limite massimo del valore di controllo. Se CV > CVMMax, quindi CV = CVMMax. Se CVMMax < CVMIn, si verifica un errore.
Gains	Ingresso	PID_GAINS	Guadagni PID per il controllore. Per configurare il parametro Guadagni, utilizzare il tipo di dati PID_GAINS.
Controllo	Ingresso	BOOL	Direzione di controllo del processo: TRUE: azionamento diretto, ad esempio raffreddamento. FALSE: azionamento inverso, ad esempio riscaldamento.
Attivo	Uscita	BOOL	Stato del controllore PID. TRUE: PID è attivo. FALSE: PID viene interrotto.
CV	Uscita	REAL	Uscita del valore di controllo. Se si è verificato un errore, CV è 0.

AbsoluteError	Uscita	REAL	L'errore assoluto è la differenza tra il valore del processo (PV) e il valore del punto preimpostato (SP).
Errore	Uscita	BOOL	Indica la presenza di una condizione di errore. TRUE: l'operazione ha riscontrato un errore. FALSE: l'operazione è stata completata o l'istruzione FFU non è in esecuzione.
ErrorID	Uscita	USINT	Numero univoco che identifica l'errore. Gli errori sono definiti nei codici di errore PID.

Tipo di dati PID_GAINS

Nella tabella seguente è descritto il tipo di dati PID_GAINS per l'istruzione PID.

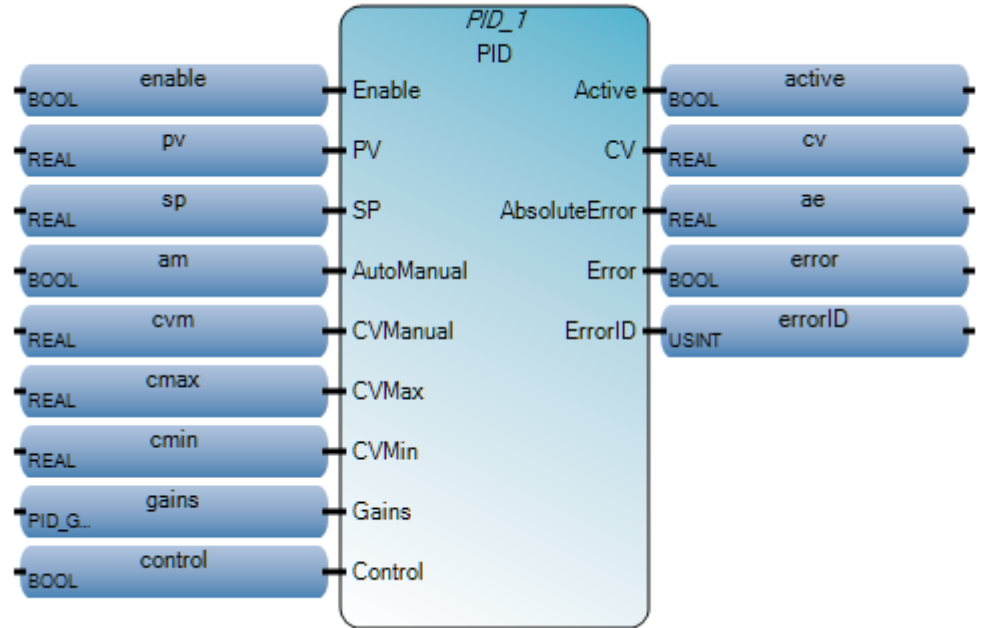
Parametri	Tipo di parametro	Tipo di dati	Descrizione
Kc	Ingresso	REAL	Guadagno controllore per PID. Proporzionale e integrale dipendono da questo guadagno. ($\geq 0,0001$). L'aumento di Kc migliora il tempo di risposta ma aumenta anche il superamento e oscillazione del PID. Se Kc non è valido, si verifica un errore.
Ti	Ingresso	REAL	Costante integrale di tempo in secondi ($\geq 0,0001$). L'aumento di Ti diminuisce il superamento e oscillazione del PID. Se Ti non è valido, si verifica un errore.
Td	Ingresso	REAL	Costante derivativa di tempo in secondi ($\geq 0,0$). Quando Td è uguale 0, non c'è nessuna azione derivata e PID diventa un controllore PI. L'aumento di Td riduce il superamento e rimuove l'oscillazione del controllore PID. Se Td non è valido, si verifica un errore.
FC	Ingresso	REAL	Costante del filtro ($\geq 0,0$). L'intervallo consigliato per FC è da 0 a 20. L'aumento di FC uniforma la risposta del controllore PID. Se FC non è valido, si verifica un errore.

Codici di errore PID

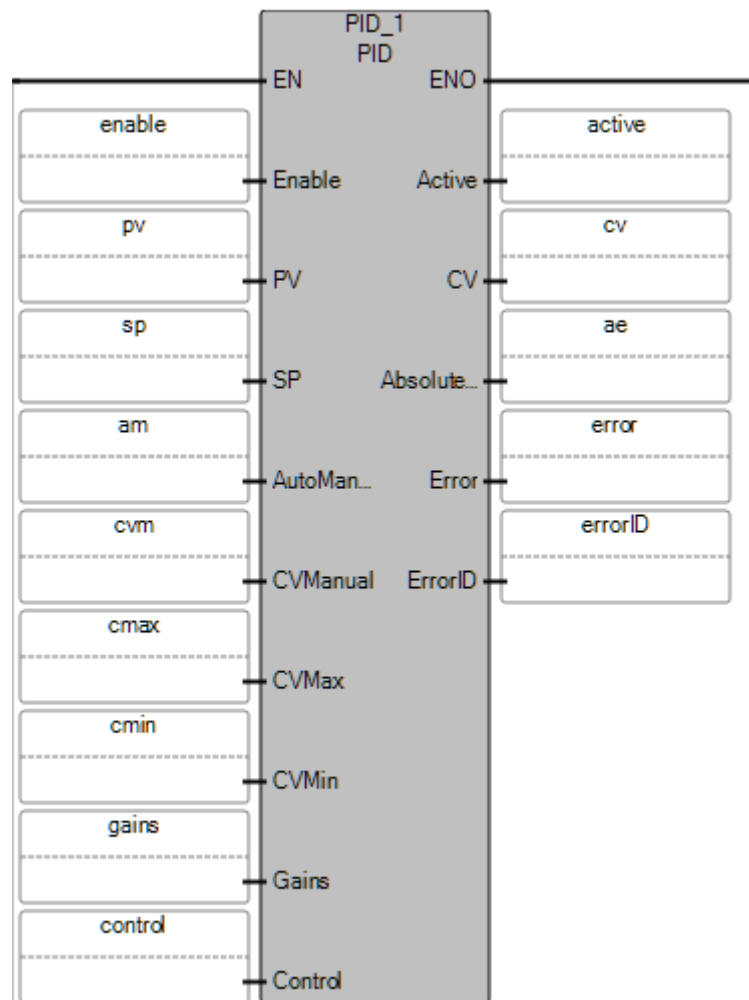
Utilizzare questa tabella per determinare i codici di errore PID e le descrizioni.

Codice errore	Descrizione errore
0	PID funziona normalmente.
1	Kc non è valido.
2	Ti non è valido.
3	Td non è valido.
4	FC non è valido.
5	$CVMin > CVMax$ o $CVMax < CVMin$
6	$CVManual < CVMin$ CVManual non è valido.
7	$CVManual > CVMax$ CVManual non è valido.

Esempio di Programmazione a blocchi funzionali PID



Esempio di Diagramma ladder PID



Esempio di Testo strutturato PID

```

1  PID_1(enable, pv, sp, am, cvm, cmax, cmin, gains, control);
2  active := PID_1.Active;
3  cv := PID_1.CV;
4  ae := PID_1.AbsoluteError;
5  error := PID_1.Error;
6  errorID := PID_1.ErrorID;

```

PID_1
void PID_1(BOOL Enable, REAL PV, REAL SP, BOOL AutoManual, REAL CVManual, REAL CVMax, REAL CVMin, PID_GAINS Gains, BOOL Control)
Tipo: PID, Derivata integrale proporzionale.

Risultati PID

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dime
PID_1	<input type="checkbox"/>	PID	
enable	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
pv	1.0	N/D	1.0	<input type="checkbox"/>	REAL	
sp	10.0	N/D	10.0	<input type="checkbox"/>	REAL	
am	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
ovm	20.0	N/D	20.0	<input type="checkbox"/>	REAL	
cmx	30.0	N/D	30.0	<input type="checkbox"/>	REAL	
cmin	1.0	N/D	1.0	<input type="checkbox"/>	REAL	
gains	<input type="checkbox"/>	PID_GAINS	
control	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
errorID	1	N/D		<input type="checkbox"/>	USINT	
error	<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
ae	9.0	N/D		<input type="checkbox"/>	REAL	
active	<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL	
ov	0.0	N/D		<input type="checkbox"/>	REAL	

Vedere anche

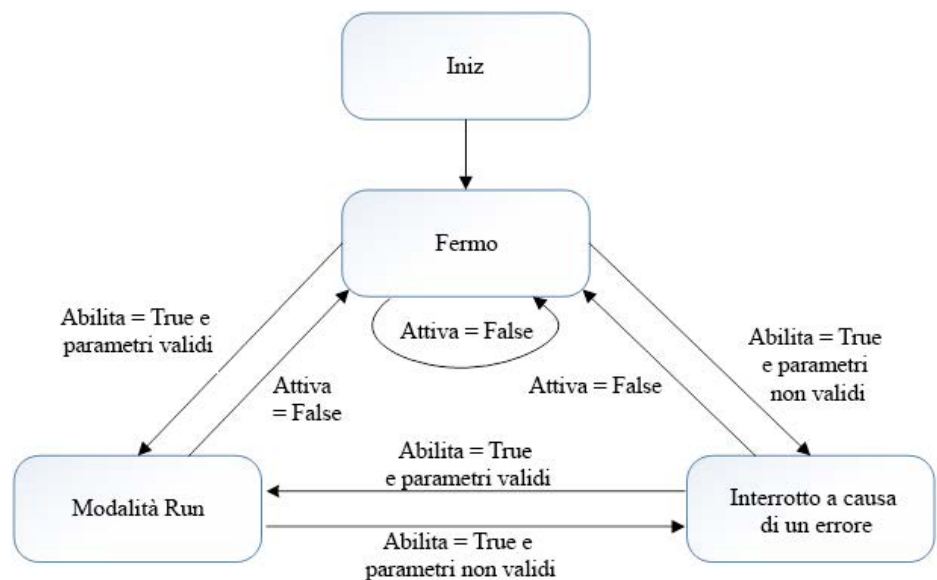
[Diagrammi di temporizzazione dell'istruzione PID](#) a pagina 592

[Stato della macchina di istruzione PID](#) a pagina 591

[Istruzioni proporzionali integrali derivative \(PID\)](#) a pagina 557

Stato della macchina di istruzione PID

Nel diagramma dello stato della macchina PID sono descritti gli stati di elaborazione per l'istruzione PID.



Vedere anche

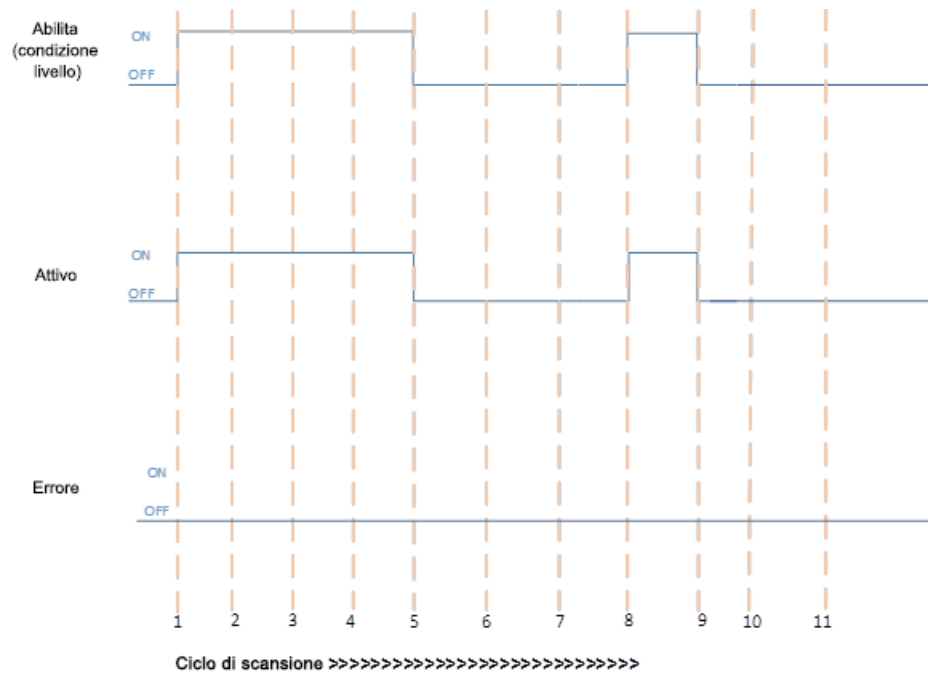
[Istruzione PID](#) a pagina 586

[Diagrammi di temporizzazione dell'istruzione PID](#) a pagina 592

Diagrammi di temporizzazione dell'istruzione PID

Gli esempi di diagramma di temporizzazione seguenti descrivono scenari di esecuzione per l'istruzione PID (proporzionale-integrale-derivativa).

Completamento dell'esecuzione PID.

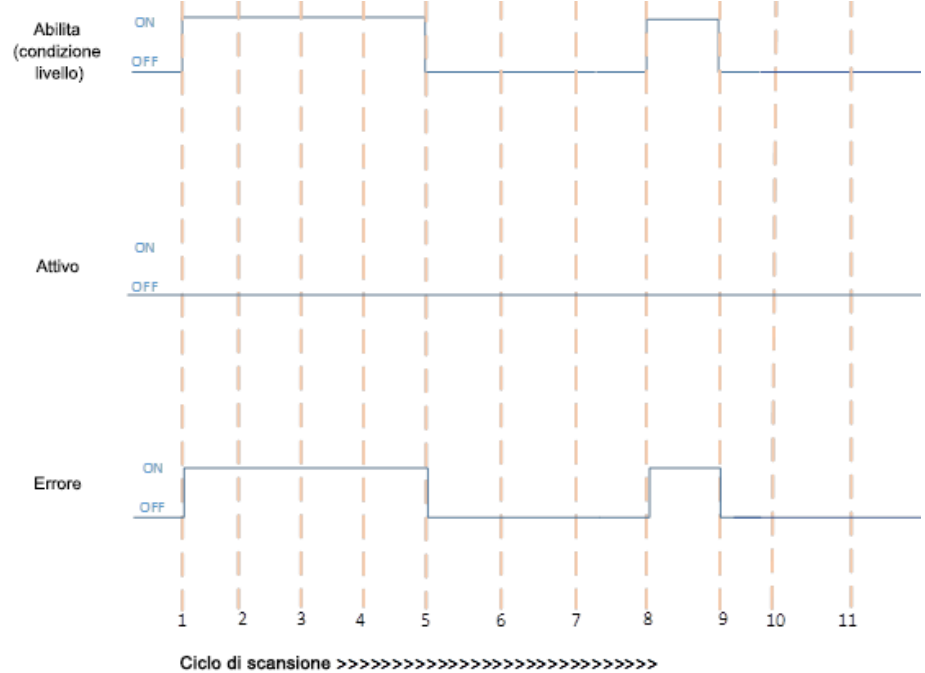


Utilizzare questa tabella per determinare i valori del parametro per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1, 8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso sono validi. • Il bit Attivo è TRUE. • Il bit Errore è FALSE.
2, 3, 4	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso sono validi. • Aggiorna i parametri di uscita del PID.
5, 9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Cancella i parametri di uscita del PID tranne AbsoluteError. • AbsoluteError calcola in base ai valori di ingresso PV e SP.

Ciclo di scansione	Descrizione
6, 7, 10, 11	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Cancella i parametri di uscita del PID tranne AbsoluteError. • AbsoluteError calcola in base ai valori di ingresso PV e SP.

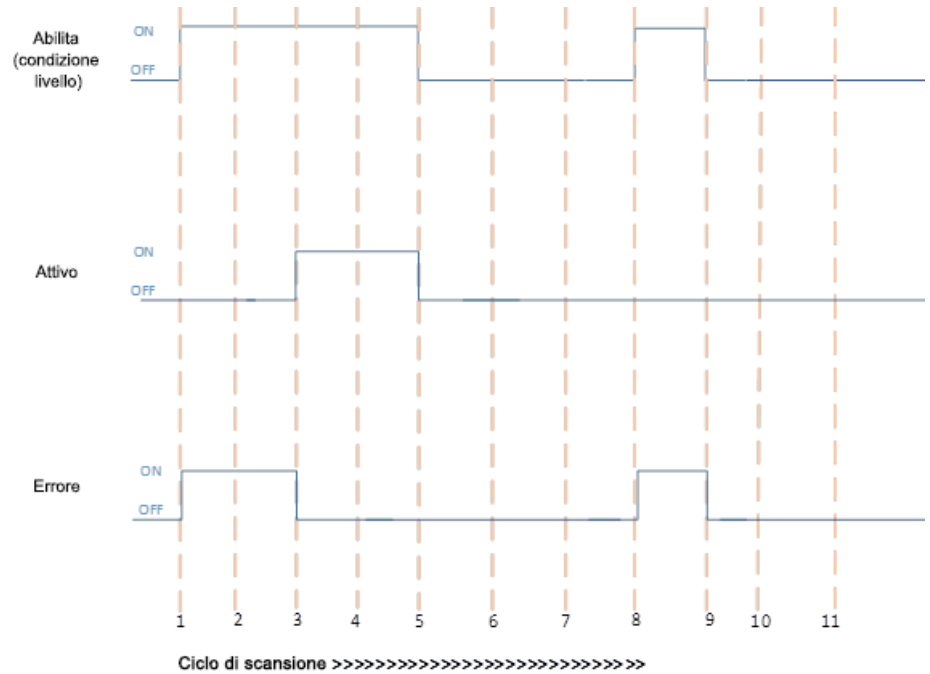
Esecuzione di PID con errore



Utilizzare questa tabella per determinare i valori del parametro per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1, 8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso non sono validi. • Il bit Attivo è FALSE. • Il bit Errore è TRUE. L'uscita ErrorID è impostata. • L'uscita CV uscita è impostata su 0. • AbsoluteError calcola in base ai valori di ingresso PV e SP.
2, 3, 4	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso non sono validi. • Aggiorna i parametri di uscita del PID.
5, 9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Cancella i parametri di uscita del PID tranne AbsoluteError. • AbsoluteError calcola in base ai valori di ingresso PV e SP.
6, 7, 10, 11	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Cancella i parametri di uscita del PID tranne AbsoluteError. • AbsoluteError calcola in base ai valori di ingresso PV e SP.

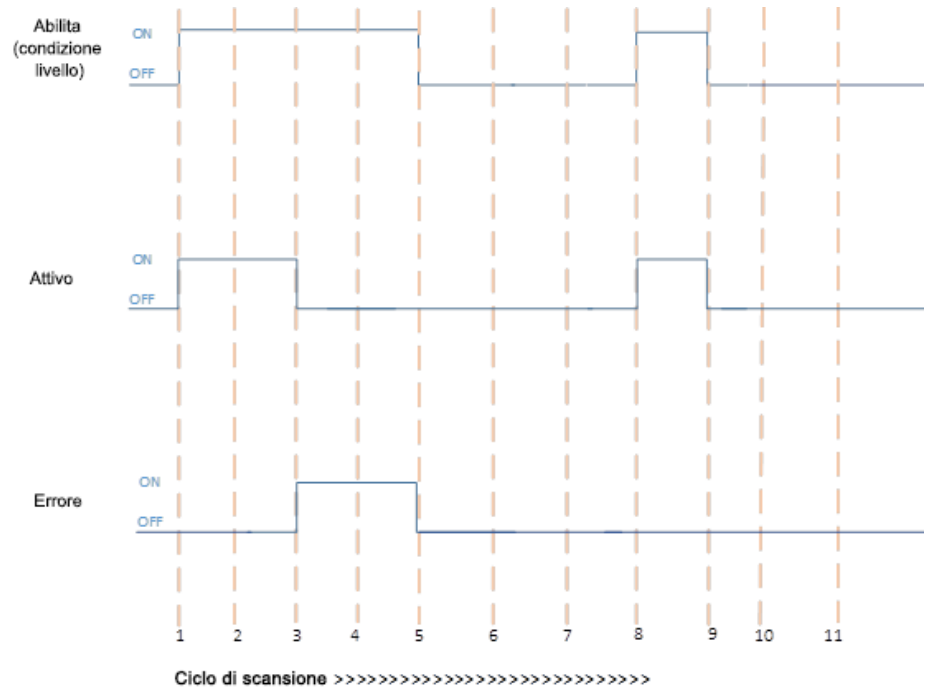
Esecuzione del PID con errore e completamento dell'esecuzione



Utilizzare questa tabella per determinare i valori del parametro per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1, 8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso non sono validi. • Il bit Attivo è FALSE. • Il bit Errore è TRUE. L'uscita ErrorID è impostata. • L'uscita CV uscita è impostata su 0. • AbsoluteError calcola in base ai valori di ingresso PV e SP.
2	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso non sono validi. • Aggiorna i parametri di uscita del PID.
3, 4	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso sono validi. • Il bit Attivo è TRUE. • Il bit Errore è FALSE. • Aggiorna i parametri di uscita del PID.
5, 9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Cancella i parametri di uscita del PID tranne AbsoluteError. • AbsoluteError calcola in base ai valori di ingresso PV e SP.
6, 7, 10, 11	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Cancella i parametri di uscita del PID tranne AbsoluteError. • AbsoluteError calcola in base ai valori di ingresso PV e SP.

Completamento dell'esecuzione PID ed Errore



Utilizzare questa tabella per determinare i valori del parametro per ogni ciclo di scansione.

Ciclo di scansione	Descrizione
1, 8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso sono validi. • Il bit Attivo è TRUE. • Il bit Errore è FALSE. • Aggiornare i parametri di uscita del PID.
2	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso sono validi. • Aggiornare i parametri di uscita del PID.
3, 4	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit di ingresso Abilita è TRUE. • I parametri di ingresso non sono validi. • Il bit Errore è TRUE. L'uscita ErrorID è impostata. • L'uscita CV uscita è impostata su 0. • AbsoluteError calcola in base ai valori di ingresso PV e SP.
5, 9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Cancella i parametri di uscita del PID tranne AbsoluteError. • AbsoluteError calcola in base ai valori di ingresso PV e SP.
6, 7, 10, 11	Nessuna modifica nella condizione ramo. <ul style="list-style-type: none"> • Il bit Abilita è FALSE. • Cancella i parametri di uscita del PID tranne AbsoluteError. • AbsoluteError calcola in base ai valori di ingresso PV e SP.

Vedere anche

[PID](#) a pagina 586

[Stato della macchina di istruzione PID](#) a pagina 591

Istruzioni Real Time Clock (RTC)

Utilizzare le istruzioni Real Time Clock per configurare il calendario e l'orologio.

Istruzione	Descrizione
RTC_READ a pagina 403	Legge le informazioni sul modulo real-time clock (RTC).
RTC_SET a pagina 405	Imposta i dati Real-Time Clock sulle informazioni del modulo RTC.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

RTC_READ (lettura Real-Time Clock)

Legge le informazioni sul modulo real-time clock (RTC).

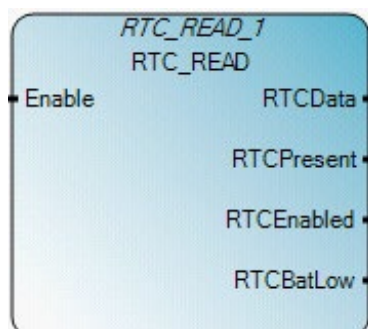
Dettagli operazione:

- Controllori Micro810 o Micro820 con RTC integrato:
 - RTCBatLow è sempre impostato su zero (0).
 - RTCEnabled è sempre impostato su (1).
- Quando l'RTC integrato perde la carica/memoria a causa di una interruzione dell'alimentazione:
 - RTCDData è impostato su 2000/1/1/0/0/0.
 - RTCEnabled è impostato su (1).

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

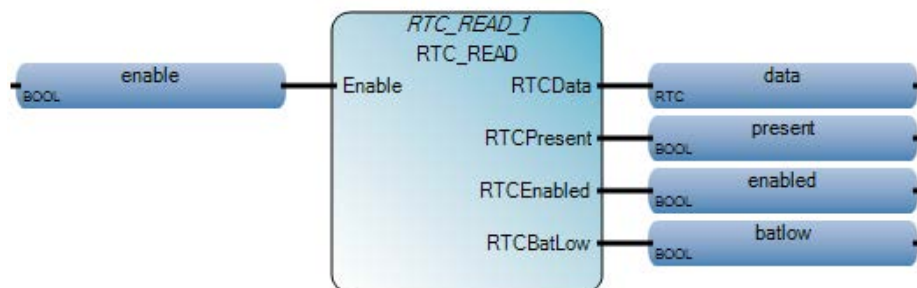
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue la lettura delle informazioni RTC. FALSE: non c'è nessuna operazione di lettura e i dati in uscita RTC non sono validi.
RTCData	Uscita	RTC	Informazioni sui dati RTC: aa/mm/gg, hh/mm/ss, settimana. L'uscita RTCData è definita con il tipo di dati RTC.
RTCPresent	Uscita	BOOL	TRUE: Free Running clock in uso oppure hardware RTC collegato. FALSE: Free Running clock non in uso oppure hardware RTC non collegato.
RTCEnabled	Uscita	BOOL	TRUE: Free Running clock in uso oppure hardware RTC abilitato (temporizzazione). FALSE: Free Running clock non in uso, hardware RTC disabilitato (nessuna temporizzazione).
RTCBatLow	Uscita	BOOL	TRUE - livello batteria RTC basso. FALSE - livello batteria RTC non è basso.
ENO	Uscita	BOOL	Abilita uscita. Applicabile solo alle programmazioni in Ladder Diagram.

Tipo di dati RTC

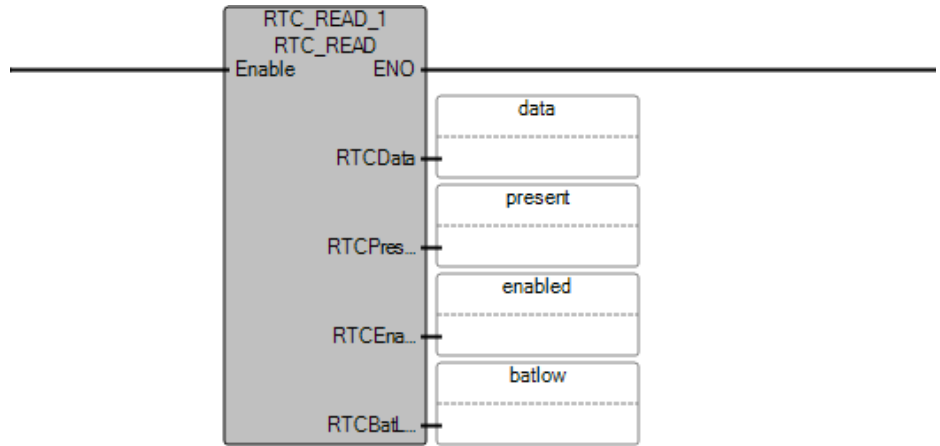
Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati RTC.

Parametri	Tipo di dati	Descrizione
Year	UINT	L'impostazione anno per RTC. Valore a 16 bit, intervallo valido da 2000 (01 gen, 00:00:00) a 2098 (31 dic, 23:59:59)
Month	UINT	L'impostazione mese per RTC.
Day	UINT	L'impostazione giorno per RTC.
Hour	UINT	L'impostazione ora per RTC.
Minute	UINT	L'impostazione minuto per RTC.
Second	UINT	L'impostazione secondi per RTC.
DayOfWeek	UINT	L'impostazione giorno della settimana per RTC. Questo parametro è ignorato per RTC.SET.

Esempio di Programmazione a blocchi funzionali RTC_READ



Esempio di Diagramma ladder RTC_READ



Esempio di Testo strutturato RTC_READ

```

RTC_READ_1(|
void RTC_READ_1(BOOL Enable)
Tipo: RTC_READ, Leggere le informazioni del modulo RTC.

1| RTC_READ_1(enable);
2| data := RTC_READ_1.RTCDData;
3| present := RTC_READ_1.RTCPresent;
4| enabled := RTC_READ_1.RTCEnabled;
5| batlow := RTC_READ_1.RTCBatLow;

```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

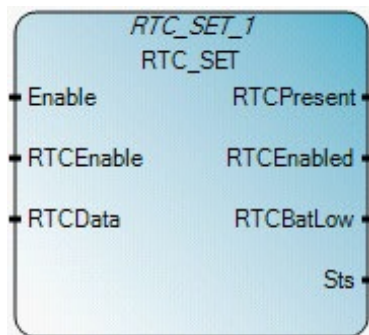
RTC_SET (impostare Real-Time Clock)

Impostare i dati RTC (Real-Time Clock) sulle informazioni del modulo RTC.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

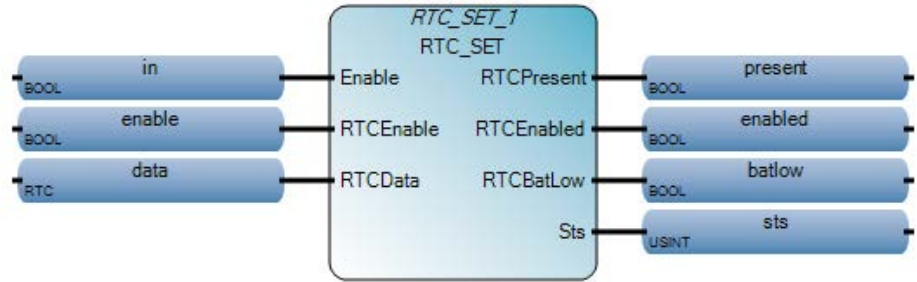
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Enable	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: esegue RTC_SET con le informazioni su RTC dall'ingresso. Eseguito in genere per 1 scansione del programma durante l'aggiornamento di RTC. FALSE - non esegue RTC_SET. Impostare su FALSE per utilizzare RTC normalmente.
RTCEnable	Ingresso	BOOL	TRUE - Per abilitare RTC i dati RTC specificati. FALSE - Per disabilitare RTC.
RTCData	Ingresso	RTC	Informazioni sui dati RTC: aa/mm/gg, hh/mm/ss, settimana definita nel tipo di dati RTC. RTCData è ignorato quando RTCEnable = 0.
RTCPresent	Uscita	BOOL	TRUE: Free Running clock in uso oppure hardware RTC collegato. FALSE: Free Running clock non in uso oppure hardware RTC non collegato.
RTCEnable	Uscita	BOOL	TRUE: Free Running clock in uso oppure hardware RTC abilitato (temporizzazione). FALSE: Free Running clock non in uso oppure hardware RTC disabilitato (nessuna temporizzazione).
RTCBatLow	Uscita	BOOL	TRUE - livello batteria RTC basso. FALSE - livello batteria RTC non è basso.
Sts	Uscita	USINT	Stato dell'operazione di lettura. Valori di stato (Sts) RTC_Set: <ul style="list-style-type: none"> • 0x00: blocco funzione non abilitato (nessuna operazione). • 0x01: operazione di impostazione RTC riuscita. • 0x02: operazione di impostazione RTC non riuscita.

Tipo di dati RTC

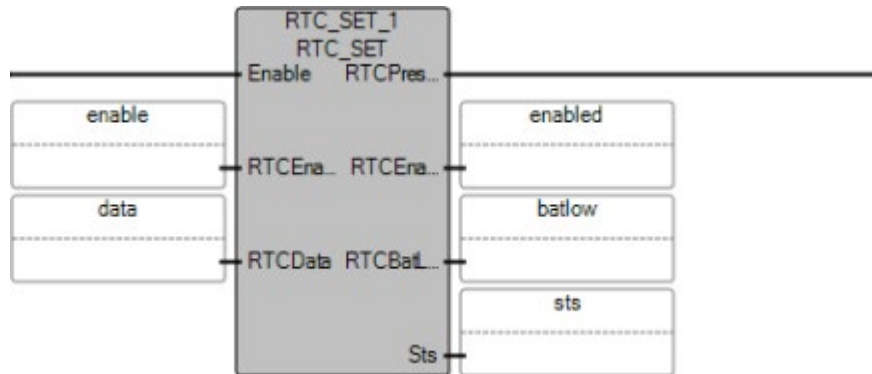
Utilizzare questa tabella per determinare i valori dei parametri del tipo di dati RTC.

Parametri	Tipo di dati	Descrizione
Year	UINT	L'impostazione anno per RTC. Valore a 16 bit, intervallo valido da 2000 (01 gen, 00:00:00) a 2098 (31 dic, 23:59:59)
Month	UINT	L'impostazione mese per RTC.
Day	UINT	L'impostazione giorno per RTC.
Hour	UINT	L'impostazione ora per RTC.
Minute	UINT	L'impostazione minuto per RTC.
Second	UINT	L'impostazione secondi per RTC.
DayOfWeek	UINT	L'impostazione giorno della settimana per RTC. Questo parametro è ignorato per RTC_SET.

Esempio di Programmazione a blocchi funzionali RTC_SET



Esempio di Diagramma ladder RTC_SET



Esempio di Testo strutturato RTC_SET

```

RTC_SET_1(
void RTC_SET_1(BOOL Enable, BOOL RTCEnable, RTC RTCData)
Tipo: RTC_SET, Impostare i dati RTC sul modulo RTC.

1 | RTC_SET_1(in, enable, data);
2 | present := RTC_SET_1.RTCPresent;
3 | enabled := RTC_SET_1.RTCEnable;
4 | batlow := RTC_SET_1.RTCBatLow;
5 | sts := RTC_SET_1.Sts;

```

Vedere anche

[Istruzioni ingresso/uscita](#) a pagina 355

Istruzioni socket

Utilizzare il protocollo del socket per le comunicazioni Ethernet verso i dispositivi che non supportano TCP Modbus ed Ethernet/IP. I socket supportano client, server, protocollo TCP (Transmission Control) e protocollo UDP (User Datagram). Le applicazioni tipiche comprendono le comunicazioni con stampanti, lettori di codici a barre e PC.

Comportamento delle istruzioni del socket in modalità **Modifica in modalità Run**:

- Se un'operazione di eliminazione viene eseguita su un'istruzione SOCKET_OPEN, SOCKET_ACCEPT, SOCKET_READ o SOCKET_WRITE durante la modalità **Modifica in modalità Run**, il Socket istanza viene eliminato.
- Nella modalità **Modifica in modalità Run**, qualsiasi modifica a un ingresso SOCKET_READ durante il funzionamento nello stato OCCUPATO genera un errore e il pacchetto ricevuto viene ignorato. I parametri di ingresso SOCKET_READ sono: Length, Offset, Data Array Size, Data Array Variable.
- SOCKET_READ è l'unica istruzione SOCKET che supporta operazioni di aggiunta o modifica durante la modalità **Modifica in modalità Run**.
- Se vengono modificate le impostazioni IP Ethernet utilizzando **Modifica in modalità Run**, tutte le istanze del socket create vengono eliminate, come per SOCKET_DELETEALL.

Elaborazione di istruzioni e aggiornamenti di uscite per le istruzioni del socket:

- **Asincrono**: Le istruzioni corrispondenti in cui tutte le uscite vengono aggiornate in modo asincrono con la scansione del programma utente, ad esempio una scansione ladder. L'uscita asincrona non può essere utilizzata per il rilevamento dell'attivazione sul fronte. I parametri dell'uscita asincrona non sono bloccati e possono essere aggiornati dopo il completamento delle rispettive istruzioni del socket.
- **Sincrono**: Le istruzioni corrispondenti in cui ogni uscita viene aggiornata in modo sincrono con le scansioni del programma utente. I parametri dell'uscita sincrona sono bloccati e non possono essere modificati dopo il completamento delle rispettive istruzioni del socket.
- **Ibrido**: Le istruzioni corrispondenti in cui alcune uscite vengono aggiornate in modo sincrono con la scansione del programma utente. Le uscite rimanenti vengono aggiornate in modo asincrono con la scansione del programma utente.
- **Esecuzione immediata dell'istruzione**: L'istruzione completa la funzione desiderata prima di passare all'istruzione successiva.
- **Esecuzione non immediata dell'istruzione**: L'istruzione richiede più scansioni del programma per completare la funzione

desiderata. Le istruzioni catturano un'istantanea dei parametri di ingresso quando viene rilevata una transizione **False > True**.

Utilizzare questa tabella per determinare l'utilizzo di istruzioni del socket.

Istruzione	Descrizione	Client TCP	Server TCP	UDP con apertura	UDP senza apertura	Elaborazione istruzioni	Aggiornamento uscita dell'istruzione
SOCKET_ACCEPT a pagina 604	Accetta una richiesta di connessione TCP da una destinazione remota e restituisce un'istanza di socket utilizzata per inviare e ricevere dati sulla connessione appena creata.	NO	SÌ	NO	NO	Non immediato	Ibrido
SOCKET_CREATE a pagina 607	Crea un'istanza del Socket e restituisce un numero di istanza che utilizza le operazioni socket successive.	SÌ	SÌ	SÌ	SÌ	Immediato	Sincrono
SOCKET_DELETE a pagina 612	Elimina un'istanza di socket creata. Le connessioni TCP vengono chiuse prima dell'eliminazione.	SÌ	SÌ	SÌ	SÌ	Non immediato	Sincrono
SOCKET_DELETEALL a pagina 615	Elimina tutte le istanze socket create.	SÌ	SÌ	SÌ	SÌ	Non immediato	Sincrono
SOCKET_INFO a pagina 618	Restituisce le informazioni per il socket come codici di errore e lo stato di esecuzione.	SÌ	SÌ	SÌ	SÌ	Immediato	Sincrono
SOCKET_OPEN a pagina 623	Le connessioni TCP vengono aperte con l'indirizzo di destinazione specificato. Le connessioni UDP associano un numero di porta e un indirizzo IP di destinazione al socket specificato.	SÌ	NO	SÌ	NO	Non immediato	Sincrono
SOCKET_READ a pagina 628	Legge i dati su un socket. Tenta di ricevere il numero di byte specificato e restituisce il numero dei byte ricevuti.	SÌ	SÌ	SÌ	SÌ	Non immediato	Ibrido
SOCKET_WRITE a pagina 632	Invia dati su un socket. Tenta di inviare il numero di byte richiesto e restituisce il numero di byte inviati.	SÌ	SÌ	SÌ	SÌ	Non immediato	Ibrido

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

[Diagrammi di temporizzazione dell'istruzione Socket](#) a pagina 641

[Diagrammi di transazione dell'istruzione Socket](#) a pagina 649

[Diagrammi di stato della macchina per TCP](#) a pagina 651

[Diagrammi dello stato della macchina per UDP](#) a pagina 654

SOCKET_ACCEPT

Solo per connessioni Transmission Control Protocol (TCP). Accetta una richiesta di connessione TCP da una destinazione remota e restituisce un'istanza di socket utilizzata per inviare e ricevere dati sulla connessione appena creata.

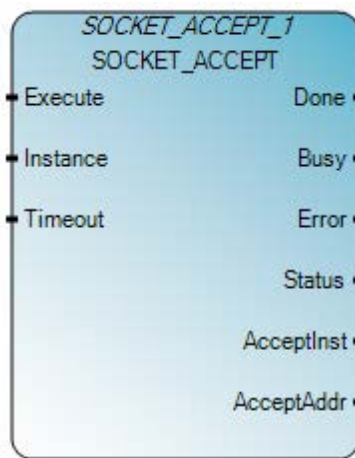
Dettagli operazione:

- Prima di eseguire Socket_Accept, eseguire SOCKET_CREATE e specificare il numero di porta locale per accettare la connessione.
- L'uscita viene aggiornata in modo sincrono dalla scansione del programma.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro820, Micro850 e Micro870. Per Micro800 Simulator, questa istruzione può essere aggiunta a un programma ma si tratta solo di un segnaposto per impedire che l'istruzione venga eliminata durante la modifica del controllore.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.

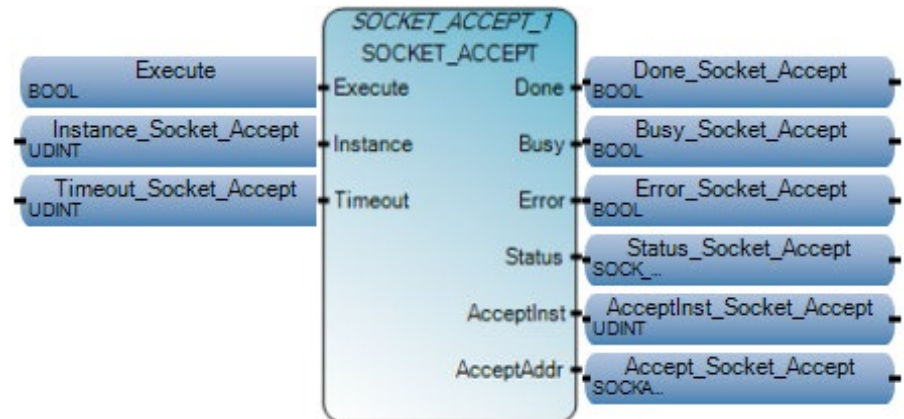


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

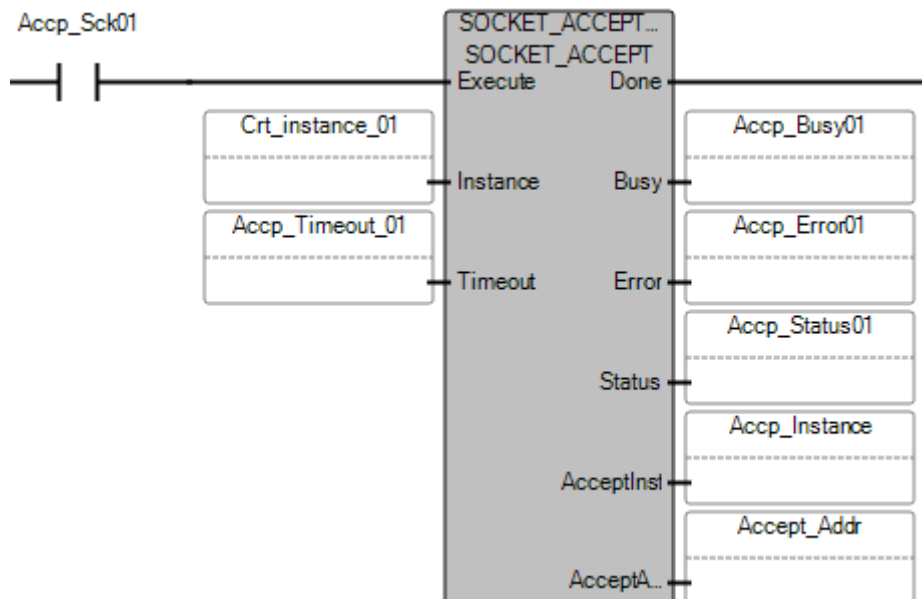
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: nessun fronte di salita rilevato, blocco istruzione non avviato.
Istanza	Ingresso	UDINT	Identifica l'istanza del socket. Copia il Socket Handler restituito da un'istruzione SOCKET_CREATE .
Timeout	Ingresso	UDINT	Timeout per le istanze del blocco istruzione SOCKET_ACCEPT . Il blocco funzione restituisce un errore se il valore di timeout è inferiore al valore minimo. Intervallo timeout: 1000-86400000 millisecondi Impostare il timeout su 0 per usare il valore predefinito 10.000 (10 secondi).
Done	Uscita	BOOL	Indica se l'operazione è stata completata. TRUE: l'operazione è stata completata correttamente. FALSE: l'operazione è in corso oppure è stato generato un errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Busy	Uscita	BOOL	TRUE: l'operazione non viene terminata. FALSE: l'operazione è completata. L'output viene aggiornato in modo sincrono alla scansione del programma.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore. L'output viene aggiornato in modo sincrono alla scansione del programma.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Stato	Uscita	SOCK_STATUS	Stato definito utilizzando il tipo di dati SOCK_STATUS che contiene informazioni su ErrorID, SubErrorID e StatusBits. L'output viene aggiornato in modo sincrono alla scansione del programma.
AcceptInst	Uscita	UDINT	Contiene l'istanza Accept per questa istanza Socket. Utilizzare il numero di istanza Accept univoco con i successivi SOCKET_READ e SOCKET_WRITE per la connessione. L'output viene aggiornato in modo sincrono alla scansione del programma.
AcceptAddr	Uscita	SOCKADDR_CFG	Una struttura dei dati che contiene l'indirizzo Accept per il socket. Per maggiori informazioni fare riferimento al Tipo di dati SOCKADDR_CFG. Per specificare l'indirizzo IP 192.168.2.100 e Porta = 12000: <ul style="list-style-type: none"> AcceptAddr.IPAddress[0]=192 AcceptAddr.IPAddress[1]=168 AcceptAddr.IPAddress[2]=2 AcceptAddr.IPAddress[3]=100 AcceptAddr.Port = 12000 L'output viene aggiornato in modo sincrono alla scansione del programma.

Esempio di Programmazione a blocchi funzionali **SOCKET_ACCEPT**



Esempio di Diagramma ladder **SOCKET_ACCEPT**



Esempio di Testo strutturato SOCKET_ACCEPT

```

SOCKET_ACCEPT_1 (
void SOCKET_ACCEPT_1(BOOL Execute, UDINT Instance, UDINT Timeout)
Tipo: SOCKET_ACCEPT, Accetta socket

SOCKET_ACCEPT_1(Execute_SOCKET_ACCEPT, Instance_SOCKET_ACCEPT, Timeout_SOCKET_ACCEPT);
Done_SOCKET_ACCEPT := SOCKET_ACCEPT. Done;
Busy_SOCKET_ACCEPT := SOCKET_ACCEPT. Busy;
Error_SOCKET_ACCEPT := SOCKET_ACCEPT. Error;
Status_SOCKET_ACCEPT := SOCKET_ACCEPT. Status;
AcceptInst_SOCKET_ACCEPT := SOCKET_ACCEPT. AcceptInst;
AcceptAddr_SOCKET_ACCEPT := SOCKET_ACCEPT. AcceptAddr;
    
```

Risultati

Server TCP

Nome	Alias	Valorelogico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
- Accep_Status01		<input type="checkbox"/>	SOCK_STATUS
Accep_Status01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Accep_Status01.SubErrorID		0	N/D		<input type="checkbox"/>	USINT
Accep_Status01.StatusBits		17	N/D		<input type="checkbox"/>	ULINT
Accep_Sec01		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Accep_Instance		1076385752	N/D		<input type="checkbox"/>	UDINT
Accep_Error01		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Accep_Busy01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
- Accep_Addr		<input type="checkbox"/>	SOCKADDR_CFG
Accep_Addr.Port		30002	N/D		<input type="checkbox"/>	USINT
Accep_Addr.IPAddress		<input type="checkbox"/>	IPADDR
Accep_Addr.IPAddress[0]		10	N/D		<input type="checkbox"/>	USINT
Accep_Addr.IPAddress[1]		224	N/D		<input type="checkbox"/>	USINT
Accep_Addr.IPAddress[2]		68	N/D		<input type="checkbox"/>	USINT
Accep_Addr.IPAddress[3]		221	N/D		<input type="checkbox"/>	USINT

Vedere anche

[Istruzioni socket](#) a pagina 603

[Tipo di dati SOCK_STATUS](#) a pagina 637

[Bit di stato per l'istruzione del socket](#) a pagina 641

[Codici di errore del socket](#) a pagina 637

[Tipo di dati SOCKADDR_CFG](#) a pagina 636

SOCKET_CREATE

Crea un'istanza del socket e restituisce un numero di istanza che viene usato come ingresso nelle operazioni socket successive.

Dettagli operazione:

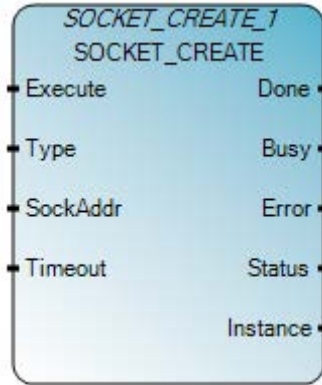
- Le istruzioni Socket supportano la comunicazione Full Duplex con i dispositivi remoti.
- Le connessioni User Datagram Protocol (UDP) supportano un massimo di otto pacchetti di datagrammi UDP in coda. La coda contiene i pacchetti più recenti.

- I controllori Micro820 e Micro850 versione 9 o successiva supportano fino a otto istanze Socket. Le istanze Socket supportano i socket UDP e TCP:
 - Usare tutte le otto istanze per la connessione del client Transmission Control Protocol (TCP).
 - Usare tutte le otto istanze per ascoltare le connessioni TCP in entrata, quindi accettare le otto connessioni da altri dispositivi.
 - Eseguire le operazioni client e server TCP.
 - Eseguire le operazioni TCP e UDP.
- Per accettare le connessioni TCP in entrata sulla stessa porta, creare una nuova istanza Socket.
- Connessione TCP persa:
 - Il programma applicativo dell'utente deve rilevare la perdita di connessioni TCP e gestire l'evento. A seconda dell'applicazione utente, prendere in considerazione l'opzione di un guasto al controllore.
 - Guasto al controllore.
 - Provare a ristabilire la connessione.
 - Per ristabilire le comunicazioni con un altro dispositivo:
 - Eliminare l'istanza Socket per la connessione persa.
 - Se la connessione è un client TCP, creare una nuova istanza Socket usando `SOCKET_CREATE` ed eseguire `SOCKET_OPEN` sul dispositivo di destinazione.
 - Se la connessione è un server TCP, creare una nuova istanza Socket usando `SOCKET_CREATE` ed eseguire `SOCKET_ACCEPT` per attendere un'altra connessione dal dispositivo remoto.
- Messaggi applicativi per le connessioni TCP:
 - Una connessione TCP è un flusso di byte tra due applicazioni. Il protocollo applicativo determina i formati del messaggio.
 - I messaggi possono avere dimensioni fisse o variabili.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro820, Micro850 e Micro870. Per Micro800 Simulator, questa istruzione può essere aggiunta a un programma ma si tratta solo di un segnaposto per impedire che l'istruzione venga eliminata durante la modifica del controllore.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.

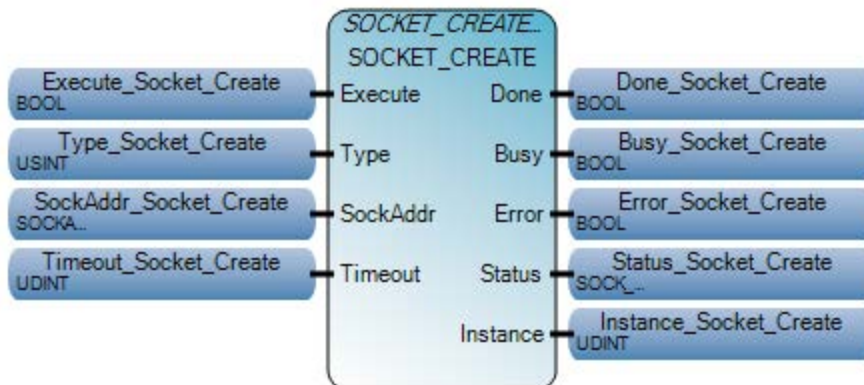


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

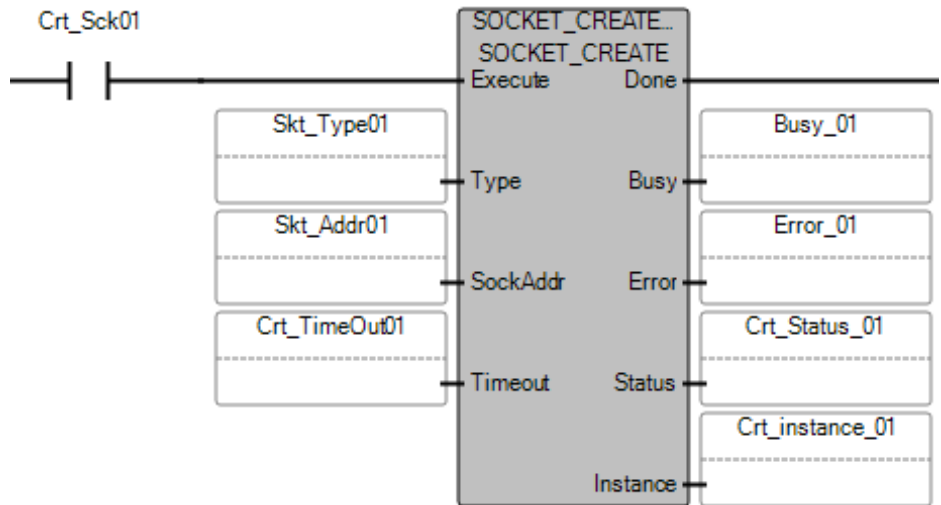
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: nessun fronte di salita rilevato.
Tipo	Ingresso	USINT	Specificare il tipo di Socket: <ul style="list-style-type: none"> • Transmission Control Protocol (TCP). • User Datagram Protocol (UDP).
SockAddr	Ingresso	SOCKADDR_CFG	Specificare la configurazione dell'indirizzo del socket. Il modulo EtherNet/IP per selezionare il numero di porta locale, impostare SockAddr su 0. Specificare un numero di porta locale in cui un'applicazione è in ascolto e riceve i dati, oppure: <ul style="list-style-type: none"> • Gli elementi array devono essere tutti zero • Per le operazioni client TCP, specificare 0 a meno che non si desideri utilizzare un numero di porta locale specifico. • Per la comunicazione del server TCP, specificare il numero di porta che dovrà accettare le richieste di connessione in entrata. • Per UDP, specificare un numero di porta locale che dovrà ricevere i datagrammi. Intervallo porte locali: da 1 a 65535. Se il numero di porta locale specificato è già utilizzato dal controllore Micro820 o Micro850, viene generato un errore. Il controllore usa i seguenti numeri di porta: Porte TCP: <ul style="list-style-type: none"> • EtherNet/IP: 44818 • ModbusTCP: 502 • Server DHCP: 67 • Client DHCP: 68 Porte UDP: <ul style="list-style-type: none"> • EtherNet/IP: 2222 • Server DHCP: 67 • Client DHCP: 68 Vedere Tipo di dati SOCKADDR_CFG .

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Timeout	Ingresso	UDINT	<p>Specificare il Timeout per inattività Socket.</p> <p>Se un'istanza Socket non riceve alcuna richiesta entro il timeout di inattività specificato, l'istanza Socket verrà eliminata.</p> <p>Se viene inviata una richiesta dopo l'eliminazione dell'istanza socket, viene restituito l'errore "istanza Socket non supportata".</p> <p>Il blocco istruzione restituisce un errore quando il valore di timeout è inferiore al valore minimo.</p> <p>Impostare il timeout in modo che sia maggiore dell'intervallo più lungo tra le operazioni del socket. Se il timeout di inattività è troppo breve, le istanze Socket potrebbero scadere.</p> <p>Intervallo timeout: 1000 - 86400000 millisecondi</p> <p>Impostare Timeout su 0 per usare il valore predefinito 300000 (5 minuti).</p>
Done	Uscita	BOOL	<p>Indica se l'operazione è stata completata.</p> <p>TRUE: l'operazione è stata completata correttamente.</p> <p>FALSE: l'operazione è in corso oppure è stato generato un errore.</p>
Busy	Uscita	BOOL	<p>TRUE: l'operazione non è completata.</p> <p>FALSE: l'operazione è completata.</p>
Errore	Uscita	BOOL	<p>Indica che si è verificato un errore.</p> <p>TRUE - È stato rilevato un errore.</p> <p>FALSE - Nessun errore.</p>
Stato	Uscita	SOCK_STATUS	<p>Stato definito utilizzando il tipo di dati SOCK_STATUS che contiene informazioni su ErrorID, SubErrorID e StatusBits.</p> <p>Verdere tipo di dati SOCK_STATUS, bit di stato istruzione Socket e codici di errore Socket.</p>
Istanza	Uscita	UDINT	<p>Contiene Socket Handler.</p> <p>Usare il parametro Instance per le istruzioni Socket successive.</p>

Esempio di Programmazione a blocchi funzionali SOCKET_CREATE



Esempio di Diagramma ladder SOCKET_CREATE



Esempio di Testo strutturato SOCKET_CREATE

```
SOCKET_CREATE_1(
void SOCKET_CREATE_1(BOOL Execute, USINT Type, SOCKADDR_CFG SockAddr, UDINT Timeout)
Tipo: SOCKET_CREATE, Crea socket

SOCKET_CREATE_1 (Execute_SOCKET_CREATE, Type_SOCKET_CREATE, SockAddr_SOCKET_CREATE, Timeout_SOCKET_CREATE);
Done_SOCKET_CREATE := SOCKET_CREATE. Done;
Busy_SOCKET_CREATE := SOCKET_CREATE. Busy;
Error_SOCKET_CREATE := SOCKET_CREATE. Error;
Status_SOCKET_CREATE := SOCKET_CREATE. Status;
Instance_SOCKET_CREATE := SOCKET_CREATE. Instance;
```

Risultati

Esempio di client TCP

Monitoraggio della variabile									
Variabili globali utente - Micro850		Variabili locali - Socket_TCP_Client		Variabili di sistema - Micro850		I/O - Micro850		Parole definite	
Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati			
Crt_Skt_Type01		1	N/D	1		USINT			
- Crt_Skt_Addr01			SOCKADDR_CFG			
Crt_Skt_Addr01.Port		0	N/D			UINT			
- Crt_Skt_Addr01.IPAddress			IPADDR			
Crt_Skt_Addr01.IPAddress[0]		0	N/D			USINT			
Crt_Skt_Addr01.IPAddress[1]		0	N/D			USINT			
Crt_Skt_Addr01.IPAddress[2]		0	N/D			USINT			
Crt_Skt_Addr01.IPAddress[3]		0	N/D			USINT			
▶ Crt_TimeOut01		30000	N/D			UDINT			
Crt_Busy_01			N/D			BOOL			
Crt_Error_01			N/D			BOOL			
- Crt_Status_01			SOCK_STATUS			
Crt_Status_01.ErrorID		0	N/D			USINT			
Crt_Status_01.SubErrorID		0	N/D			UINT			
Crt_Status_01.StatusBits		17	N/D			UINT			
Crt_Instance_01		1075386296	N/D			UDINT			
Crt_Sck01			N/D			BOOL			

Esempio di server TCP

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
+ SOCKET_CREATE_1						
Crt_TimeOut01		0	N/D		<input checked="" type="checkbox"/>	UDINT
- Crt_Status_01						
Crt_Status_01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Crt_Status_01.SubErrorID		0	N/D		<input type="checkbox"/>	USINT
Crt_Status_01.StatusBits		17	N/D		<input type="checkbox"/>	UINT
Crt_Skt_Type01						
Crt_Skt_Type01		1	N/D	1	<input type="checkbox"/>	USINT
- Crt_Skt_Addr01						
Crt_Skt_Addr01.Port		1400	N/D	1400	<input type="checkbox"/>	UINT
- Crt_Skt_Addr01.IPAddress						
Crt_Skt_Addr01.IPAddress[0]		0	N/D		<input type="checkbox"/>	USINT
Crt_Skt_Addr01.IPAddress[1]		0	N/D		<input type="checkbox"/>	USINT
Crt_Skt_Addr01.IPAddress[2]		0	N/D		<input type="checkbox"/>	USINT
Crt_Skt_Addr01.IPAddress[3]		0	N/D		<input type="checkbox"/>	USINT
Crt_Sck01		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Crt_instance_01		1076298724	N/D		<input type="checkbox"/>	UDINT
Crt_Error_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Crt_Busy_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL

Esempio di UDP

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
+ SOCKET_CREATE_1						
Crt_TimeOut01		20000	N/D	20000	<input checked="" type="checkbox"/>	UDINT
- Crt_Status_01						
Crt_Status_01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Crt_Status_01.SubErrorID		0	N/D		<input type="checkbox"/>	USINT
Crt_Status_01.StatusBits		17	N/D		<input type="checkbox"/>	UINT
Crt_Skt_Type01						
Crt_Skt_Type01		2	N/D	2	<input type="checkbox"/>	USINT
- Crt_Skt_Addr01						
Crt_Skt_Addr01.Port		1000	N/D	1000	<input type="checkbox"/>	UINT
- Crt_Skt_Addr01.IPAddress						
Crt_Skt_Addr01.IPAddress[0]		0	N/D		<input type="checkbox"/>	USINT
Crt_Skt_Addr01.IPAddress[1]		0	N/D		<input type="checkbox"/>	USINT
Crt_Skt_Addr01.IPAddress[2]		0	N/D		<input type="checkbox"/>	USINT
Crt_Skt_Addr01.IPAddress[3]		0	N/D		<input type="checkbox"/>	USINT
Crt_Sck01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Crt_instance_01		1076298724	N/D		<input type="checkbox"/>	UDINT
Crt_Error_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Crt_Busy_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL

Vedere anche

[Codici di errore del socket](#) a pagina 637

[Istruzioni socket](#) a pagina 603

[Tipo di dati SOCK_STATUS](#) a pagina 637

[Bit di stato per l'istruzione del socket](#) a pagina 641

[Tipo di dati SOCKADDR_CFG](#) a pagina 636

SOCKET_DELETE

Elimina un'istanza di socket creata. Per le connessioni Transmission Control Protocol (TCP), **SOCKET_DELETE** chiude anche (chiusura

passiva) la connessione prima di eliminare l'istanza. L'uscita viene aggiornata in modo sincrono dalla scansione del programma.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro820, Micro850 e Micro870. Per Micro800 Simulator, questa istruzione può essere aggiunta a un programma ma si tratta solo di un segnaposto per impedire che l'istruzione venga eliminata durante la modifica del controllore.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.



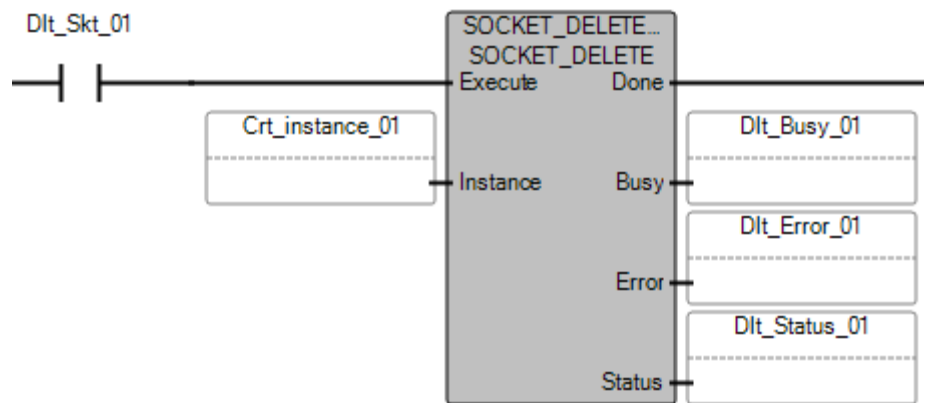
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: nessun fronte di salita rilevato.
Istanza	Ingresso	UDINT	Copia il Socket Handler restituito da un'istruzione SOCKET_CREATE o SOCKET_ACCEPT per eliminare il rispettivo socket. <ul style="list-style-type: none"> • Per i tipi di socket client TCP e UDP, copiare il Socket Handler restituito da un'istruzione SOCKET_CREATE. • Per il tipo di socket server TCP, copiare il Socket Handler restituito da un'istruzione SOCKET_ACCEPT.
Done	Uscita	BOOL	Indica se l'operazione è stata completata. TRUE: l'operazione è stata completata correttamente. FALSE: l'operazione è in corso oppure è stato generato un errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Busy	Uscita	BOOL	TRUE: l'operazione non viene terminata. FALSE: l'operazione è completata. L'output viene aggiornato in modo sincrono alla scansione del programma.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Stato	Uscita	SOCK_STATUS	Stato definito utilizzando il tipo di dati SOCK_STATUS che contiene informazioni su ErrorID, SubErrorID e StatusBits. L'output viene aggiornato in modo sincrono alla scansione del programma.

Esempio di Programmazione a blocchi funzionali SOCKET_DELETE



Esempio di Diagramma ladder SOCKET_DELETE



Esempio di Testo strutturato SOCKET_DELETE

```

SOCKET_DELETE_1 (
    void SOCKET_DELETE_1(BOOL Execute, UDINT Instance)
    Tipo: SOCKET_DELETE, Elimina socket

SOCKET_DELETE_1 (Execute_SOCKET_DELETE, Instance_SOCKET_DELETE);
Done_SOCKET_DELETE := SOCKET_DELETE. Done;
Busy_SOCKET_DELETE := SOCKET_DELETE. Busy;
Error_SOCKET_DELETE := SOCKET_DELETE. Error;
Status_SOCKET_DELETE := SOCKET_DELETE. Status;
    
```

Risultati

Nome	Alias	Valore logico	Valore fisico	Valore ini	Blocca	Tipo di dati
Dlt_Busy_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Dlt_Error_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Dlt_Skt_01		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
- Dlt_Status_01		<input type="checkbox"/>	SOCK_STATUS
Dlt_Status_01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Dlt_Status_01.SubErrorID		0	N/D		<input type="checkbox"/>	UINT
Dlt_Status_01.StatusBits		17	N/D		<input type="checkbox"/>	UINT

Vedere anche

[Istruzioni socket](#) a pagina 603

[Bit di stato per l'istruzione del socket](#) a pagina 641

[Tipo di dati SOCK_STATUS](#) a pagina 637

[Codici di errore del socket](#) a pagina 637

SOCKET_DELETEALL

Elimina tutte le istanze socket create.

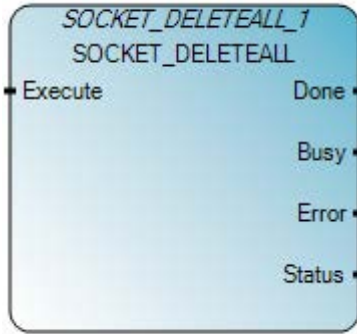
Dettagli operazione:

- Se il cavo Ethernet viene disconnesso dal controllore o l'indirizzo IP del controllore viene modificato, è possibile eseguire SOCKET_DELETEALL per eliminare tutte le istanze del socket precedentemente create.
- L'uscita viene aggiornata in modo sincrono dalla scansione del programma.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro820, Micro850 e Micro870. Per Micro800 Simulator, questa istruzione può essere aggiunta a un programma ma si tratta solo di un segnaposto per impedire che l'istruzione venga eliminata durante la modifica del controllore.

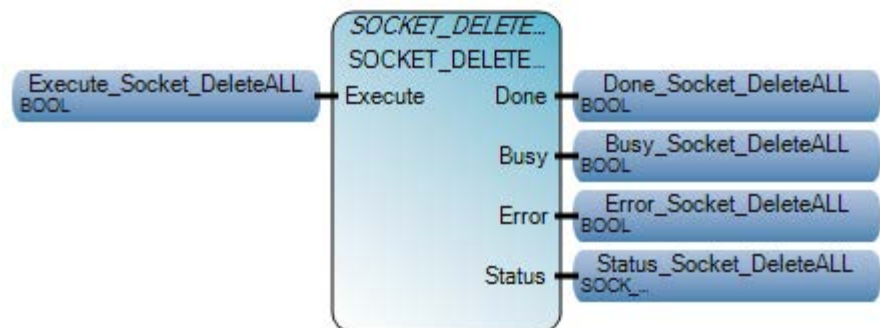
Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.



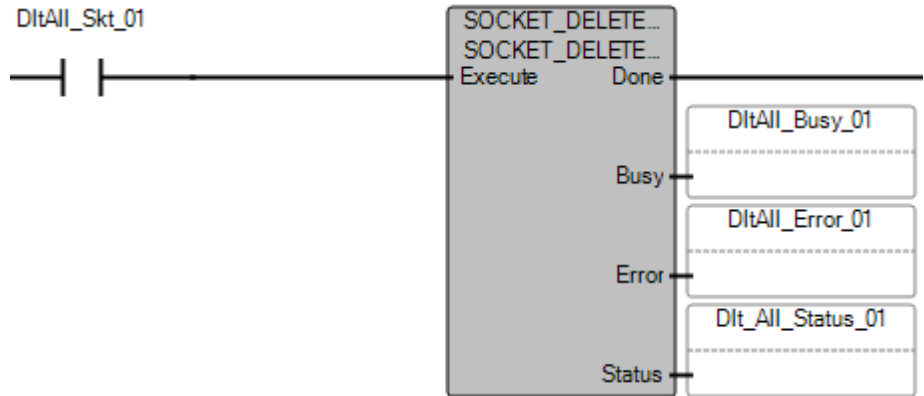
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: nessun fronte di salita rilevato.
Done	Uscita	BOOL	Indica se l'operazione è stata completata. TRUE: l'operazione è stata completata correttamente. FALSE: l'operazione è in corso oppure è stato generato un errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Busy	Uscita	BOOL	TRUE: l'operazione non viene terminata. FALSE: l'operazione è completata. L'output viene aggiornato in modo sincrono alla scansione del programma.
Errore	Uscita	BOOL	Questo campo è impostato su TRUE quando viene rilevata una condizione di errore durante l'esecuzione del blocco funzione. Per maggiori informazioni fare riferimento ai codici di errore del Socket. L'output viene aggiornato in modo sincrono alla scansione del programma.
Stato	Uscita	SOCK_STATUS	Stato definito utilizzando il tipo di dati SOCK_STATUS che contiene informazioni su ErrorID, SubErrorID e StatusBits. L'output viene aggiornato in modo sincrono alla scansione del programma.

Esempio di Programmazione a blocchi funzionali SOCKET_DELETEALL



Esempio di Diagramma ladder SOCKET_DELETEALL

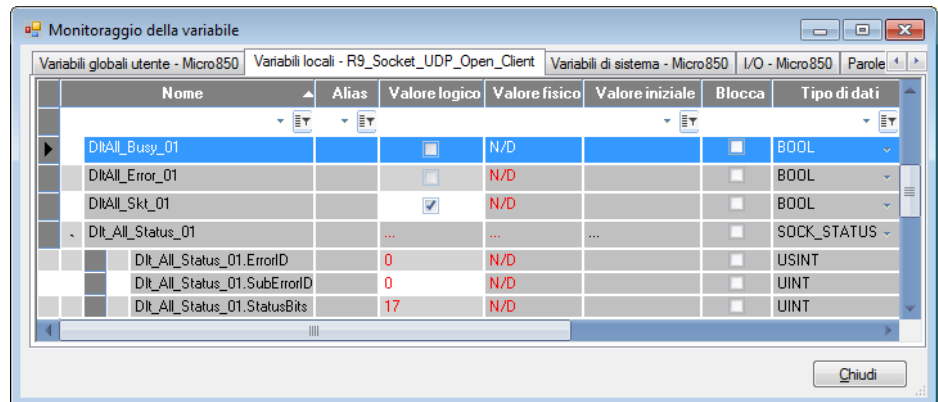


Esempio di Testo strutturato SOCKET_DELETEALL

```
SOCKET_DELETEALL_1 {
void SOCKET_DELETEALL_1(BOOL Execute)
Tipo: SOCKET_DELETEALL, Elimina tutti i socket

SOCKET_DELETEALL (Execute_SOCKET_DELETEALL);
Done_Socket_DeleteAll := SOCKET_DELETE. Done;
Busy_Socket_DeleteAll := SOCKET_DELETE. Busy;
Error_Socket_DeleteAll := SOCKET_DELETE. Error;
Status_Socket_DeleteAll := SOCKET_DELETE. Status;
```

Risultati



Vedere anche

- [Istruzioni socket](#) a pagina 603
- [Tipo di dati SOCK_STATUS](#) a pagina 637
- [Bit di stato per l'istruzione del socket](#) a pagina 641
- [Codici di errore del socket](#) a pagina 637

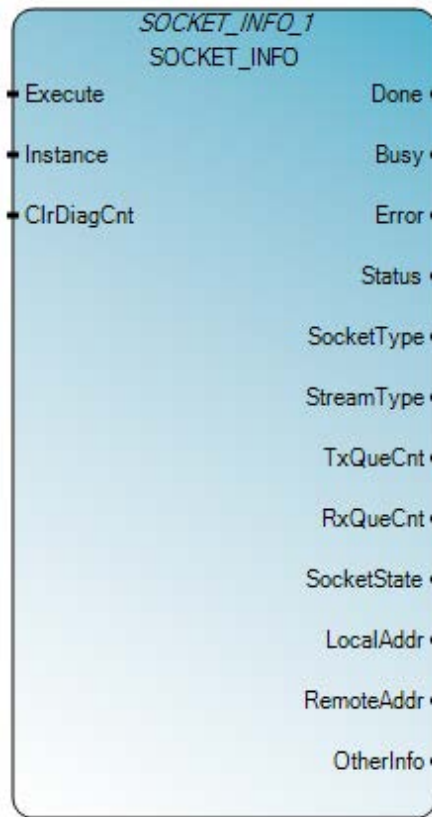
SOCKET_INFO

Restituisce informazioni per un'istanza socket come codici di errore e stato di esecuzione. Le uscite vengono aggiornate in modo asincrono dalla scansione del programma.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro820, Micro850 e Micro870. Per Micro800 Simulator, questa istruzione può essere aggiunta a un programma ma si tratta solo di un segnaposto per impedire che l'istruzione venga eliminata durante la modifica del controllore.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.



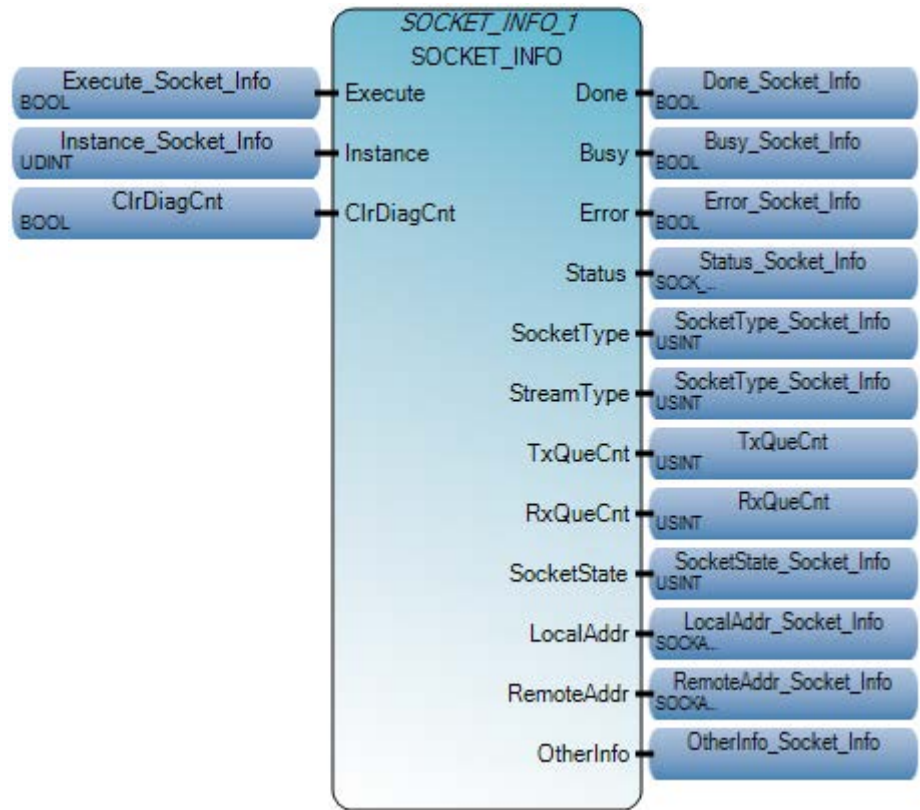
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: nessun fronte di salita rilevato.
Istanza	Ingresso	UDINT	Copia il Socket Handler restituito da un'istruzione SOCKET_CREATE o SOCKET_ACCEPT per eliminare il rispettivo socket. <ul style="list-style-type: none"> Per i tipi di socket client TCP e UDP, copiare il Socket Handler restituito da un'istruzione SOCKET_CREATE. Per il tipo di socket server TCP, copiare il Socket Handler restituito da un'istruzione SOCKET_ACCEPT. Se Instance è 0, restituisce un riepilogo di tutte le istanze Socket.

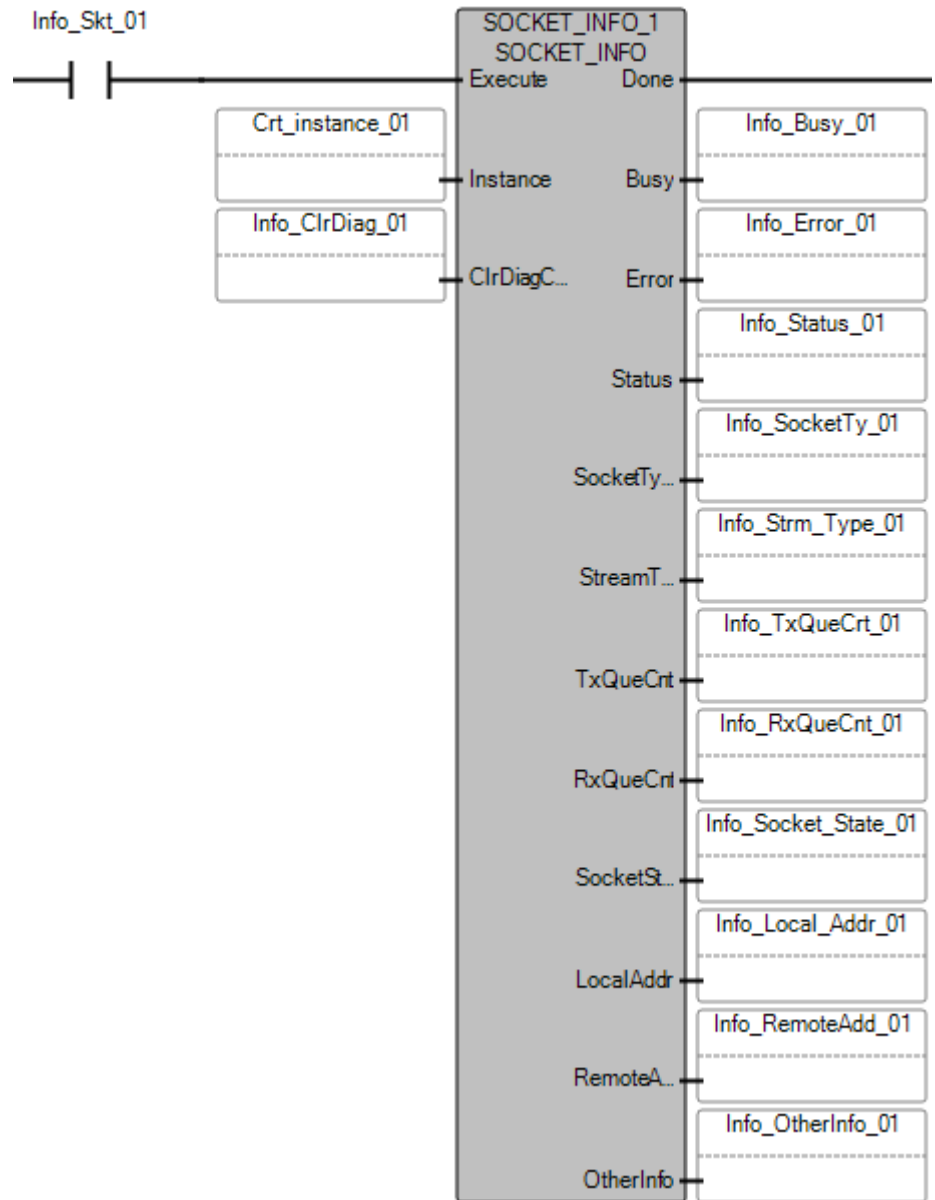
Parametro	Tipo di parametro	Tipo di dati	Descrizione
ClrDiagCnt	Ingresso	BOOL	TRUE: cancella le informazioni sul contatore di diagnostica Transmission Control Protocol (TCP) o User Datagram Protocol (UDP). FALSE: nessuna cancellazione delle informazioni del contatore TCP o UDP. Come TCP e UDP, l'indice del vettore OtherInfo da 1 a 6. Se l'Istanza è 0, cancellare l'indice del vettore OtherInfo da 7 a 14.
Done	Uscita	BOOL	Indica se l'operazione è stata completata. TRUE: l'operazione è stata completata correttamente. FALSE: l'operazione è in corso oppure è stato generato un errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Busy	Uscita	BOOL	TRUE: l'operazione non viene terminata. FALSE: l'operazione è completata. L'output viene aggiornato in modo sincrono alla scansione del programma.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Stato	Uscita	SOCK_STATUS	Stato definito utilizzando il tipo di dati SOCK_STATUS che contiene informazioni su ErrorID, SubErrorID e StatusBits. L'output viene aggiornato in modo sincrono alla scansione del programma.
SocketType	Uscita	USINT	Tipo di istanza Socket: <ul style="list-style-type: none"> • 0 - Non utilizzato • 1 - TCP • 2 - UDP Se l' Istanza Socket_Info è 0, SocketType viene visualizzato come 0.
StreamType	Uscita	USINT	Tipo di Socket Stream: <ul style="list-style-type: none"> • 0 - Nessuno • 1 - Server TCP • 2 - Client TCP Se l' Istanza Socket_Info è 0, StreamType viene visualizzato come 0.
TxQueCnt	Uscita	USINT	Numero di messaggi Tx attualmente in coda. Se l' Istanza Socket_Info è 0, TxQueCnt viene visualizzato come 0.
RxQueCnt	Uscita	USINT	Numero di messaggi Rx attualmente in coda. Se l' Istanza Socket_Info è 0, RxQueCnt viene visualizzato come 0.
SocketState	Uscita	USINT	Informazioni sullo stato dell'istruzione Socket. Per maggiori informazioni fare riferimento a Stato macchina Socket. Se l' Istanza Socket_Info è 0, SocketState viene visualizzato come 0.
LocalAddr	Uscita	SOCKETADDR_CFG	Indirizzo locale per il socket. Per maggiori informazioni fare riferimento al Tipo di dati SOCKETADDR_CFG. Se l' Istanza Socket_Info è 0, LocalAddr viene visualizzato come 0.
RemoteAddr	Uscita	SOCKETADDR_CFG	Indirizzo remoto per il socket. Per maggiori informazioni fare riferimento al Tipo di dati SOCKETADDR_CFG. RemoteAddr viene visualizzato come 0 nei casi seguenti: <ul style="list-style-type: none"> • L'Istanza Socket_Info è 0 • Connessioni User Datagram Protocol (UDP) senza SOCKET_OPEN • UDP con SOCKET_OPEN e RxFilter disattivato.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
OtherInfo		UDINT[1..15]	<p>L'Istanza socket è configurata come TCP, la descrizione dell'indice del vettore è:</p> <ul style="list-style-type: none"> ● 1 - Pacchetto inviato: Numero totale di pacchetti TCP inviati su un Socket. ● 2 - Byte inviati: Numero totale di byte TCP inviati su un Socket. ● 3 - Pacchetto ricevuto: Numero totale di pacchetti TCP ricevuti per un Socket. ● 4 - Byte ricevuti: Numero totale di byte TCP ricevuti su un Socket. ● 5 - Pacchetti ritrasmessi: Numero totale di ritrasmissioni del pacchetto TCP. ● 6 - Errori Checksum: Numero totale di pacchetti TCP con errori Checksum su un Socket. ● 7 - Stato TCP: Stato corrente di un Socket. ● (Da 8 a 11): OtherInfo non è supportato per TCP, viene visualizzato come 0. ● 12 ,13,14,15 - Visualizzato come 0. <p>Istanza Socket configurata come UDP, la descrizione dell'indice Array è:</p> <ul style="list-style-type: none"> ● 1 - Pacchetto inviato: Numero totale di pacchetti UDP inviati su un Socket. ● 2 - Byte inviati: Numero totale di byte UDP inviati su un Socket. ● 3 - Pacchetto ricevuto: Numero totale di pacchetti UDP ricevuti su un Socket. ● 4 - Byte ricevuti: Numero totale di byte UDP ricevuti su un Socket. ● 5 - Pacchetti scartati: Numero totale di pacchetti UDP ricevuti e scartati per un Socket perché è stato superato il limite di dimensione massima della coda pari a 8. ● 6 - Errori Checksum: Numero totale di pacchetti UDP con errori checksum su Socket. ● Da 7 a 15 - Visualizzato come 0. <p>Istanza Socket configurata come 0, la descrizione dell'indice Array è:</p> <ul style="list-style-type: none"> ● 1 - Numero istanze Socket disponibili. Numero massimo di socket supportati. ● 2 - Numero istanze Socket usate. Numero di socket creati correttamente. ● 3 - Numero di istanze Socket create come TCP. ● 4 - Numero di istanze Socket create come client TCP. ● 5 - Numero di istanze Socket create come server TCP. ● 6 - Numero di istanze Socket create come UDP. ● 7 - Numero di SOCKET_READ completate correttamente quando l'istanza Socket è configurata come TCP. ● 8 - numero di SOCKET_WRITE completate correttamente quando l'istanza Socket è configurata come TCP. ● 9 - numero di SOCKET_READ non riuscite quando l'istanza Socket è configurata come TCP. ● 10 - numero di SOCKET_WRITE non riuscite quando l'istanza Socket è configurata come TCP. ● 11 - numero di SOCKET_READ completate correttamente quando l'istanza Socket è configurata come UDP. ● 12 - numero di SOCKET_WRITE completate correttamente quando l'istanza Socket è configurata come UDP. ● 13 - numero di SOCKET_READ non riuscite quando l'istanza Socket è configurata come UDP. ● 14 - Numero di SOCKET_WRITE non riuscite quando l'istanza Socket è configurata come UDP. ● 15 - Visualizzato come 0.

Esempio di Programmazione a blocchi funzionali SOCKET_INFO



Esempio di Diagramma ladder SOCKET_INFO

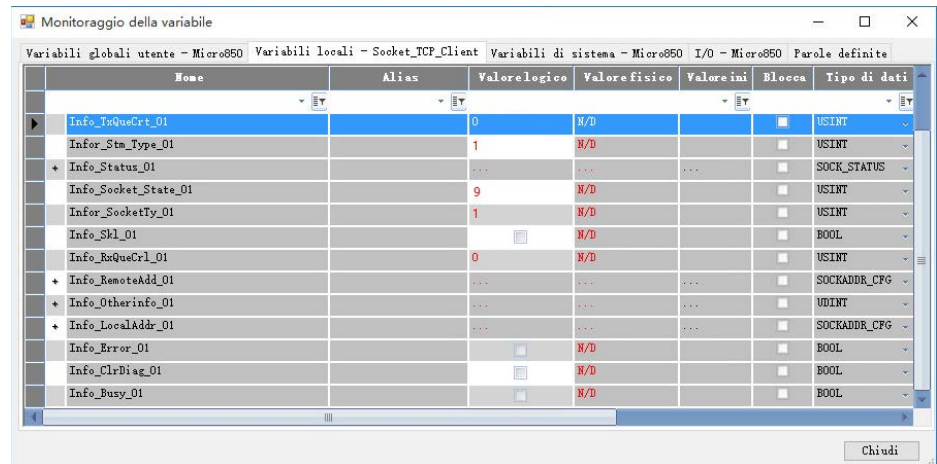


Esempio di Testo strutturato SOCKET_INFO

```
SOCKET_INFO_1 |
void SOCKET_INFO_1(BOOL Execute, UDINT Instance, BOOL ClrDiagCnt)
Tipo: SOCKET_INFO, Informazioni istanza socket
```

```
SOCKET_INFO_1 (Execute_SOCKET_INFO, Instance_SOCKET_INFO, ClrDiagC_SOCKET_INFO);
Done_SOCKET_INFO := SOCKET_INFO. Done;
Busy_SOCKET_INFO := SOCKET_INFO. Busy;
Error_SOCKET_INFO := SOCKET_INFO. Error;
Status_SOCKET_INFO := SOCKET_INFO. Status;
SocketType_SOCKET_INFO := SOCKET_INFO. SocketTy;
StreamType_SOCKET_INFO := SOCKET_INFO. StreamType;
TxQueCnt_SOCKET_INFO := SOCKET_INFO. TxQueCnt;
RxQueCnt_SOCKET_INFO := SOCKET_INFO. RxQueCnt;
SocketState_SOCKET_INFO := SOCKET_INFO. SocketState;
LocalAddr_SOCKET_INFO := SOCKET_INFO. LocalAddr;
RemoteAddr_SOCKET_INFO := SOCKET_INFO. RemoteAddr;
OtherInfo_SOCKET_INFO := SOCKET_INFO. OtherInfo;
```

Risultati



Vedere anche

- [Istruzioni socket](#) a pagina 603
- [Bit di stato per l'istruzione del socket](#) a pagina 641
- [Codici di errore del socket](#) a pagina 637
- [Tipo di dati SOCKADDR_CFG](#) a pagina 636
- [Tipo di dati SOCK_STATUS](#) a pagina 637

SOCKET_OPEN

Apri la connessione per l'indirizzo di destinazione specificato per le connessioni Transmission Control Protocol (TCP). Per le connessioni User Datagram Protocol (UDP), associa un indirizzo IP di destinazione e il numero di porta al socket specificato.

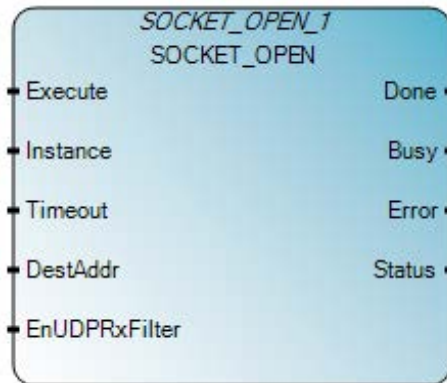
Dettagli operazioni:

- Per le connessioni User Datagram Protocol (UDP) che usano **SOCKET_OPEN**, l'indirizzo IP e il numero di porta non sono obbligatori ogni volta che i dati vengono inviati.
- Per le connessioni UDP che non usano **SOCKET_OPEN**, è necessario specificare l'indirizzo di destinazione ogni volta che **SOCKET_WRITE** viene usato per inviare i dati.
- Se si utilizza **SOCKET_READ**, oltre a ricevere i dati, viene restituito l'indirizzo del mittente. L'indirizzo dei mittenti può essere usato per inviare una risposta usando **SOCKET_WRITE**.
- Potrebbe essere restituita un'operazione **SOCKET_OPEN** prima del timeout senza creare una connessione Transmission Control Protocol (TCP). Se si verifica questa condizione, significa che il dispositivo di destinazione è in esecuzione ma non è in ascolto delle connessioni sul numero di porta specificato, **SOCKET_OPEN** restituisce un errore prima del timeout.
- Le uscite vengono aggiornate in modo asincrono dalla scansione del programma.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro820, Micro850 e Micro870. Per Micro800 Simulator, questa istruzione può essere aggiunta a un programma ma si tratta solo di un segnaposto per impedire che l'istruzione venga eliminata durante la modifica del controllore.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.

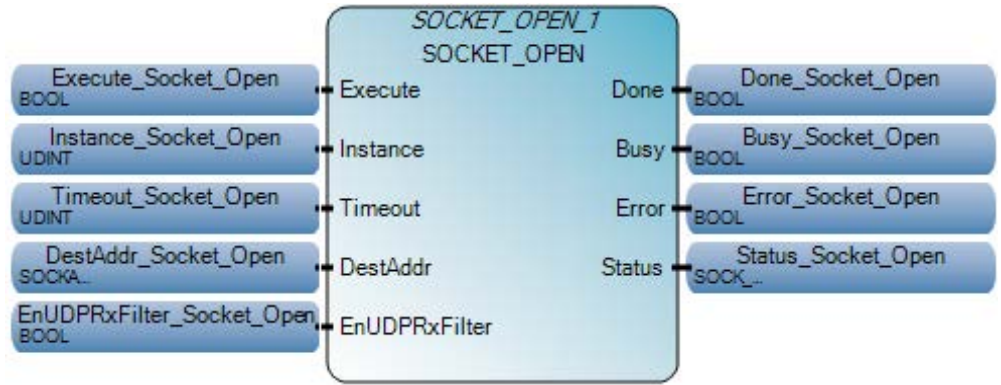


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

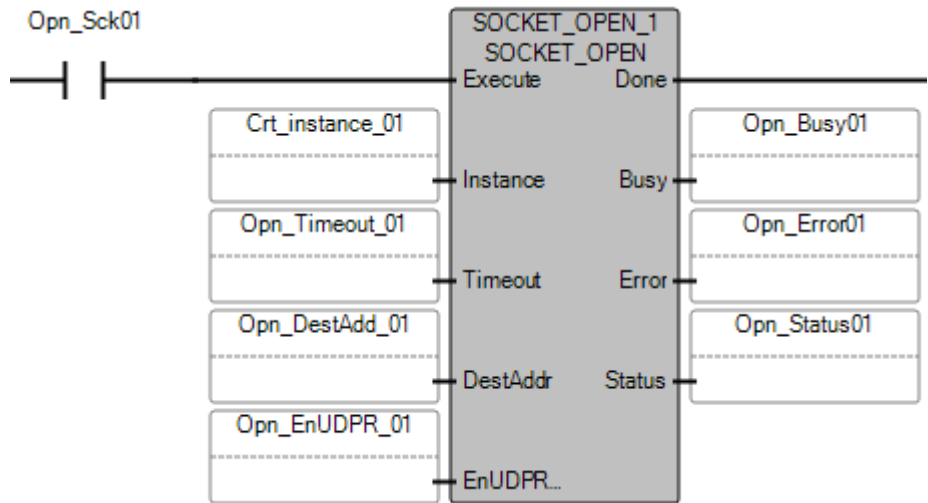
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: nessun fronte di salita rilevato.
Istanza	Ingresso	UDINT	Copiare dal Socket Handler restituito da un blocco funzione SOCKET_CREATE .

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Timeout	Ingresso	UDINT	Timeout per il blocco funzione SOCKET_OPEN. Il blocco funzione restituisce un errore se il valore di timeout è inferiore al valore minimo. Intervallo timeout: 1000 - 1800000 millisecondi Impostare Timeout su 0 per utilizzare il valore predefinito 10.000 (10 secondi).
DestAddr	Ingresso	SOCKADDR_CFG	L'indirizzo della connessione di destinazione. È necessario indicare una connessione tra l'indirizzo IP e il numero di porta dell'host remoto. I seguenti indirizzi IP non sono supportati per DestAddr: <ul style="list-style-type: none"> • Indirizzo IP automatico • Indirizzo Loopback • 0.0.0.0 • Indirizzo di trasmissione (supportato solo per TCP), indirizzi di eccezione: <ul style="list-style-type: none"> • Indirizzo multicast Class D (224.x.x.x) • Indirizzo di collegamento locale (169.254.x.x) Esempio per indirizzo IP 192.168.2.100 e Porta 12000: <ul style="list-style-type: none"> • DestAddr.IPAddress[0]=192 • DestAddr.IPAddress[1]=168 • DestAddr.IPAddress[2]=2 • DestAddr.IPAddress[3]=100 • DestAddr.Port = 12000
EnUDPRxFilter	Ingresso	BOOL	Per il socket UDP, se vengono usati SOCKET_OPEN e Abilita EnUDPRxFilter, non è necessario ogni volta un filtro del pacchetto per l'indirizzo IP specifico e il numero di porta per leggere i dati. <ul style="list-style-type: none"> • Viene creato un socket UDP con apertura. DestAddr come IP 192.168.1.157/Porta 161. Abilita EnUDPRxFilter, esegue Socket_Read. • Quando il controllore riceve dati da DestAddr (192.168.1.157 / 161), Socket_Read completa l'operazione. Se il controllore riceve dati da un altro IP o un'altra Porta, socket_Read ignora tale pacchetto e attende il pacchetto DestAddr. EnUDPRxFilter disattivato: <ul style="list-style-type: none"> • Eseguire SOCKET_READ. Il controllore riceve dati sulla porta UDP di configurazione, SOCKET_READ completa l'operazione. • L'applicazione controlla se il pacchetto in entrata arriva dal dispositivo previsto o no. • L'applicazione gestisce i filtri in base all'UDPAAddr del parametro di output SOCKET_READ.
Done	Uscita	BOOL	Indica se l'operazione è stata completata. TRUE: l'operazione è stata completata correttamente. FALSE: l'operazione è in corso oppure è stato generato un errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Busy	Uscita	BOOL	TRUE: l'operazione non viene terminata. FALSE: l'operazione è completata. L'output viene aggiornato in modo sincrono alla scansione del programma.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Stato	Uscita	SOCK_STATUS	Stato definito utilizzando il tipo di dati SOCK_STATUS che contiene informazioni su ErrorID, SubErrorID e StatusBits. L'output viene aggiornato in modo sincrono alla scansione del programma.

Esempio di Programmazione a blocchi funzionali SOCKET_OPEN



Esempio di Diagramma ladder SOCKET_OPEN



Esempio di Testo strutturato SOCKET_OPEN

```
SOCKET_OPEN_1 {
    void SOCKET_OPEN_1(BOOL Execute, UDINT Instance, UDINT Timeout, SOCKADDR_CFG DestAddr, BOOL EnUDPRxFilter)
    Tipo: SOCKET_OPEN, Apri socket

    SOCKET_OPEN_1 (Execute_SOCKET_OPEN, Instance_SOCKET_OPEN, Timeout_SOCKET_OPEN, BestAddr_SOCKET_OPEN, EnVDPR_SOCKET_OPEN)
    Done_SOCKET_OPEN := _SOCKET_OPEN. Done:
    Busy_SOCKET_OPEN := _SOCKET_OPEN. Busy:
    Error_SOCKET_OPEN := _SOCKET_OPEN. Error:
    Status_SOCKET_OPEN := _SOCKET_OPEN. Status:
}
```

Risultati

Esempio di TCP

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
Opn_Busy01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
- Opn_DestAddr_01		<input type="checkbox"/>	SOCKADDR_CFG
Opn_DestAddr_01.Port		1200	N/D		<input type="checkbox"/>	UINT
Opn_DestAddr_01.IPAddress		<input type="checkbox"/>	IPADDR
Opn_DestAddr_01.IPAddress[0]		10	N/D		<input type="checkbox"/>	USINT
Opn_DestAddr_01.IPAddress[1]		224	N/D		<input type="checkbox"/>	USINT
Opn_DestAddr_01.IPAddress[2]		69	N/D		<input type="checkbox"/>	USINT
Opn_DestAddr_01.IPAddress[3]		18	N/D		<input type="checkbox"/>	USINT
▶ Opn_EnUDPR_01		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Opn_Error_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Opn_Sck01		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
- Opn_Status01		<input type="checkbox"/>	SOCK_STATUS
Opn_Status01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Opn_Status01.SubErrorID		0	N/D		<input type="checkbox"/>	UINT
Opn_Status01.StatusBits		17	N/D		<input type="checkbox"/>	UINT
Opn_Timeout_01		20000	N/D		<input type="checkbox"/>	UDINT

Esempio di UDP

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
Opn_Busy01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
- Opn_DestAddr_01		<input type="checkbox"/>	SOCKADDR_CFG
Opn_DestAddr_01.Port		1200	N/D		<input type="checkbox"/>	UINT
Opn_DestAddr_01.IPAddress		<input type="checkbox"/>	IPADDR
Opn_DestAddr_01.IPAddress[0]		10	N/D		<input type="checkbox"/>	USINT
Opn_DestAddr_01.IPAddress[1]		224	N/D		<input type="checkbox"/>	USINT
Opn_DestAddr_01.IPAddress[2]		69	N/D		<input type="checkbox"/>	USINT
Opn_DestAddr_01.IPAddress[3]		18	N/D		<input type="checkbox"/>	USINT
Opn_EnUDPR_01		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Opn_Error_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Opn_Sck01		<input checked="" type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
▶ Opn_Status01		<input type="checkbox"/>	SOCK_STATUS
Opn_Status01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Opn_Status01.SubErrorID		0	N/D		<input type="checkbox"/>	UINT
Opn_Status01.StatusBits		17	N/D		<input type="checkbox"/>	UINT
Opn_Timeout_01		20000	N/D		<input type="checkbox"/>	UDINT

Vedere anche

[Istruzioni socket](#) a pagina 603

[Tipo di dati SOCKADDR_CFG](#) a pagina 636

[Tipo di dati SOCK_STATUS](#) a pagina 637

[Bit di stato per l'istruzione del socket](#) a pagina 641

[Codici di errore del socket](#) a pagina 637

SOCKET_READ

Legge i dati su un socket e restituisce il numero di byte specificato. Per Transmission Control Protocol (TCP), restituisce fino al numero di byte richiesti quando si ricevono dati. Per User Datagram Protocol (UDP), verrà completato quando un datagram è disponibile.

Dettagli operazione:

Il seguente comportamento **SOCKET_READ** può influenzare le comunicazioni esistenti inclusa la comunicazione non-socket.

- Se l'operazione **SOCKET_READ** non viene eseguita in sincronia con il dispositivo remoto, il controllore continua a fare in modo che il dispositivo remoto riceva pacchetti fino a quando non si verifica una delle seguenti condizioni:
 - Viene eseguita l'istruzione **SOCKET_READ**.
 - Il timeout del Socket scade.
 - Viene ricevuta l'istruzione RST da un dispositivo remoto.
 - Viene eseguita l'istruzione **SOCKET_DELETE** o **SOCKET_DELETEALL**.
 - L'utente esegue una **Modifica in modalità Run** che elimina tutte le istanze Socket create.
 - Il controllore passa dalla modalità Run alla modalità di programmazione e tutte le istanze Socket create vengono eliminate.
 - Il controllore passa dalla modalità Run alla modalità di programmazione e le informazioni sul contatore di diagnostica del socket e sul contatore di ogni Socket vengono cancellate.
- Se il valore del parametro Length o Offset viene modificato mentre l'operazione **SOCKET_READ** è in corso (BUSY = True), si verifica un errore e il pacchetto ricevuto viene ignorato.
- L'istruzione **SOCKET_READ** può restituire meno byte di quelli richiesti. RxLength contiene il numero di byte di dati ricevuti. Scrivere programmi per controllare RxLength e generare richieste di lettura aggiuntive per ricevere un intero messaggio dell'applicazione.
- Nella modalità **Modifica in modalità Run**, la modifica a un ingresso **SOCKET_READ** durante il funzionamento nello stato OCCUPATO genera un errore e il pacchetto ricevuto viene ignorato.
- Le uscite vengono aggiornate in modo asincrono dalla scansione del programma.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro820, Micro850 e Micro870. Per Micro800 Simulator, questa istruzione può essere aggiunta a un programma ma si tratta solo di un segnaposto per impedire che l'istruzione venga eliminata durante la modifica del controllore.

Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.

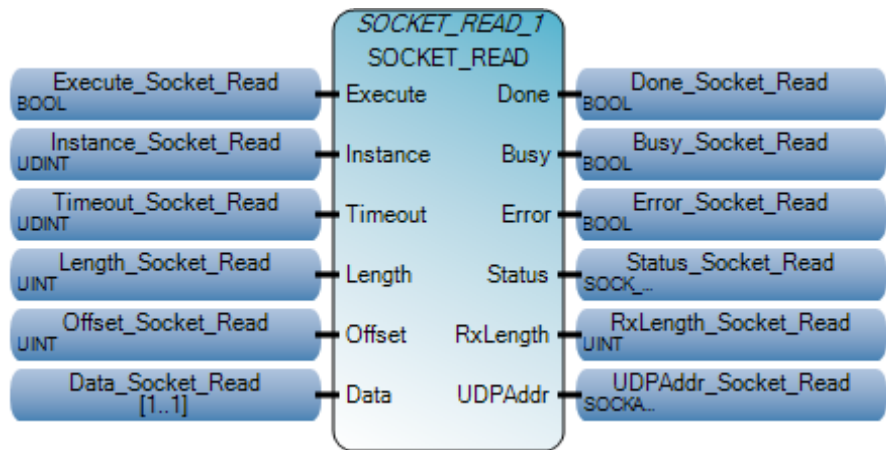


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

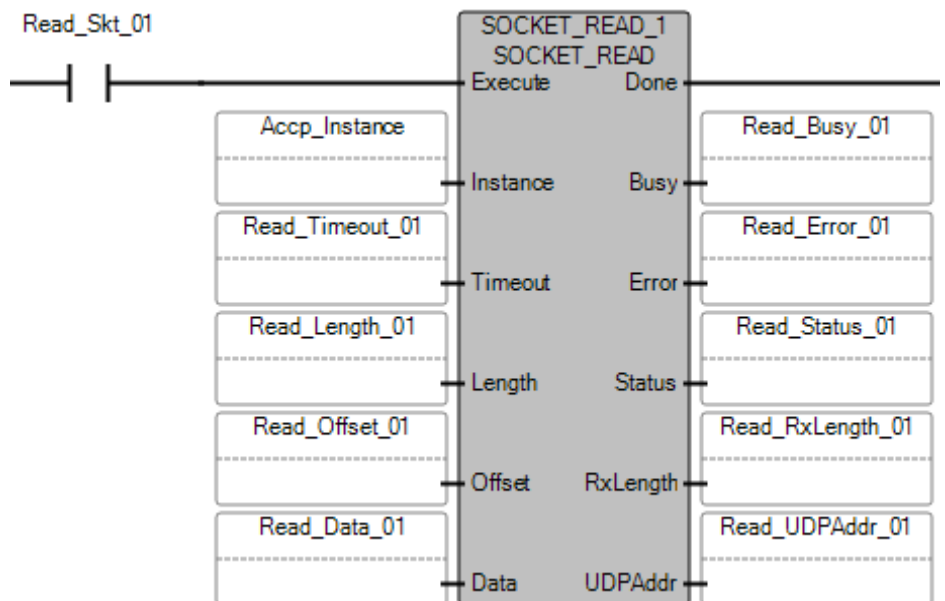
Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione istruzioni. TRUE: quando viene rilevato fronte di salita, avvia l'istruzione, ammesso che l'ultima operazione sia stata completata. FALSE: non avvia l'istruzione.
Istanza	Ingresso	UDINT	Copiare dal Socket Handler restituito da un'istruzione SOCKET_CREATE o SOCKET_ACCEPT . <ul style="list-style-type: none"> • Per i tipi di socket client TCP e UDP, copiare il Socket Handler restituito da un'istruzione SOCKET_CREATE. • Per il tipo di socket server TCP, copiare il Socket Handler restituito da un'istruzione SOCKET_ACCEPT.
Timeout	Ingresso	UDINT	Timeout per SOCKET_READ . Il blocco istruzione restituisce un errore se il valore di timeout è inferiore al valore minimo. Intervallo timeout: 1000-86400000 millisecondi Impostare il timeout su 0 per usare il valore predefinito 10.000 (10 secondi).
Lunghezza	Ingresso	UINT	Definisce il numero di byte da leggere. Controllare RxLength per il numero effettivo di byte letti. SOCKET_READ può restituire meno byte di quelli richiesti. Supporta fino a 256 byte.
Offset	Ingresso	UNIT	Offset nel vettore di Dati . Avviare la lettura dei dati letti da questa posizione.
Dati	Uscita	USINT[1..1]	Un vettore usato per memorizzare i dati letti da SOCKET_READ . <ul style="list-style-type: none"> • La dimensione del vettore di dati deve essere \geq (Offset + Length). • Il vettore di dati può essere maggiore di Length del socket letto. L'output viene aggiornato in modo sincrono alla scansione del programma.
Done	Uscita	BOOL	Indica se l'operazione è stata completata. TRUE: l'operazione è stata completata correttamente. FALSE: l'operazione è in corso oppure è stato generato un errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Busy	Uscita	BOOL	TRUE: l'operazione non viene terminata. FALSE: l'operazione è completata. L'output viene aggiornato in modo sincrono alla scansione del programma.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Stato	Uscita	SOCK_STATUS	Stato definito utilizzando il tipo di dati SOCK_STATUS che contiene informazioni su ErrorID, SubErrorID e StatusBits. L'output viene aggiornato in modo sincrono alla scansione del programma.
RxLength	Uscita	UNIT	Contiene il numero di byte di dati ricevuti.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
UDPAddr	Uscita	SOCKADDR_CFG	<p>L'indirizzo del dispositivo che invia i dati User Datagram Protocol (UDP).</p> <p>L'esempio definisce un UDPAddr di 192.168.2.100 e Porta 12000: UDPAddr.IPAddress[0]=192 UDPAddr.IPAddress[1]=168 UDPAddr.IPAddress[2]=2 UDPAddr.IPAddress[3]=100 UDPAddr.Port = 12000</p> <p>Per Transmission Control Protocol (TCP), questa struttura non viene usata e conterrà tutti zero. La connessione TCP trasmette le informazioni di indirizzo remoto.</p>

Esempio di Programmazione a blocchi funzionali SOCKET_READ



Esempio di Diagramma ladder SOCKET_READ



Esempio di Testo strutturato SOCKET_READ

```
SOCKET_READ_1(  
void SOCKET_READ_1(BOOL Execute, UDINT Instance, UDINT Timeout, UINT Length, UINT Offset, USINT[1..1] Data)  
Tipo: SOCKET_READ, Leggi socket  
  
SOCKET_READ_1 (Execute_SOCKET_READ, Instance_SOCKET_READ, Timeout_SOCKET_READ, Length_SOCKET_READ, Offset_SOCKET_READ, Data_SOCKET_READ)  
Nome_SOCKET_READ := SOCKET_READ; Descr;  
Busy_SOCKET_READ := SOCKET_READ.Busy;  
Error_SOCKET_READ := SOCKET_READ.Error;  
Status_SOCKET_READ := SOCKET_READ.Status;  
RxLength_SOCKET_READ := SOCKET_READ.RxLength;  
UDFAddr_SOCKET_READ := SOCKET_READ.UDFAddr;
```

Risultati

Esempio di TCP:

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
Read_Busy_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Read_Error_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Read_Length_01		256	N/D		<input type="checkbox"/>	UINT
Read_Offset_01		0	N/D		<input type="checkbox"/>	UINT
Read_RxLength_01		256	N/D		<input type="checkbox"/>	UINT
Read_Status_01		<input checked="" type="checkbox"/>	SOCK_STATUS
Read_Status_01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Read_Status_01.SubErrorID		0	N/D		<input type="checkbox"/>	USINT
Read_Status_01.StatusBits		16	N/D		<input type="checkbox"/>	UINT
Read_Timeout_01		20000	N/D		<input type="checkbox"/>	UDINT
Read_UDPAddr_01		<input type="checkbox"/>	SOCKADDR_CF
Read_UDPAddr_01.Port		1900	N/D		<input type="checkbox"/>	UINT
Read_UDPAddr_01.IPAddress		<input type="checkbox"/>	IPADDR
Read_UDPAddr_01.IPAddress[0]		10	N/D		<input type="checkbox"/>	USINT
Read_UDPAddr_01.IPAddress[1]		224	N/D		<input type="checkbox"/>	USINT
Read_UDPAddr_01.IPAddress[2]		68	N/D		<input type="checkbox"/>	USINT
Read_UDPAddr_01.IPAddress[3]		221	N/D		<input type="checkbox"/>	USINT

Esempio di UDP:

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
Read_Busy_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Read_Error_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Read_Length_01		256	N/D		<input type="checkbox"/>	UINT
Read_Offset_01		0	N/D		<input type="checkbox"/>	UINT
Read_RxLength_01		256	N/D		<input type="checkbox"/>	UINT
Read_Status_01		<input checked="" type="checkbox"/>	SOCK_STATUS
Read_Status_01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Read_Status_01.SubErrorID		0	N/D		<input type="checkbox"/>	USINT
Read_Status_01.StatusBits		17	N/D		<input type="checkbox"/>	UINT
Read_Timeout_01		3000	N/D		<input type="checkbox"/>	UDINT
Read_UDPAddr_01		<input type="checkbox"/>	SOCKADDR_CF
Read_UDPAddr_01.Port		1000	N/D		<input type="checkbox"/>	UINT
Read_UDPAddr_01.IPAddress		<input type="checkbox"/>	IPADDR
Read_UDPAddr_01.IPAddress[0]		10	N/D		<input type="checkbox"/>	USINT
Read_UDPAddr_01.IPAddress[1]		224	N/D		<input type="checkbox"/>	USINT
Read_UDPAddr_01.IPAddress[2]		68	N/D		<input type="checkbox"/>	USINT
Read_UDPAddr_01.IPAddress[3]		241	N/D		<input type="checkbox"/>	USINT

Vedere anche

[Codici di errore del socket](#) a pagina 637

[Istruzioni socket](#) a pagina 603

[Bit di stato per l'istruzione del socket](#) a pagina 641

[Tipo di dati SOCK_STATUS](#) a pagina 637

[Tipo di dati SOCKADDR_CFG](#) a pagina 636

SOCKET_WRITE

Invia dati su un socket.

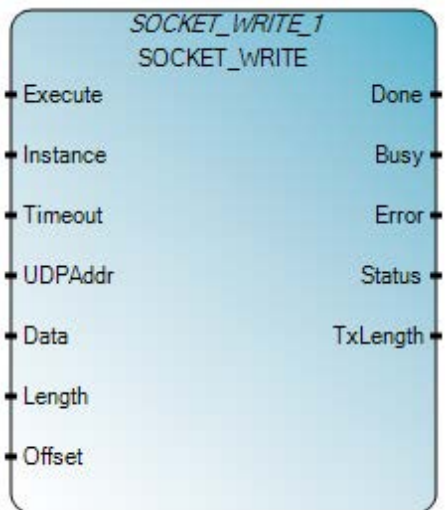
Dettagli operazione:

- **SOCKET_WRITE** prova a inviare il numero richiesto di byte di dati specificato nel parametro Length. Quando l'operazione di invio viene completata, **SOCKET_WRITE** restituisce il numero di byte di dati scritti al parametro TxLength.
- L'output viene aggiornato in modo asincrono rispetto alla scansione del programma.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione si applica ai controllori Micro820, Micro850 e Micro870. Per Micro800 Simulator, questa istruzione può essere aggiunta a un programma ma si tratta solo di un segnaposto per impedire che l'istruzione venga eliminata durante la modifica del controllore.

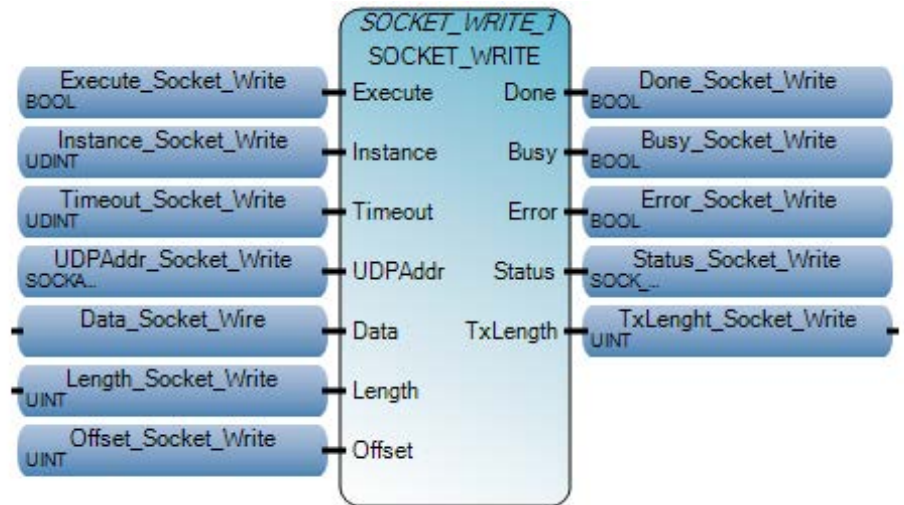
Se l'istruzione è applicata al controllore simulato (2080-LC50-48QWB-SIM) le uscite vengono sempre ripristinate.



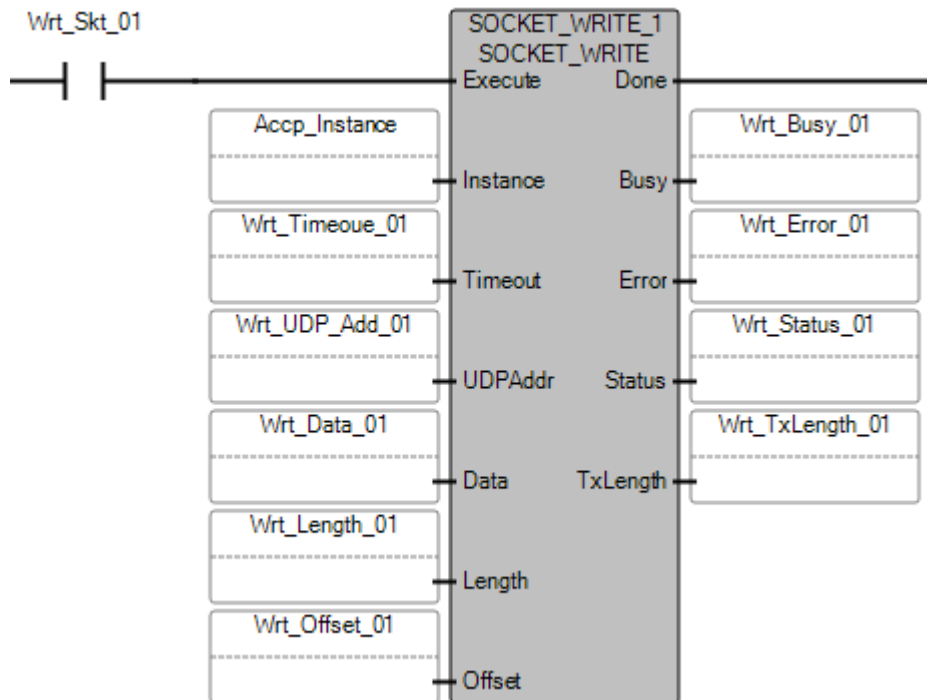
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
Execute	Ingresso	BOOL	Abilitazione del blocco istruzione. TRUE: quando viene rilevato fronte di salita, avvia il blocco istruzione ammesso che l'ultima operazione sia stata completata. FALSE: nessun fronte di salita rilevato.
Istanza	Ingresso	UDINT	Copiare il Socket Handler restituito da un'istruzione SOCKET_CREATE o SOCKET_ACCEPT . <ul style="list-style-type: none"> Per i tipi di socket client TCP e UDP, copiare il Socket Handler restituito da un'istruzione SOCKET_CREATE. Per il tipo di socket server TCP, copiare dal Socket Handler restituito da un'istruzione SOCKET_ACCEPT.
Timeout	Ingresso	UDINT	Timeout per istanze SOCKET_WRITE . L'istruzione restituisce un errore se il valore di timeout è inferiore al valore minimo. Intervallo timeout: 1000-1800000 millisecondi Impostare il timeout su 0 per usare il valore predefinito 10000 (10 secondi).
UDPAddr	Ingresso	SOCKADDR_CFG	L'indirizzo di destinazione UDP su cui vengono scritti i dati se non sono state eseguite istruzioni SOCKET_OPEN dopo la creazione del socket. Per TCP, o quando l'istruzione SOCKET_OPEN è stata eseguita per UDP, questa struttura non viene usata e deve contenere tutti zero. La connessione TCP e l'istruzione SOCKET_OPEN per UDP comunica tutte le informazioni sugli indirizzi remoti. Esempio per un UDPAddr di 192.168.2.100 e Porta 12000: <pre> UDPAddr.IPAddress[0]=192 UDPAddr.IPAddress[1]=168 UDPAddr.IPAddress[2]=2 UDPAddr.IPAddress[3]=100 UDPAddr.Port = 12000 </pre> Usare il tipo di dati SOCKADDR_CFG per definire UDPAddr.
Dati	Ingresso	USINT[1..1]	Un vettore usato per memorizzare i dati scritti nella relativa istanza socket usando l'istruzione SOCKET_WRITE . <ul style="list-style-type: none"> La dimensione del vettore di dati deve essere \geq (Offset + Length). Il vettore di dati può essere maggiore della lunghezza di SOCKET_WRITE.
Lunghezza	Ingresso	UINT	Il numero di byte di dati da scrivere. Il valore massimo è 256 byte.
Offset	Ingresso	UNIT	Offset nel vettore di Dati. I dati scritti da SOCKET_WRITE iniziano da questa posizione.
Done	Uscita	BOOL	Indica se l'operazione è stata completata. TRUE: l'operazione è stata completata correttamente. FALSE: l'operazione è in corso oppure è stato generato un errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Busy	Uscita	BOOL	TRUE: l'operazione non viene terminata. FALSE: l'operazione è completata. L'output viene aggiornato in modo sincrono alla scansione del programma.
Errore	Uscita	BOOL	Indica che si è verificato un errore. TRUE - È stato rilevato un errore. FALSE - Nessun errore. L'output viene aggiornato in modo sincrono alla scansione del programma.
Stato	Uscita	SOCK_STATUS	Stato definito utilizzando il tipo di dati SOCK_STATUS che contiene informazioni su ErrorID, SubErrorID e StatusBits. L'output viene aggiornato in modo sincrono alla scansione del programma.
TxLength	Uscita	UNIT	Il numero di byte dei dati scritti.

Esempio di Programmazione a blocchi funzionali SOCKET_WRITE



Esempio di Diagramma ladder SOCKET_WRITE



Esempio di Testo strutturato SOCKET_WRITE

```

SOCKET_WRITE_1
void SOCKET_WRITE_1(BOOL Execute, UDINT Instance, UDINT Timeout, SOCKADDR_CFG UDPAddr, USINT(1..1)Data, UINT Length, UINT Offset)
Tipo: SOCKET_WRITE, Scrivi socket

SOCKET_WRITE_1(Execute_SOCKET_WRITE, Instance_SOCKET_WRITE, Timeout_SOCKET_WRITE, UDPAddr_SOCKET_WRITE, Length_SOCKET_WRITE, Offset_SOCKET_WRITE)
Done_SOCKET_WRITE := SOCKET_WRITE.Done;
Busy_SOCKET_WRITE := SOCKET_WRITE.Busy;
Error_SOCKET_WRITE := SOCKET_WRITE.Error;
Status_SOCKET_WRITE := SOCKET_WRITE.Status;
TxLength_SOCKET_WRITE := SOCKET_WRITE.TxLength;
    
```

Risultati

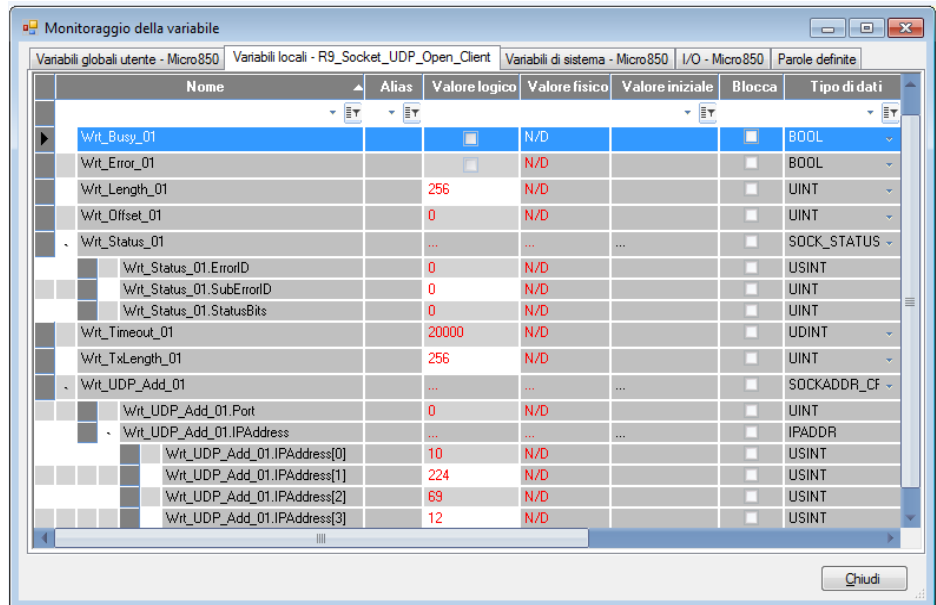
Esempio di TCP:

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
Writ_Busy_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Writ_Error_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Writ_Length_01		256	N/D		<input type="checkbox"/>	UINT
Writ_Offset_01		0	N/D		<input type="checkbox"/>	UINT
Writ_Status_01		<input checked="" type="checkbox"/>	SOCK_STATUS
Writ_Status_01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Writ_Status_01.SubErrorID		0	N/D		<input type="checkbox"/>	UINT
Writ_Status_01.StatusBits		17	N/D		<input type="checkbox"/>	UINT
Writ_Timeout_01		20000	N/D		<input type="checkbox"/>	UDINT
Writ_TxLength_01		256	N/D		<input type="checkbox"/>	UINT
Writ_UDP_Add_01		<input type="checkbox"/>	SOCKADDR_CF
Writ_UDP_Add_01.Port		1800	N/D		<input type="checkbox"/>	UINT
Writ_UDP_Add_01.IPAddress		<input type="checkbox"/>	IPADDR
Writ_UDP_Add_01.IPAddress[0]		10	N/D		<input type="checkbox"/>	USINT
Writ_UDP_Add_01.IPAddress[1]		224	N/D		<input type="checkbox"/>	USINT
Writ_UDP_Add_01.IPAddress[2]		68	N/D		<input type="checkbox"/>	USINT
Writ_UDP_Add_01.IPAddress[3]		239	N/D		<input type="checkbox"/>	USINT

Esempio di UDP senza apertura:

Nome	Alias	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
Writ_Busy_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Writ_Error_01		<input type="checkbox"/>	N/D		<input type="checkbox"/>	BOOL
Writ_Length_01		256	N/D		<input type="checkbox"/>	UINT
Writ_Offset_01		0	N/D		<input type="checkbox"/>	UINT
Writ_Status_01		<input checked="" type="checkbox"/>	SOCK_STATUS
Writ_Status_01.ErrorID		0	N/D		<input type="checkbox"/>	USINT
Writ_Status_01.SubErrorID		0	N/D		<input type="checkbox"/>	UINT
Writ_Status_01.StatusBits		17	N/D		<input type="checkbox"/>	UINT
Writ_Timeout_01		20000	N/D		<input type="checkbox"/>	UDINT
Writ_TxLength_01		256	N/D		<input type="checkbox"/>	UINT
Writ_UDP_Add_01		<input type="checkbox"/>	SOCKADDR_CF
Writ_UDP_Add_01.Port		1000	N/D		<input type="checkbox"/>	UINT
Writ_UDP_Add_01.IPAddress		<input type="checkbox"/>	IPADDR
Writ_UDP_Add_01.IPAddress[0]		10	N/D		<input type="checkbox"/>	USINT
Writ_UDP_Add_01.IPAddress[1]		224	N/D		<input type="checkbox"/>	USINT
Writ_UDP_Add_01.IPAddress[2]		68	N/D		<input type="checkbox"/>	USINT
Writ_UDP_Add_01.IPAddress[3]		253	N/D		<input type="checkbox"/>	USINT

Esempio di UDP con apertura:



Vedere anche

- [Istruzioni socket](#) a pagina 603
- [Bit di stato per l'istruzione del socket](#) a pagina 641
- [Tipo di dati SOCKADDR_CFG](#) a pagina 636
- [Tipo di dati SOCK_STATUS](#) a pagina 637
- [Codici di errore del socket](#) a pagina 637

Tipo di dati SOCKADDR_CFG La tabella seguente descrive il tipo di dati **SOCKADDR_CFG**.

Parametri	Tipo di dati	Descrizione
Porta	UINT	Specificare un numero di porta locale in cui un'applicazione è in ascolto e riceve i dati.
Indirizzo IP[4]	USINT	Specificare un indirizzo IP. Esempio di indirizzo IP 192.168.2.100: <ul style="list-style-type: none"> • IPAddress[0]=192 • IPAddress[1]=168 • IPAddress[2]=2 • IPAddress[3]=100

Vedere anche

- [Codici di errore del socket](#) a pagina 637
- [Tipo di dati SOCK_STATUS](#) a pagina 637

Tipo di dati SOCK_STATUS

La tabella seguente descrive il tipo di dati **SOCK_STATUS**.

Parametri	Tipo di dati	Descrizione
ErrorID	USINT	Valore codice di errore.
SubErrorID	UINT	Valore codice di errore secondario.
StatusBits	UINT	Bit dello stato di esecuzione per le istruzioni del socket.

Vedere anche

[Codici di errore del socket](#) a pagina 637

[Tipo di dati SOCKADDR_CFG](#) a pagina 636

[Bit di stato per l'istruzione del socket](#) a pagina 641

Codici di errore del socket

La tabella seguente descrive i codici di errore dello stato per le istruzioni del socket:

Codice ErrorID	Codice SubErrorID	Descrizione errore	Azione correttiva
0		L'istruzione del socket ha completato correttamente l'operazione.	
1		L'istruzione del socket è in sospeso.	
2		L'istanza del socket non è disponibile.	La conferma dell'istanza del socket non è stata eliminata o è stato superato il valore di timeout.
3		L'operazione SOCKET_DELETEALL è in corso.	Per terminare, attendere il completamento dell'operazione di SOCKET_DELETEALL in attesa.
4	1	Parametro non valido, canale non valido.	Nessuna azione. Riservato per uso futuro.
4	2	Parametro non valido, indirizzo IP del socket non valido. <ul style="list-style-type: none"> • L'errore si verifica quando SOCKET_CREATE contiene un SockAddr con valori diversi da zero. • SOCKET_OPEN o SOCKET_WRITE vengono eseguiti con un indirizzo IP di destinazione non valido. Gli indirizzi IP di destinazione non validi sono: <ul style="list-style-type: none"> • Indirizzo IP automatico • 0.0.0.0 • Indirizzo loopback (127.x.x.x) • Indirizzo Multicast Class D (224.x.x.x) • Indirizzo di collegamento locale (169.254.x.x) • Indirizzo di trasmissione (applicabile solo per l'istanza del socket TCP) 	Sostituire con un indirizzo IP valido.

Codice ErrorID	Codice SubErrorID	Descrizione errore	Azione correttiva
4	3	Parametro non valido, indirizzo della porta del socket non valido. L'errore si verifica quando le seguenti porte vengono specificate in SOCKET_CREATE , SOCKET_OPEN o SOCKET_WRITE : <ul style="list-style-type: none"> • Porte TCP <ul style="list-style-type: none"> • 44818 - Ethernet/IP • 502 - ModbusTCP • 67 - Server DHCP • 68 - Client DHCP • 0 - Porta non valida • Porte UDP <ul style="list-style-type: none"> • 2222 - Ethernet/IP • 67 - Server DHCP • 68 - Client DHCP • 0 - Porta non valida 	Modificare l'indirizzo della porta.
4	4	Parametro non valido, tipo di socket non valido. I tipi di socket validi sono: <ul style="list-style-type: none"> • TCP - 1 • UDP - 2 	Modificare il tipo di socket.
4	5	Parametro non valido, valore di timeout del socket non valido. I valori di timeout validi sono: <ul style="list-style-type: none"> • SOCKET_CREATE, SOCKET_ACCEPT e SOCKET_READ: <ul style="list-style-type: none"> • 0 (predefinito) o qualsiasi valore tra 1000 e 86400000 ms. • SOCKET_OPEN e SOCKET_WRITE: <ul style="list-style-type: none"> • 0 (predefinito) o qualsiasi valore tra 1000 e 86400000 ms. 	Modificare il valore di timeout con un valore valido.
4	6	Parametro non valido, istanza del socket non valida. <ul style="list-style-type: none"> • Tipo di socket del server TCP: <ul style="list-style-type: none"> • Utilizzare l'istanza del socket restituita in SOCKET_ACCEPT per SOCKET_WRITE, SOCKET_READ, SOCKET_DELETE e SOCKET_INFO. • Socket UDP e tipi di socket del client TCP: <ul style="list-style-type: none"> • Utilizzare l'istanza del socket restituita in SOCKET_CREATE per SOCKET_OPEN, SOCKET_WRITE, SOCKET_READ, SOCKET_DELETE e SOCKET_INFO. 	Utilizzare il numero dell'istanza restituita nelle istruzioni SOCKET_CREATE e SOCKET_ACCEPT dopo il completamento dell'esecuzione.
4	7	Parametro non valido, lunghezza dell'array non valida.	Aumentare la dimensione dell'array utilizzato per contenere i dati di lettura e scrittura del socket.
4	8	Parametro non valido, dimensione dell'array non valida.	Utilizzare l'array unidimensionale per contenere i dati di lettura e scrittura del socket.
5		Il socket richiede l'annullamento dell'operazione. Si verifica un errore quando: <ul style="list-style-type: none"> • Il collegamento Ethernet è disattivato o il cavo Ethernet è disconnesso • L'operazione SOCKET_DELETE viene eseguita quando l'operazione SOCKET_OPEN, SOCKET_ACCEPT, SOCKET_READ o SOCKET_WRITE è in corso. • L'operazione Modifica in modalità Run viene eseguita quando l'operazione SOCKET_OPEN, SOCKET_ACCEPT, SOCKET_READ o SOCKET_WRITE è in corso. • La collisione dell'indirizzo IP rilevata quando l'operazione SOCKET_OPEN, SOCKET_ACCEPT, SOCKET_READ o SOCKET_WRITE è in corso. 	Riavviare l'operazione del socket in base al tipo di socket. Consultare lo stato della macchina corrispondente per riavviare l'operazione del socket.
6	1	Sequenza del socket non valida, operazione apertura socket in corso o connesso. Si verifica un errore quando: <ul style="list-style-type: none"> • SOCKET_OPEN è in corso (stato BUSY) e l'utente esegue un'altra istanza SOCKET_OPEN con la stessa istanza del socket. • SOCKET_OPEN è in corso (stato BUSY) e l'utente esegue SOCKET_ACCEPT con la stessa istanza del socket. 	Utilizzare l'esecuzione unidirezionale SOCKET_OPEN per l'istanza del socket corrispondente. Non eseguire l'operazione SOCKET_ACCEPT per la stessa istanza del socket.

Codice ErrorID	Codice SubErrorID	Descrizione errore	Azione correttiva
6	2	Sequenza del socket non valida, operazione accettazione socket in corso o connesso. Si verifica un errore quando: <ul style="list-style-type: none"> • SOCKET_ACCEPT è in corso (stato BUSY) e l'utente esegue un'altra istanza SOCKET_ACCEPT con la stessa istanza del socket. • SOCKET_ACCEPT è in corso (stato BUSY) e l'utente esegue SOCKET_OPEN con la stessa istanza del socket. 	Utilizzare l'esecuzione unidirezionale SOCKET_ACCEPT per l'istanza del socket corrispondente. Non eseguire l'operazione SOCKET_OPEN per la stessa istanza del socket.
6	3	Sequenza del socket non valida, socket configurato come client TCP.	
6	4	Sequenza del socket non valida, socket configurato come server TCP.	
6	5	Sequenza del socket non valida, il socket è connesso.	Verificare che il socket non sia ancora connesso con un dispositivo remoto prima di eseguire nuovamente SOCKET_OPEN o SOCKET_ACCEPT .
6	6	Sequenza del socket non valida, socket configurato come UDP.	Verificare che le istanze del socket UDP non vengano utilizzate con SOCKET_ACCEPT .
6	7	Sequenza del socket non valida, il socket non è connesso.	Confermare che il socket sia connesso alla destinazione.
7		Timeout dell'istanza del socket.	Di conseguenza, configurare i valori di timeout per SOCKET_ACCEPT , SOCKET_OPEN , SOCKET_READ e SOCKET_WRITE .
8		Il modulo del socket non viene inizializzato.	Verificare che il collegamento Ethernet sia abilitato o che il cavo Ethernet del controllore sia connesso alla rete o risolvere il conflitto dell'indirizzo IP del controllore.
9	1	Errore irreversibile dell'istruzione del socket, istanza del socket mancante.	
9	2	Errore irreversibile dell'istruzione del socket, istanza del socket non valida.	
9	3	Errore irreversibile dell'istruzione del socket, istanza del socket di blocco non valida.	
9	4	Errore irreversibile dell'istruzione del socket, tipo di socket non valido.	
9	5	Errore irreversibile dell'istruzione del socket, gestore annullamento mancante.	
10	1	Errore di elaborazione del socket in background, indirizzo in uso.	
10	2	Errore di elaborazione del socket in background, UDP ha ricevuto un pacchetto che supera la lunghezza del socket di lettura.	Verificare che la lunghezza SOCKET_READ sia maggiore o uguale alla dimensione del pacchetto ricevuto. È consigliata la dimensione di lettura massima. La dimensione massima per SOCKET_READ è 256.
10	3	Errore di elaborazione del socket in background, TCP ha ricevuto un pacchetto che supera la lunghezza del socket di lettura.	Verificare che la lunghezza SOCKET_READ sia maggiore o uguale alla dimensione del pacchetto ricevuto. Dato che il client potrebbe inviare dati fino alla dimensione massima di 256, 256 è il valore consigliato.
10	4	Errore di elaborazione del socket in background, RST ricevuto o Disconnessione dal dispositivo remoto.	Riavviare il server o il client TCP del controllore come mostrato nel diagramma dello stato. Verificare il dispositivo di destinazione che ha inviato l'RST e assicurarsi che la destinazione venga riavviata con lo stato corretto.
10	5	Errore di elaborazione del socket in background, pacchetto UDP ricevuto da un dispositivo diverso, pacchetto scartato. Riservato per uso futuro.	
10	6	Errore di elaborazione del socket in background, la coda è piena. Si verifica un errore quando: <ul style="list-style-type: none"> • SOCKET_READ o SOCKET_WRITE eseguono quattro istruzioni per la stessa istanza del socket nella stessa scansione. • Eseguire un SOCKET_READ quando le quattro istruzioni SOCKET_READ sono nello stato BUSY per stessa istanza del socket. 	Attendere la disponibilità della coda per SOCKET_READ e SOCKET_WRITE per le rispettive istanze del socket.
10	7	Errore di elaborazione del socket in background, modifica al parametro SOCKET_READ (Length, Offset, Data Array Size e Data Array Variable).	Quando lo stato SOCKET_READ è BUSY , non modificare i parametri di ingresso.
11	1	Errore irreversibile di elaborazione del socket in background, puntatore dati mancanti.	

Codice ErrorID	Codice SubErrorID	Descrizione errore	Azione correttiva
11	2	Errore irreversibile di elaborazione del socket in background, puntatore sessioni mancanti.	
11	3	Errore irreversibile di elaborazione del socket in background, puntatore socket TCP o UDP non valido.	
11	4	Errore irreversibile di elaborazione del socket in background, tipo di istanza del socket non valido.	
11	5	Errore irreversibile di elaborazione del socket in background, istanza del socket mancante.	
11	6	Errore irreversibile di elaborazione del socket in background, istanza del socket non valida.	
11	7	Errore irreversibile di elaborazione del socket in background, stato del socket non valido.	
11	8	Errore irreversibile di elaborazione del socket in background, tipo di socket non valido.	
11	9	Errore irreversibile di elaborazione del socket in background, errore di eliminazione del TCP.	
11	10	Errore irreversibile di elaborazione del socket in background, errore di eliminazione dell'UDP.	
128	1	Nessun pacchetto disponibile per il messaggio di disconnessione.	
128	2	Spazio insufficiente per anteporre l'intestazione TCP.	
128	3	Puntatore per l'aggiunta del pacchetto non valido.	
128	7	Puntatore socket non valido.	
128	10	Opzione tipo di servizio, frammento o durata non valida.	
128	17	Chiamante non valido per questo servizio.	
128	18	Pacchetto non valido.	
128	20	Questo componente non è stato attivato.	
128	21	Questo componente è già stato attivato.	
128	22	Ascolto non abilitato per la porta specificata.	
128	26	La sospensione richiesta è stata annullata.	
128	33	Indirizzo IP del server non valido.	
128	34	Questo socket è associato a un'altra porta.	
128	35	La porta è associata a un socket diverso.	
128	36	Il socket non è associato.	
128	38	Il socket non è stato associato durante l'attesa sospesa di un pacchetto di ricezione.	
128	39	Il socket non è stato creato.	
128	51	Nessuna ulteriore struttura di richiesta di ascolto disponibile.	
128	52	Esiste già una richiesta di attesa attiva per questa porta.	
128	53	Il socket non ha uno stato chiuso.	
128	54	Il socket del server fornito non è in stato di ascolto.	
128	55	Nessun tempo di attesa specificato, il tentativo di connessione è in corso.	
128	56	Connessione non riuscita.	
128	57	La richiesta supera la dimensione in byte della finestra annunciata del destinatario.	
128	64	Un altro thread è stato sospeso. È consentito un solo thread.	
128	65	La disconnessione non è stata completata nell'intervallo di timeout.	
128	66	Il socket è associato.	
128	69	Nessuna porta disponibile	
128	70	Porta non valida.	
128	71	Esiste già un puntatore del socket valido per questa porta o per la porta specificata non è presente una richiesta di ascolto attiva.	

Codice ErrorID	Codice SubErrorID	Descrizione errore	Azione correttiva
128	72	Come per NX_SUCCESS , con l'unica differenza che durante la chiamata è stata elaborata una richiesta di connessione in coda.	
128	73	Errore per superamento della coda di trasmissione del TCP.	

Vedere anche

[Istruzioni socket](#) a pagina 603

[Tipo di dati SOCK_STATUS](#) a pagina 637

[Bit di stato per l'istruzione del socket](#) a pagina 641

Bit di stato per l'istruzione del socket

La tabella seguente descrive l'esecuzione dei Bit dello stato per le istruzioni del socket.

Numero di bit	Nome	Descrizione
0	EN - Abilita bit	Il bit EN viene impostato quando l'istruzione è abilitata a causa di una transizione da False a True, ma non è ancora stata completata o ha generato errori.
1	EW - Abilita bit nella coda Attesa	Il bit EW viene impostato quando il controllore rileva che una richiesta di istruzioni del socket è stata inserita in coda. Il controllore esegue il reset del bit EW quando viene impostato il bit ST.
2	ST - Avvia bit	Il bit ST viene impostato durante l'esecuzione dell'istruzione in coda. Quando viene impostato il bit DN o ER, viene eseguito il reset del bit ST.
3	ER - Errore bit	Indica che si è verificato un errore durante l'esecuzione dell'istruzione. Viene eseguito il reset del bit ER al successivo passaggio in cui si verifica la condizione di transizione da False a True.
4	DN - Bit completato	Il bit DN viene impostato quando l'istruzione del socket viene completata correttamente. Viene eseguito il reset del bit DN al successivo passaggio in cui si verifica la condizione di transizione da False a True.

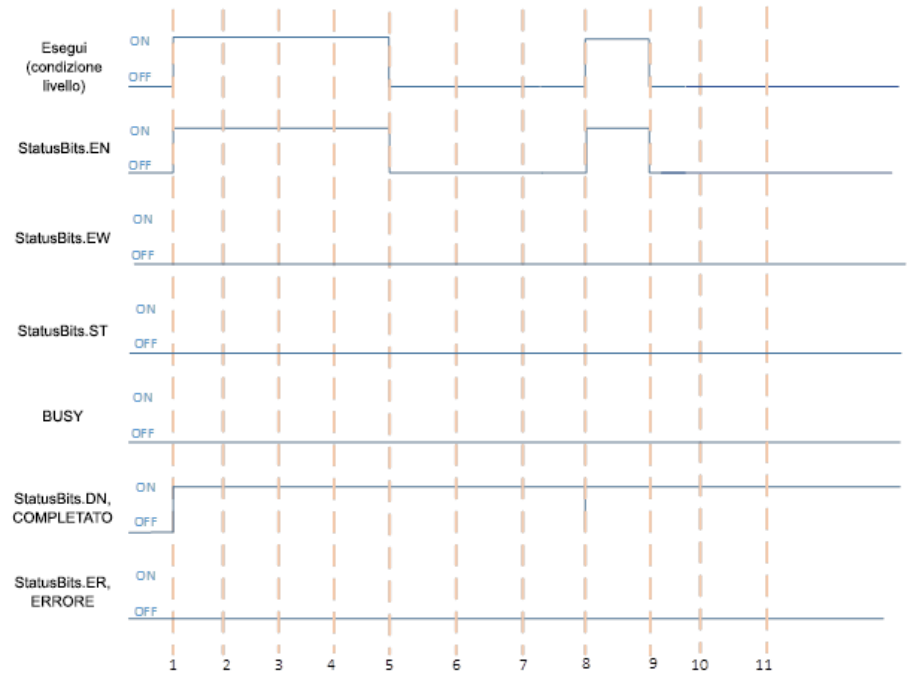
Vedere anche

[Codici di errore del socket](#) a pagina 637

[Tipo di dati SOCKADDR_CFG](#) a pagina 636

Diagrammi di temporizzazione dell'istruzione Socket

Esecuzione corretta delle istruzioni del socket quando il processo è immediato



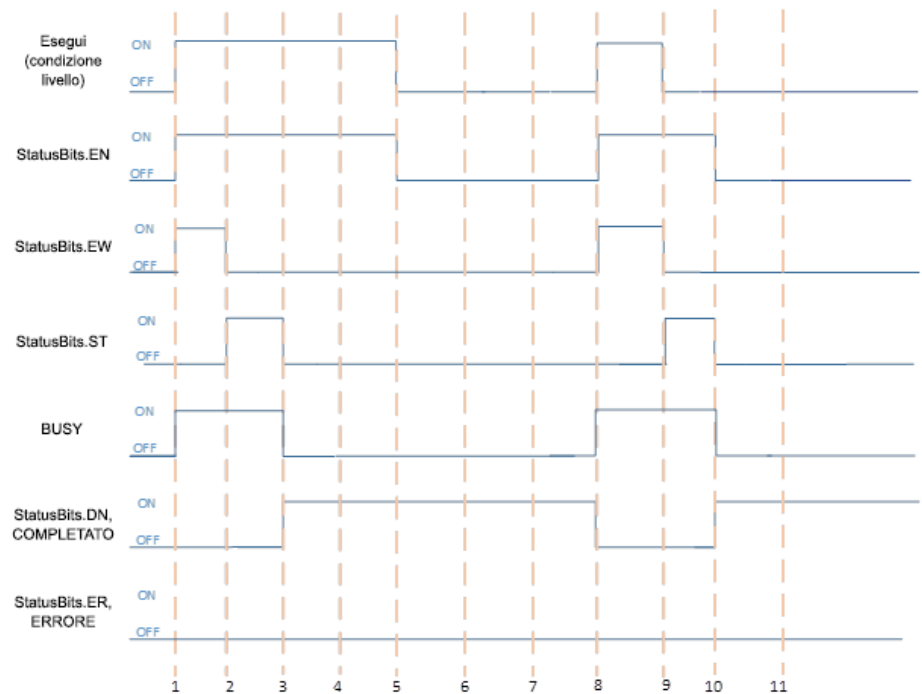
Condizione A: La condizione ramo è TRUE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
1	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • Le istruzioni del socket completano l'esecuzione correttamente. • L'uscita per il bit DN e DONE viene impostata.
2, 3, 4	Nessuna modifica nella condizione ramo.
5	La condizione ramo diventa FALSE quando il bit EN viene cancellato.
6,7	Nessuna modifica nella condizione ramo.

Condizione B: La condizione ramo è FALSE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Le istruzioni del socket completano l'esecuzione correttamente. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • L'uscita per il bit DN e DONE viene impostata.
9	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit EN è cancellato.
10,11	Nessuna modifica nella condizione ramo.

Esecuzione corretta delle istruzioni del socket quando il processo non è immediato



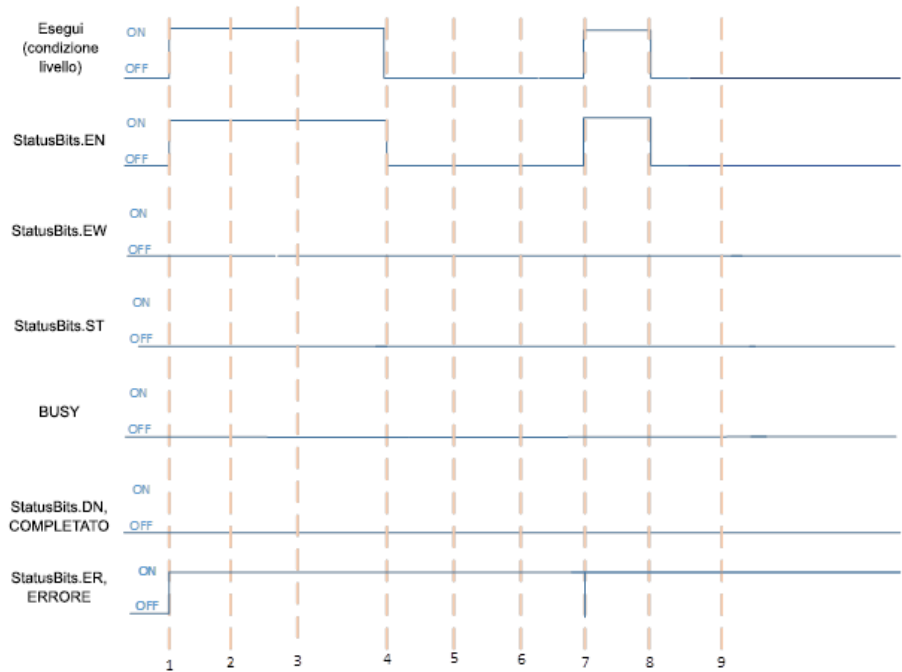
Condizione A: La condizione ramo è TRUE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
1	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • Le istruzioni del socket eseguono l'elaborazione in background. • L'uscita per il bit EW e BUSY viene impostata. • Parametro ingresso delle istruzioni del socket bloccato per l'elaborazione in background.
2	Le istruzioni del socket avviano l'esecuzione quando: <ul style="list-style-type: none"> • Il bit EW è cancellato. • Il bit ST viene impostato.
3	Le istruzioni del socket completano l'esecuzione correttamente quando: <ul style="list-style-type: none"> • L'uscita per il bit ST e BUSY viene cancellata. • L'uscita per il bit DN e DONE viene impostata.
4	Nessuna modifica nella condizione ramo.
5	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit EN è cancellato.
6,7	Nessuna modifica nella condizione ramo.

Condizione B: La condizione ramo diventa FALSE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • Le istruzioni del socket inviano l'elaborazione in background. • L'uscita per il bit EW e BUSY viene impostata. • Parametro ingresso delle istruzioni del socket bloccato per l'elaborazione in background.
9	Le istruzioni del socket avviano l'esecuzione quando: <ul style="list-style-type: none"> • Il bit EW è cancellato. • Il bit ST viene impostato. • La condizione ramo diventa FALSE.
10	Le istruzioni del socket completano l'esecuzione correttamente quando: <ul style="list-style-type: none"> • L'uscita per il bit ST e BUSY viene cancellata. • L'uscita per il bit DN e DONE viene impostata. • La condizione ramo è FALSE. • Il bit EN è cancellato.

Errore delle istruzioni del socket quando EN è TRUE ed EW e ST sono FALSE



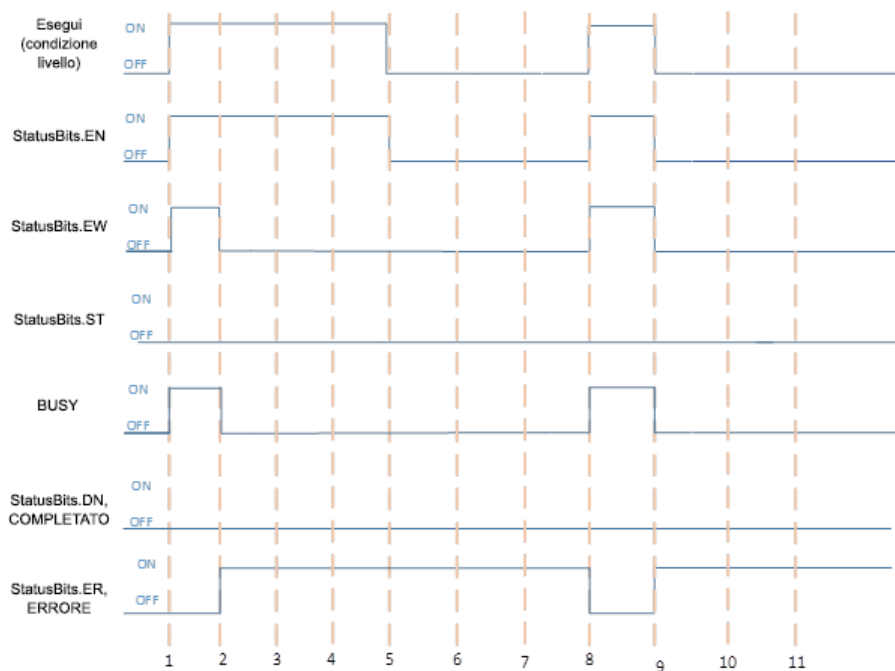
Condizione A: La condizione ramo è TRUE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
1	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • Le istruzioni del socket consentono di completare l'esecuzione con errore. • L'uscita per il bit ER e l'errore viene impostata.
2,3	Nessuna modifica nella condizione ramo.
4	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit EN è cancellato.
5,6	Nessuna modifica nella condizione ramo.

Condizione B: La condizione ramo diventa FALSE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
7	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • Le istruzioni del socket consentono di completare l'esecuzione con errore. • L'uscita per il bit ER e l'errore viene impostata.
8	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit EN è cancellato.
9	Nessuna modifica nella condizione ramo.

Errore delle istruzioni del socket quando EW è TRUE e il processo dell'istruzione non è immediato



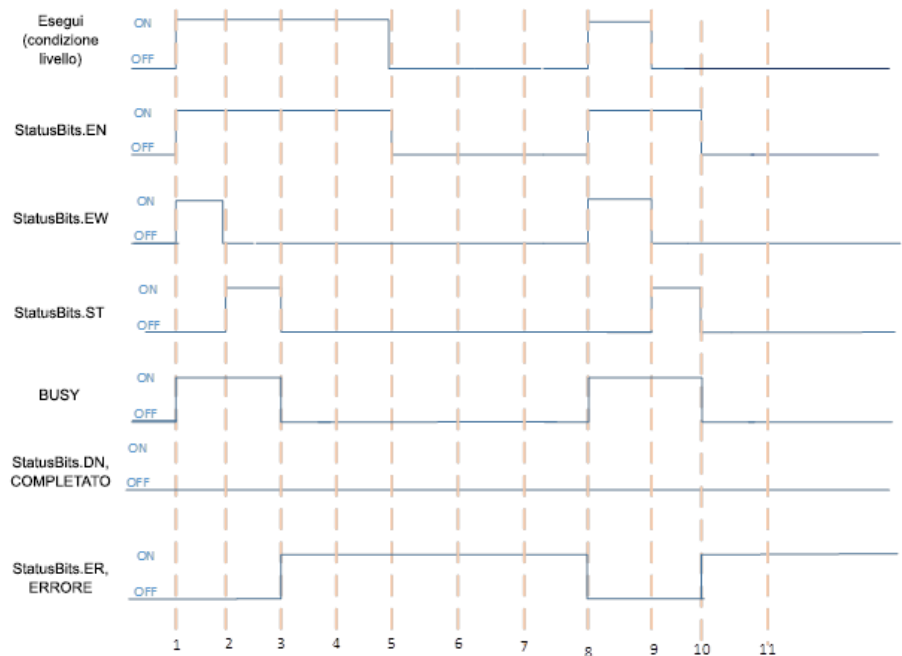
Condizione A: La condizione ramo è TRUE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
1	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • Le istruzioni del socket inviano l'elaborazione in background. • L'uscita per il bit EW e BUSY viene impostata. • Parametro ingresso delle istruzioni del socket bloccato per l'elaborazione in background.
2	Le istruzioni del socket completano l'esecuzione con errore quando: <ul style="list-style-type: none"> • L'uscita per il bit EW e BUSY viene cancellata. • L'uscita per il bit ER e l'errore viene impostata.
3,4	Nessuna modifica nella condizione ramo.
5	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit EN è cancellato.
6,7	Nessuna modifica nella condizione ramo.

Condizione B: La condizione ramo diventa FALSE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • Le istruzioni del socket inviano l'elaborazione in background. • L'uscita per il bit EW e BUSY viene impostata. • Parametro ingresso delle istruzioni del socket bloccato per l'elaborazione in background.
9	Le istruzioni del socket completano l'esecuzione con errore quando: <ul style="list-style-type: none"> • L'uscita per il bit EW e BUSY viene cancellata. • L'uscita per il bit ER e l'errore viene impostata. • La condizione ramo diventa FALSE. • Il bit EN è cancellato.
10,11	Nessuna modifica nella condizione ramo.

Errore delle istruzioni del socket quando ST è TRUE e il processo dell'istruzione non è immediato



Condizione A: La condizione ramo è TRUE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
1	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • Le istruzioni del socket inviano l'elaborazione in background. • L'uscita per il bit EW e BUSY viene impostata. • Parametro ingresso delle istruzioni del socket bloccato per l'elaborazione in background.
2	Le istruzioni del socket avviano l'esecuzione quando: <ul style="list-style-type: none"> • Il bit EW è cancellato. • Il bit ST viene impostato.
3	Le istruzioni del socket completano l'esecuzione con errore quando: <ul style="list-style-type: none"> • L'uscita per il bit ST e l'uscita BUSY viene cancellata. • L'uscita per il bit ER e l'errore viene impostata.
4	Nessuna modifica nella condizione ramo.
5	La condizione ramo diventa FALSE quando: <ul style="list-style-type: none"> • Il bit EN è cancellato.
6,7	Nessuna modifica nella condizione ramo.

Condizione B: La condizione ramo diventa FALSE durante l'esecuzione delle istruzioni

Condizione ramo	Descrizione
8	La condizione ramo diventa TRUE quando: <ul style="list-style-type: none"> • L'esecuzione delle istruzioni del socket è abilitata. • Viene impostato il bit EN e tutti gli altri bit vengono cancellati. • Le istruzioni del socket inviano l'elaborazione in background. • L'uscita per il bit EW e BUSY viene impostata. • Parametro ingresso delle istruzioni del socket bloccato per l'elaborazione in background.
9	Le istruzioni del socket avviano l'esecuzione quando: <ul style="list-style-type: none"> • Il bit EW viene cancellato e il bit ST impostato.
10	Le istruzioni del socket completano l'esecuzione con errore quando: <ul style="list-style-type: none"> • L'uscita per il bit ST e BUSY viene cancellata. • L'uscita per il bit ER e l'errore viene impostata. • La condizione ramo diventa FALSE. • Il bit EN è cancellato.
11	Nessuna modifica nella condizione ramo.

Vedere anche

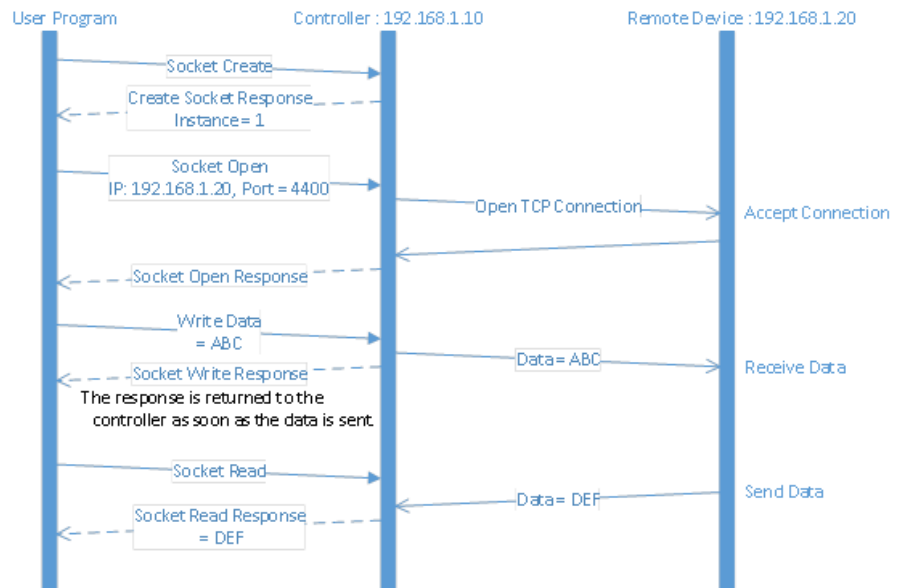
[Istruzioni socket](#) a pagina 603

[Bit di stato per l'istruzione del socket](#) a pagina 641

Diagrammi di transazione dell'istruzione Socket

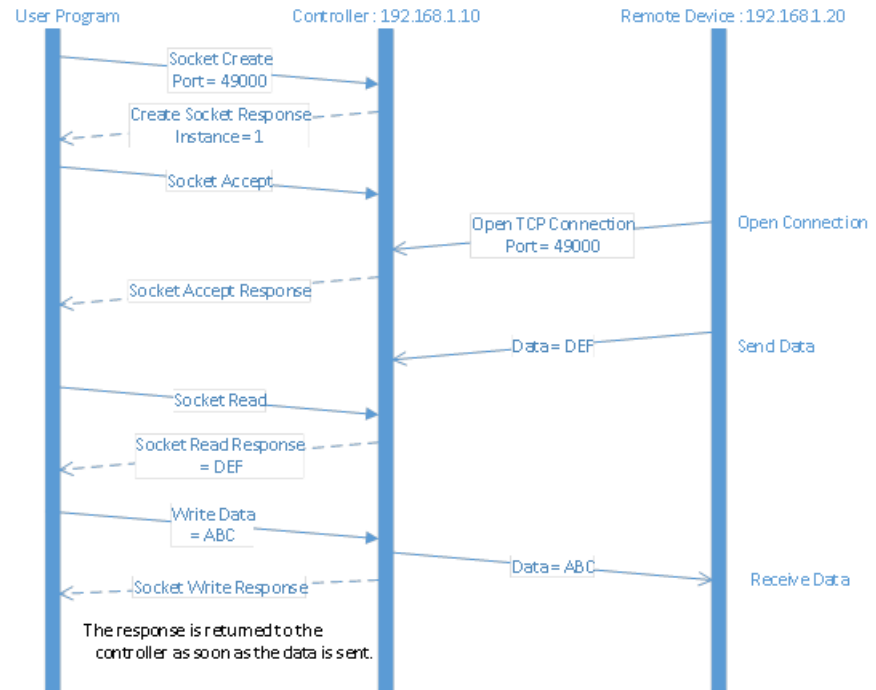
Transazioni per client TCP

Il diagramma seguente mostra una sequenza tipica di transazioni dell'interfaccia del socket con il controllore che opera come client TCP.



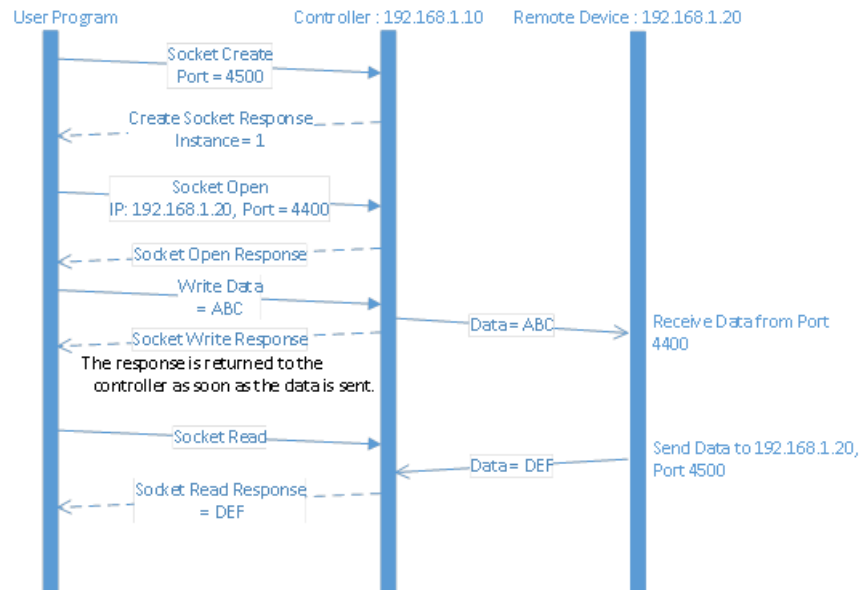
Transazioni per server TCP

Il diagramma seguente mostra una sequenza tipica di transazioni dell'interfaccia del socket con il controllore come un server TCP.



Transazioni per UDP con connessione aperta

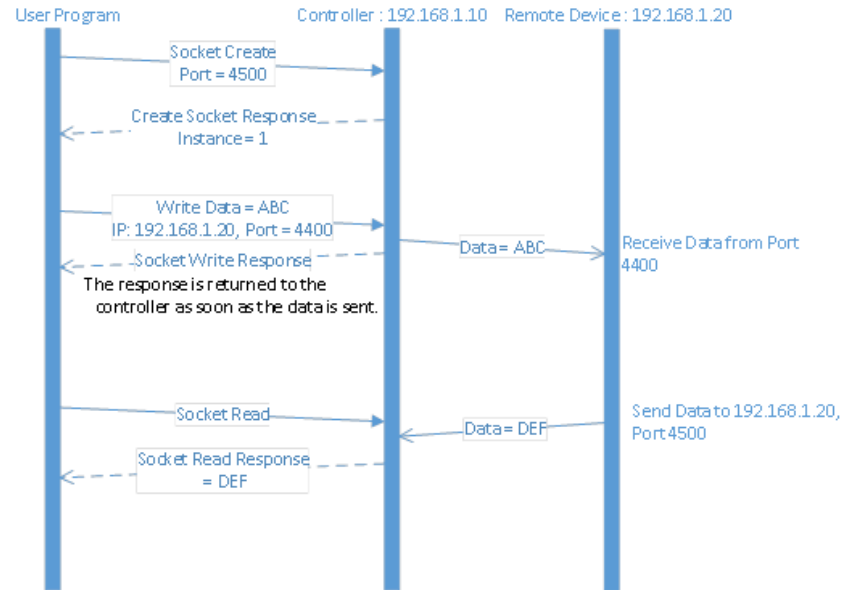
Il diagramma seguente mostra una sequenza tipica di transazioni dell'interfaccia del socket per le comunicazioni UDP quando si utilizza il servizio di connessione aperta per specificare l'indirizzo di destinazione.



Transazioni per UDP senza connessioni aperte

Il diagramma seguente mostra una sequenza tipica di transazioni dell'interfaccia del socket per le comunicazioni UDP senza utilizzare il servizio di connessione aperta per specificare l'indirizzo di destinazione. In questo caso, il controllore specifica la destinazione per

ogni datagramma e riceve l'indirizzo del mittente insieme a ogni datagramma ricevuto.



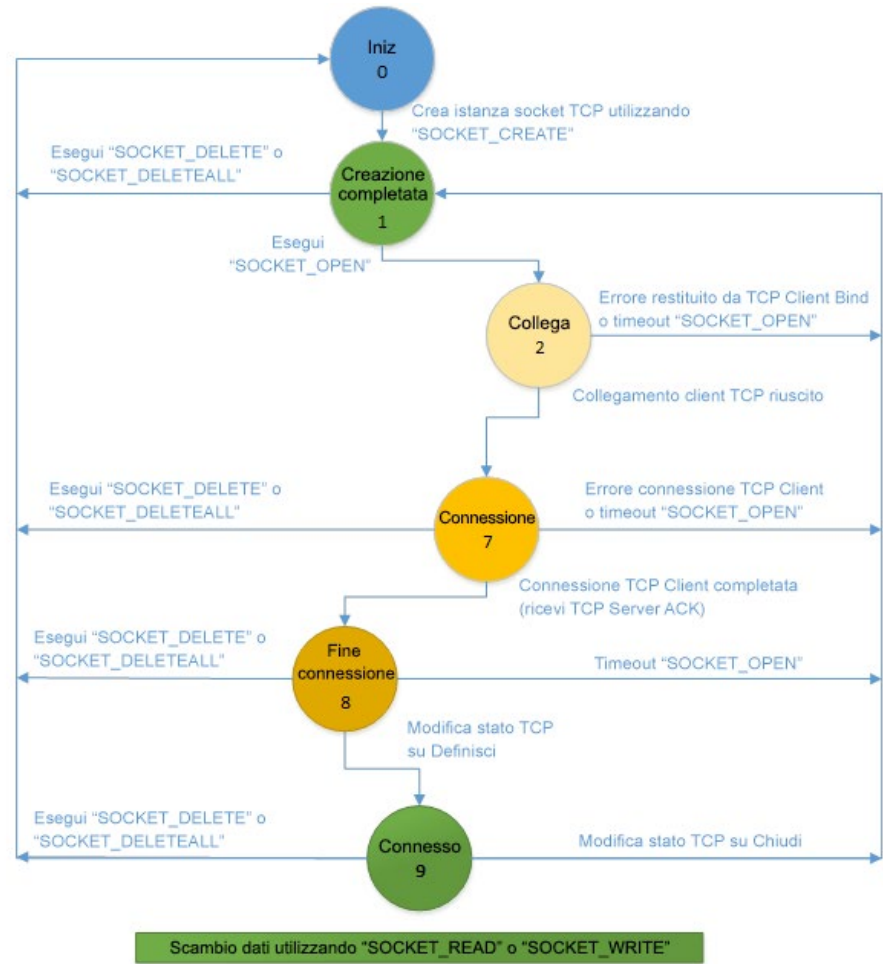
Vedere anche

[Istruzioni socket](#) a pagina 603

[Diagrammi di temporizzazione dell'istruzione Socket](#) a pagina 641

Diagrammi di stato della macchina per TCP

Client TCP tramite le istruzioni del socket



Server TCP tramite le istruzioni del socket

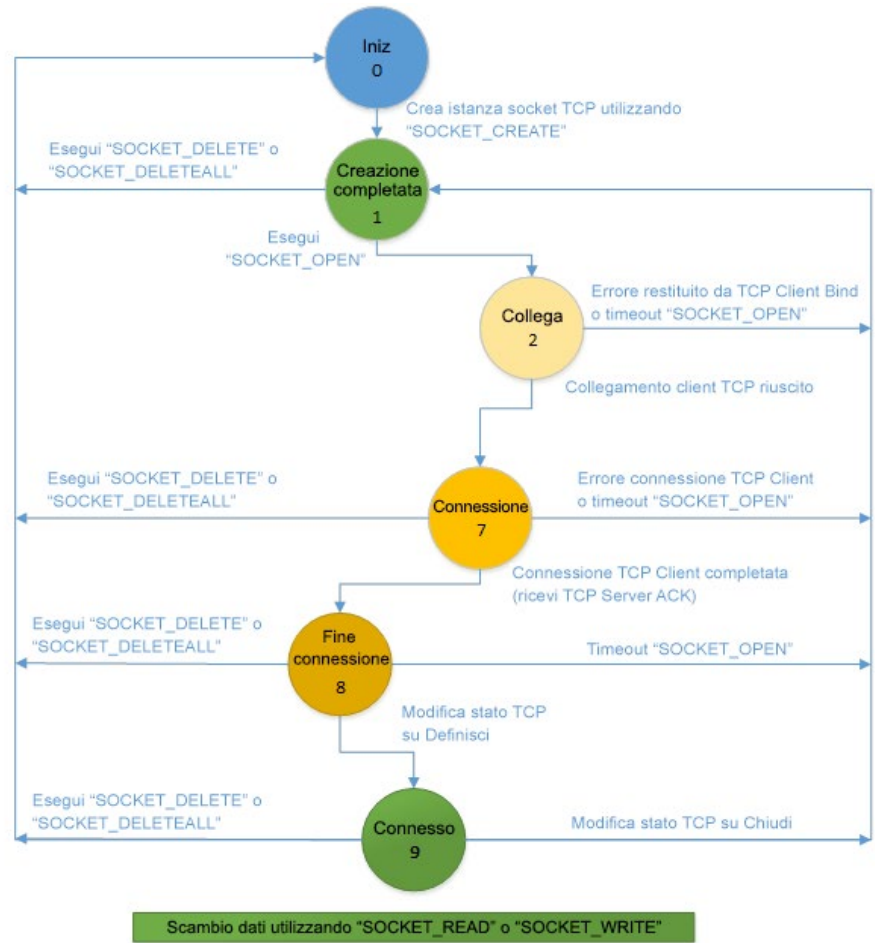
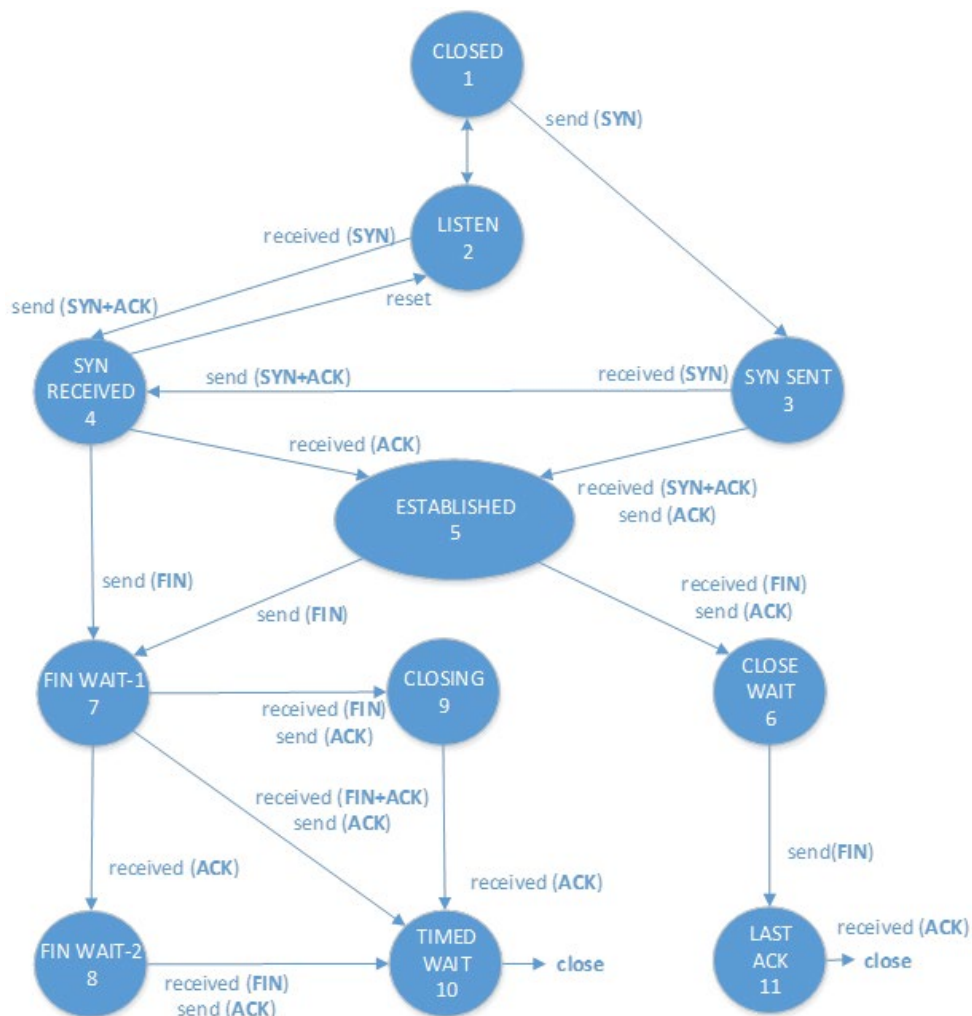


Diagramma dello stato del TCP



Vedere anche

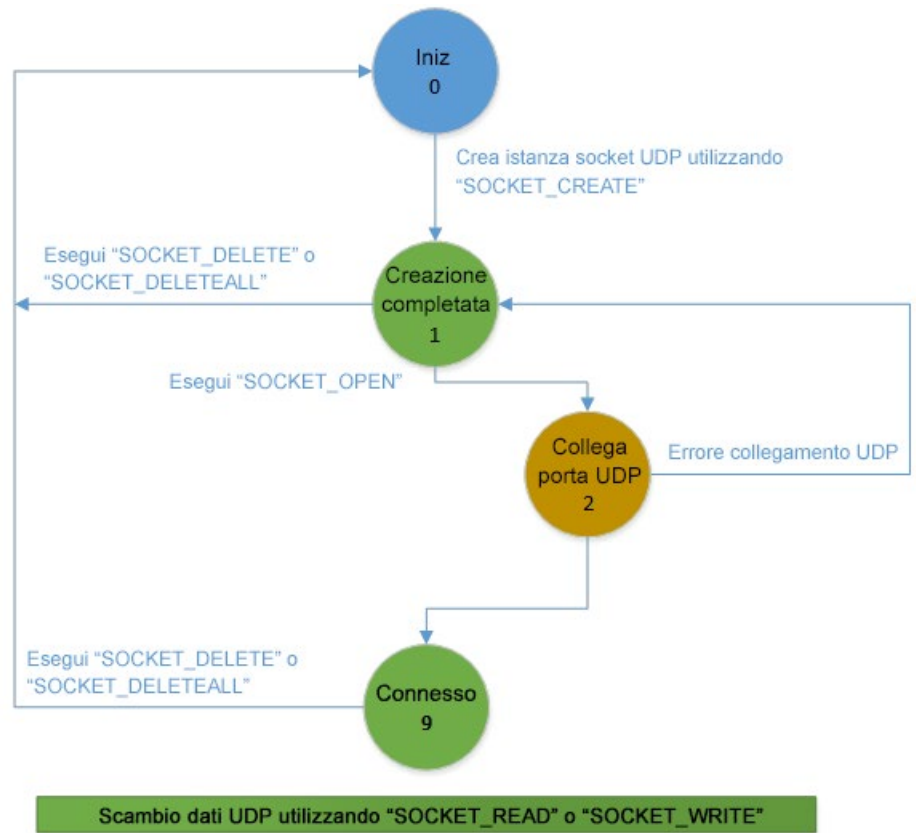
[Istruzioni socket](#) a pagina 603

[Diagrammi di temporizzazione dell'istruzione Socket](#) a pagina 641

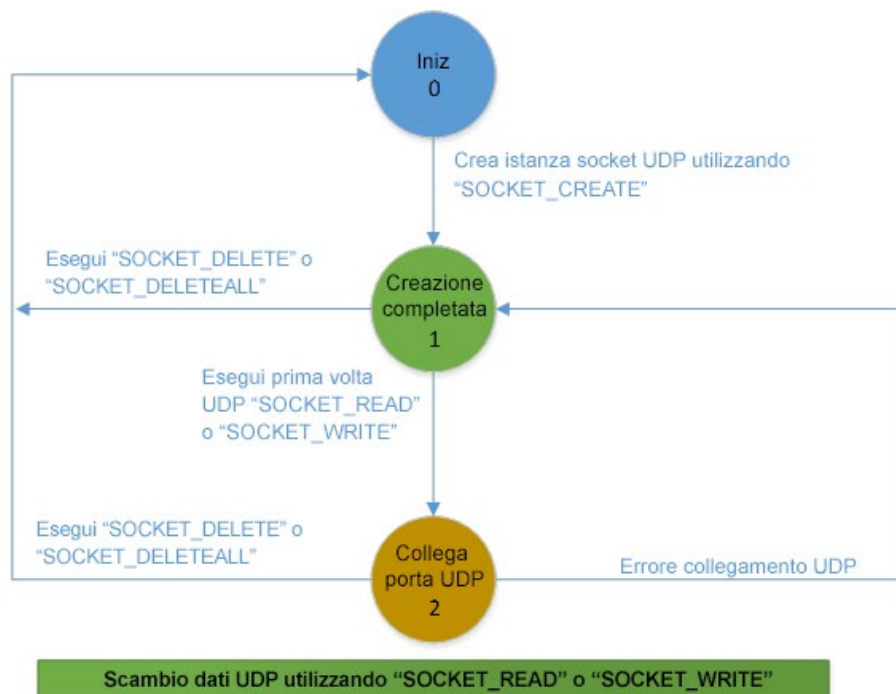
[Diagrammi di stato della macchina per UDP](#) a pagina 654

Diagrammi dello stato della macchina per UDP

Datagramma UDP con istruzione SOCKET_OPEN



Datagramma UDP senza istruzione SOCKET_OPEN



Vedere anche

[Istruzioni socket](#) a pagina 603

[Diagrammi di temporizzazione dell'istruzione Socket](#) a pagina 641

[Diagrammi di stato della macchina per TCP](#) a pagina 651

Istruzioni di elaborazione della stringa

Le istruzioni di elaborazione della stringa si utilizzano per alterare una sequenza di simboli scelti da una serie o dall'alfabeto al fine di modificare lo stato delle uscite. per leggere correttamente le stringhe di ingresso contenenti caratteri speciali, inserire i caratteri delle stringhe quando il programma che contiene l'istanza del blocco funzione è online.

Istruzione	Descrizione
ASCII a pagina 657	Restituire il codice ASCII per i caratteri in una stringa. Carattere -> codice ASCII.
CHAR a pagina 659	Restituisce una stringa di un carattere per un codice ASCII. Codice ASCII -> Carattere.
DELETE a pagina 661	Elimina i caratteri da una stringa.
FIND a pagina 663	individua e fornisce la posizione delle sottostringhe nelle stringhe.
INSERT a pagina 664	inserisce sottostringhe in posizioni definite dall'utente all'interno delle stringhe.
LEFT a pagina 667	Estrae i caratteri dal lato sinistro di una stringa.
MID a pagina 668	Estrae i caratteri dal centro di una stringa.
MLEN a pagina 671	Calcola la lunghezza di una stringa.
REPLACE a pagina 674	Sostituisce le parti di una stringa con nuovi set di caratteri.
RIGHT a pagina 672	Estrae i caratteri dal lato destro di una stringa.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

ASCII

Restituisce il codice ASCII per i caratteri nelle stringhe.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



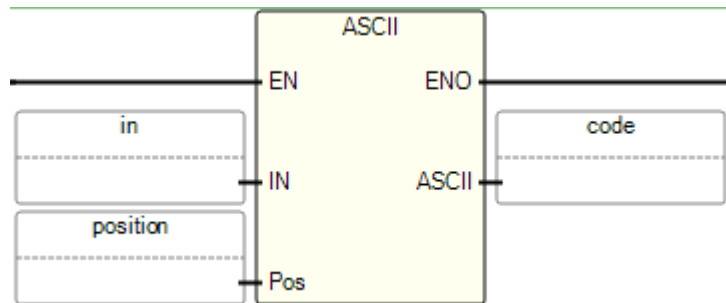
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: visualizza il codice ASCII per i caratteri. FALSE: nessuna operazione di visualizzazione. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	STRING	Qualsiasi stringa non vuota.
Pos	Ingresso	DINT	Posizione del carattere selezionato nel set [1.. len] (len rappresenta la lunghezza della stringa IN)
ASCII	Uscita	DINT	Codice ASCII del carattere selezionato (nel set [0 .. 255]) produce 0 se Pos non rientra nella stringa.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali ASCII



Esempio di diagramma ladder ASCII



Esempio di testo strutturato ASCII

```

1 | position := 1;
2 | code := ASCII(in, position);

```

ASCII (

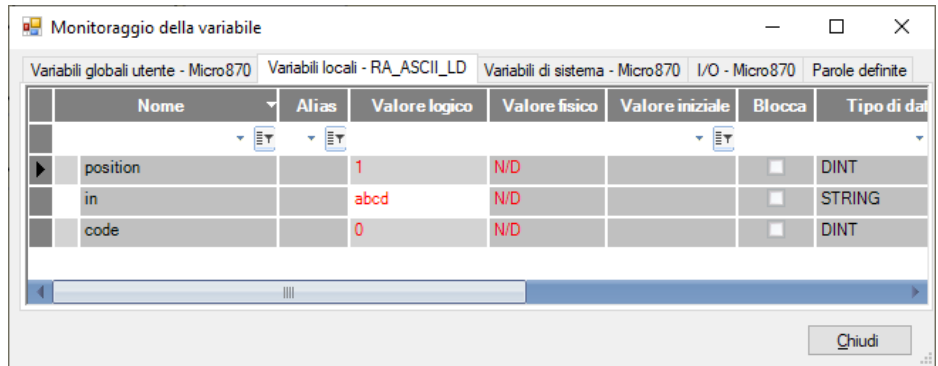
DINT ASCII (STRING IN, DINT Pos)
Carattere -> Codice ASCII

(* Equivalenza ST: *)

```
FirstChr := ASCII (message, 1);
```

(* FirstChr è il codice ASCII del primo carattere della stringa *)

Risultati



Vedere anche

[Istruzioni di elaborazione della stringa](#) a pagina 657

CHAR (codice ASCII a carattere della stringa)

Restituisce una stringa di un carattere per un codice ASCII. Codice ASCII -> Carattere.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



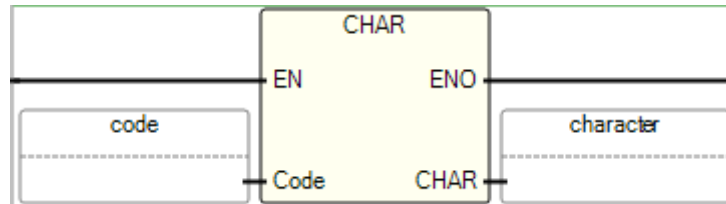
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: fornisce una stringa con un solo carattere. FALSE: nessuna operazione. Applicabile alle programmazioni in Ladder Diagram.
Codice	Ingresso	DINT	Codice ASCII nel set [0 .. 255].
CHAR	Uscita	STRING	Stringa con un carattere. Il carattere è dotato del codice ASCII dato nel codice di ingresso.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali CHAR



Esempio di diagramma ladder CHAR



Esempio di testo strutturato CHAR

```

1 | code := 97;
2 | character := CHAR(code);

```

CHAR (

STRING CHAR(DINT Code)
Codice ASCII -> Carattere

(* Equivalenza ST: *)

Display := CHAR (value + 48);

(* il valore è nel set [0..9] *)

(* 48 è il codice ascii di '0' *)

(* il risultato è una stringa con un carattere da '0' a '9' *)

Risultati

Monitoraggio della variabile						
Variabili globali utente - Micro870		Variabili locali - RA_CHAR_LD		Variabili di sistema - Micro870		I/O - Micro870
Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	
code	97	N/D		<input type="checkbox"/>	DINT	
character	a	N/D		<input type="checkbox"/>	STRING	

Vedere anche

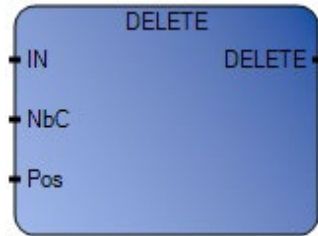
[Istruzioni di elaborazione della stringa](#) a pagina 657

DELETE (eliminare una sottostringa)

Elimina i caratteri da una stringa.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

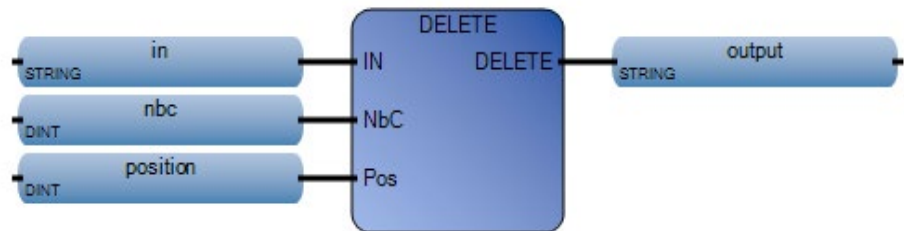
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



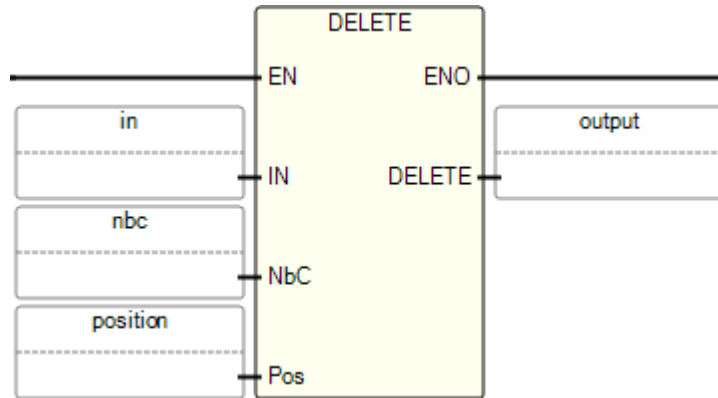
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: eliminare parti specifiche di una stringa. FALSE: nessuna operazione. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	STRING	Qualsiasi stringa non vuota.
NbC	Ingresso	DINT	Numero di caratteri da eliminare.
Pos	Ingresso	DINT	Posizione del primo carattere eliminato (il primo carattere della stringa è in posizione 1).
DELETE	Uscita	STRING	Il risultato è: <ul style="list-style-type: none"> • Una stringa modificata. • Una stringa vuota (se Pos < 1). • La stringa iniziale (se Pos > lunghezza stringa IN). • La stringa iniziale (se NbC <= 0).
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali DELETE



Esempio di diagramma ladder DELETE



Esempio di testo strutturato DELETE

```
DELETE {
  STRING DELETE (STRING IN, DINT NbC, DINT Pos)
  Elimina stringa secondaria
```

```
1 nbc := 3;
2 position := 2;
3 output := DELETE (in, nbc, position);
```

(* Equivalenza ST: *)

stringa_completa := INSERT ('ABCD ', 'EFGH', 5); (* stringa_completa è 'ABCDEFGH' *)

sotto_stringa := DELETE (stringa_completa, 4, 3); (* sotto_stringa è 'ABGH' *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
in	abcdefg	N/D		<input type="checkbox"/>	STRING
nbc	3	N/D		<input type="checkbox"/>	DINT
output	aefg	N/D		<input checked="" type="checkbox"/>	STRING
position	2	N/D		<input type="checkbox"/>	DINT

Vedere anche

[Istruzioni di elaborazione della stringa](#) a pagina 657

FIND (trova sottostringa)

individua e fornisce la posizione delle sottostringhe nelle stringhe.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

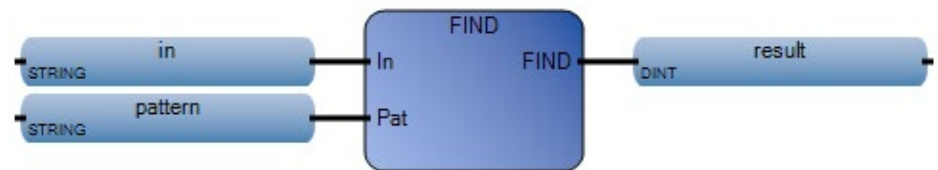
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



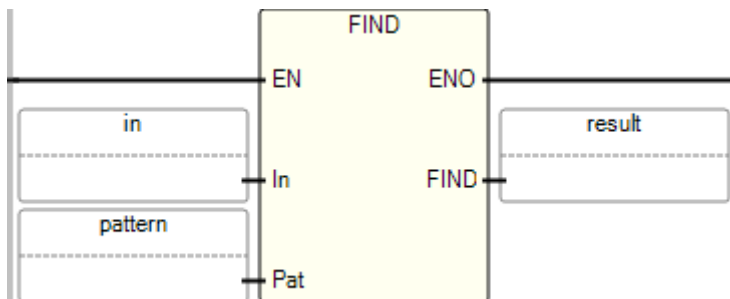
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: individua la posizione all'interno delle stringhe. FALSE: nessuna operazione di individuazione. Applicabile alle programmazioni in Ladder Diagram.
In	Ingresso	STRING	Qualsiasi stringa non vuota.
Pat	Ingresso	STRING	Qualunque stringa non vuota (pattern).
FIND	Uscita	DINT	Il risultato è: <ul style="list-style-type: none"> • 0 se la sottostringa Pat non è stata trovata. • la posizione del primo carattere della prima occorrenza della sottostringa Pat (la prima posizione è 1) Questa istruzione fa distinzione tra maiuscole e minuscole.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

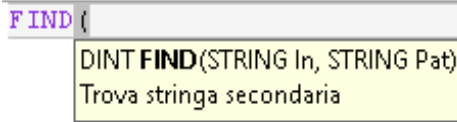
Esempio di programmazione a blocchi funzionali FIND



Esempio di diagramma ladder FIND



Esempio di testo strutturato FIND



```
1 result := FIND(in, pattern);
```

(* Equivalenza ST: *)

stringa_completa := 'ABCD' + 'EFGH'; (* stringa_completa è 'ABCDEFGH' *)

trovato := FIND (stringa_completa, 'CDEF'); (* trovato è 3 *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
in	abcd	N/D		<input type="checkbox"/>	STRING
pattern	bc	N/D		<input type="checkbox"/>	STRING
result	2	N/D		<input type="checkbox"/>	DINT

Vedere anche

[Istruzioni di elaborazione della stringa](#) a pagina 657

INSERT (inserimento stringa)

inserisce sottostringhe in posizioni definite dall'utente all'interno delle stringhe.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

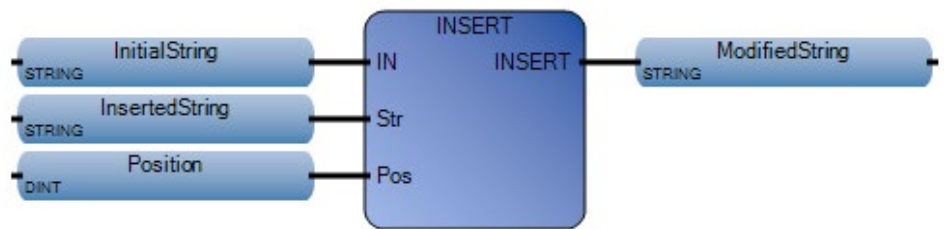
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



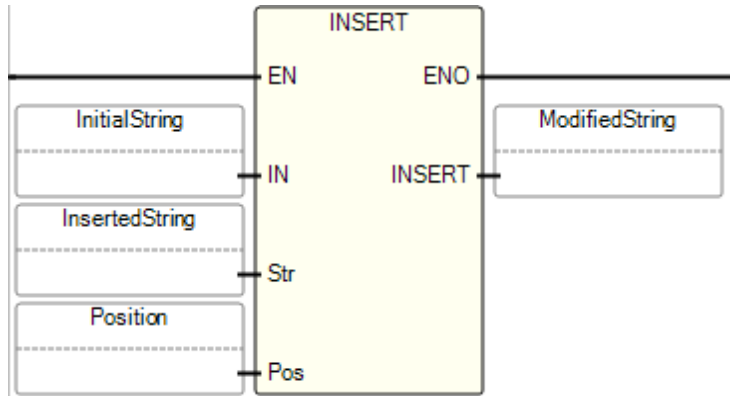
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: inserisce sottostringhe in una stringa. FALSE: nessuna operazione. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	STRING	Stringa iniziale.
Str	Ingresso	STRING	Stringa da inserire.
Pos	Ingresso	DINT	Posizione di inserimento l'inserimento avviene prima della posizione (la prima posizione valida è 1).
INSERT	Uscita	STRING	Stringa modificata. Può essere: <ul style="list-style-type: none"> • stringa vuota se Pos <= 0 • concatenamento di entrambe le stringhe se Pos è maggiore della lunghezza della stringa IN
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali INSERT



Esempio di diagramma ladder INSERT



Esempio di testo strutturato INSERT

```
INSERT (
  STRING IN, STRING Str, DINT Pos)
  Inserisci stringa
```

```
1 Position := 3;
2 ModifiedString := INSERT(InitialString, InsertedString, Position);
```

(* Equivalenza ST: *)

MyName := INSERT ('Mr JONES', 'Frank ', 4);

(* MyName è 'Mr Frank JONES' *)

Risultati

Monitoraggio della variabile						
Variabili globali utente - Micro870						
Variabili locali - RA_INSERT_LD						
Variabili di sistema - Micro870						
I/O - Micro870						
Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	
Position	3	N/D		<input type="checkbox"/>	DINT	
ModifiedString	abefgcd	N/D		<input type="checkbox"/>	STRING	
InsertedString	efg	N/D		<input type="checkbox"/>	STRING	
InitialString	abcd	N/D		<input type="checkbox"/>	STRING	

Vedere anche

[Istruzioni di elaborazione della stringa](#) a pagina 657

LEFT (estrae a sinistra di una stringa)

Estrarre i caratteri dal lato sinistro di una stringa.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

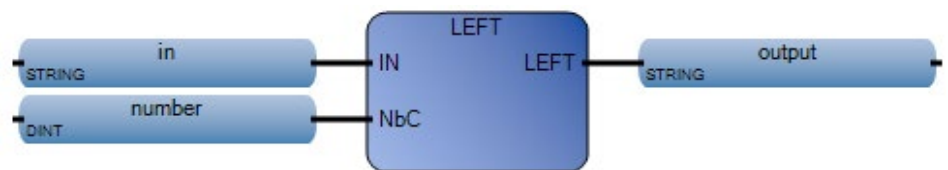
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



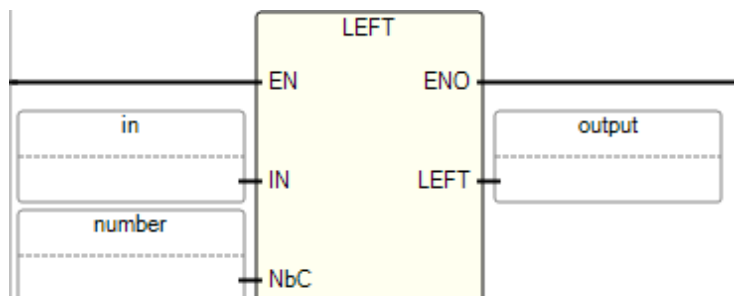
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: calcola il numero di caratteri della stringa partendo da sinistra. FALSE: nessuna operazione. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	STRING	Qualsiasi stringa non vuota.
NbC	Ingresso	DINT	Numero di caratteri da estrarre. Questo numero non può essere maggiore della lunghezza della stringa IN.
LEFT	Uscita	STRING	Parte sinistra della stringa IN (lunghezza = NbC). Può essere: <ul style="list-style-type: none"> • stringa vuota se NbC <= 0 • una stringa IN completa se NbC >= lunghezza stringa IN
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali LEFT



Esempio di diagramma ladder LEFT



Esempio di testo strutturato LEFT

```
LEFT (
  STRING LEFT (STRING IN, DINT NbC)
  Estrai parte sinistra di una stringa
```

```
1 number := 3;
2 output := LEFT(in, number);
```

(* Equivalenza ST: *)

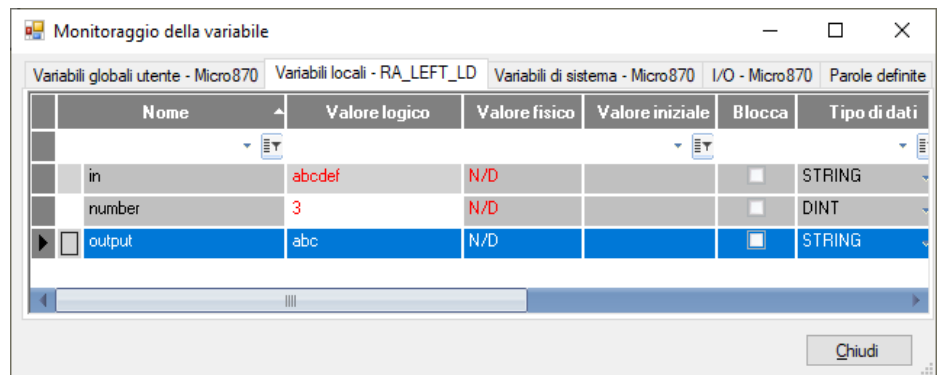
stringa_completa := RIGHT ('12345678', 4), LEFT ('12345678', 4), 5;

(* la stringa_completa è '56781234'

il valore restituito dalla chiamata a RIGHT è '5678'

il valore ottenuto dalla chiamata LEFT è '1234'*)

Risultati



Vedere anche

[Istruzioni di elaborazione della stringa](#) a pagina 657

MID (estrae dal centro di una stringa)

Estrarre i caratteri dal centro di una stringa. Utilizzare la posizione e il numero di caratteri forniti per calcolare le parti richieste delle stringhe.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



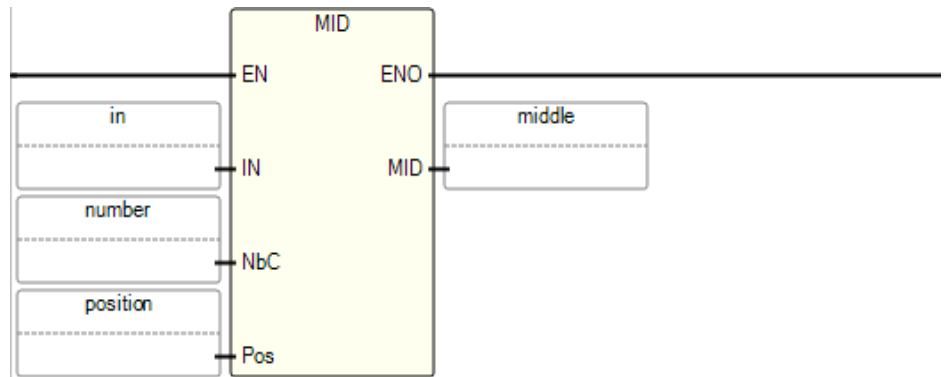
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: genera la porzione di una stringa. FALSE: nessuna operazione di generazione. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	STRING	Qualsiasi stringa non vuota.
NbC	Ingresso	DINT	Il numero di caratteri da estrarre non può essere superiore alla lunghezza della stringa IN.
Pos	Ingresso	DINT	Posizione della sottostringa. Il primo carattere della sottostringa è il primo a cui Pos fa riferimento (la prima posizione valida è 1).
MID	Uscita	STRING	Parte centrale della stringa (lunghezza = NbC). Quando il numero di caratteri da estrarre supera la lunghezza della stringa IN, NbC è ricalcolato automaticamente per ottenere solo il resto della stringa. Quando i valori di NbC o Pos sono uguali a zero o numeri negativi, viene restituita una stringa vuota.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali MID



Esempio di diagramma ladder MID



Esempio di testo strutturato MID

```
MID (
STRING MID (STRING IN, DINT NbC, DINT Pos)
Estrai parte centrale di una stringa

1 number := 3;
2 position := 2;
3 middle := MID (in, number, position);
```

(* Equivalenza ST: *)

sotto_stringa := MID ('abcdefgh', 2, 4);

(* sotto_stringa è 'de' *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati
in	ATTESA	N/D		<input type="checkbox"/>	STRING
middle	bcd	N/D		<input type="checkbox"/>	STRING
number	3	N/D		<input type="checkbox"/>	DINT
position	2	N/D		<input type="checkbox"/>	DINT

Vedere anche

[Istruzioni di elaborazione della stringa](#) a pagina 657

MLEN (lunghezza della stringa)

Calcola la lunghezza di una stringa.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

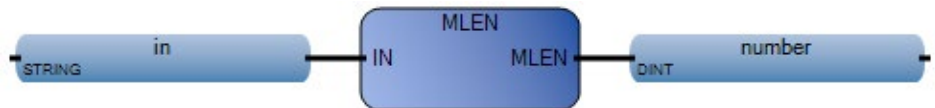
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



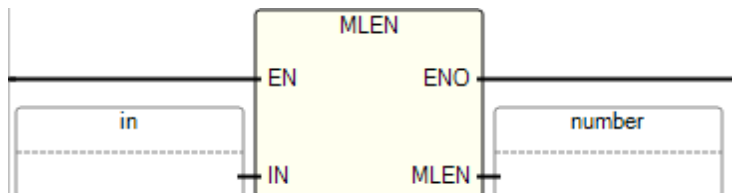
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: calcola la lunghezza della stringa. FALSE: nessuna operazione. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	STRING	Qualsiasi stringa.
MLEN	Uscita	DINT	Numero di caratteri nella stringa IN.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali MLEN



Esempio di diagramma ladder MLEN



Esempio di testo strutturato MLEN

```
MLEN (
  DINT MLEN (STRING IN)
  Recupera lunghezza stringa
1 | number := MLEN (in);
```

(* Equivalenza ST: *)

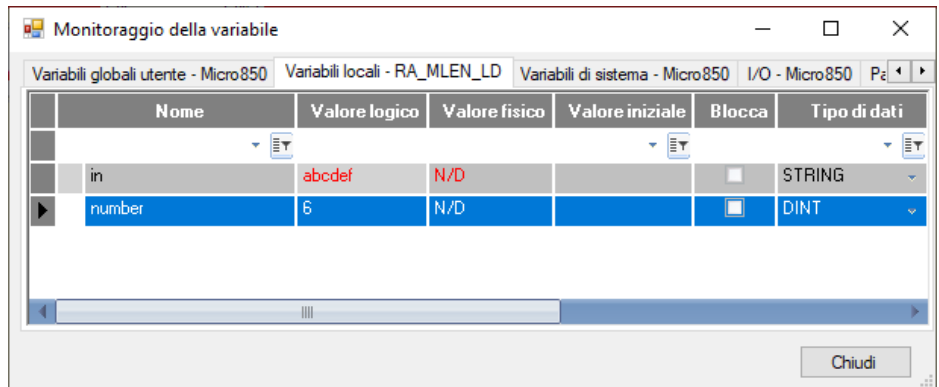
nbchar := MLEN (stringa_completa);

SE (nbchar < 3) Then Return; End_if;

prefix := LEFT (stringa_completa, 3);

(* questo programma estrae i 3 caratteri a sinistra della stringa e posiziona il risultato nella variabile della stringa di prefisso. Se la lunghezza della stringa è inferiore a 3 caratteri non viene eseguita alcuna operazione *)

Risultati



Vedere anche

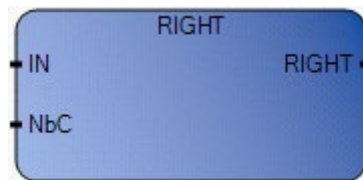
[Istruzioni di elaborazione della stringa](#) a pagina 657

RIGHT (estrazione a destra di una stringa)

Estrarre i caratteri dal lato destro di una stringa.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

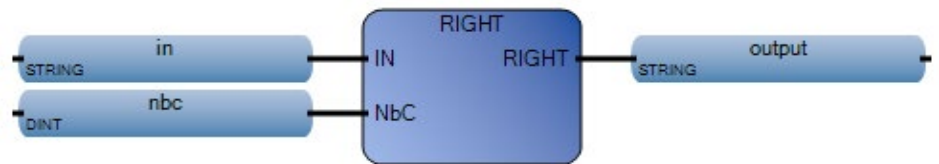
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



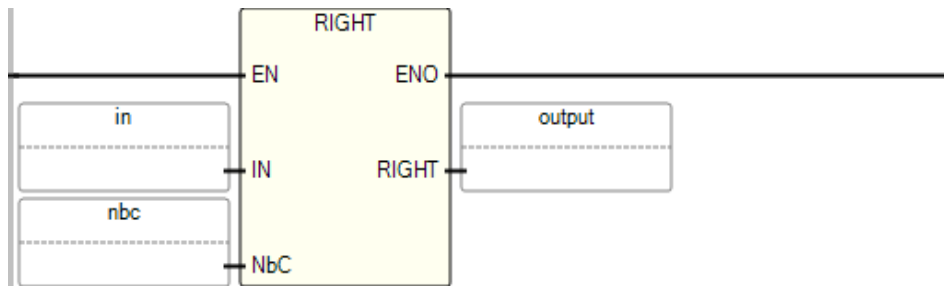
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: estrae il numero di caratteri specificato dall'estremità destra della stringa. FALSE: nessuna operazione. Applicabile alle programmazioni in Ladder Diagram.
IN	Ingresso	STRING	Qualsiasi stringa non vuota.
NbC	Ingresso	DINT	Numero di caratteri da estrarre. Questo numero non può essere maggiore della lunghezza della stringa IN.
RIGHT	Uscita	STRING	Parte destra della stringa (lunghezza = NbC). Può essere: <ul style="list-style-type: none"> • stringa vuota se NbC <= 0 • stringa completa se NbC >= lunghezza della stringa
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali RIGHT



Esempio di diagramma ladder RIGHT



Esempio di testo strutturato RIGHT

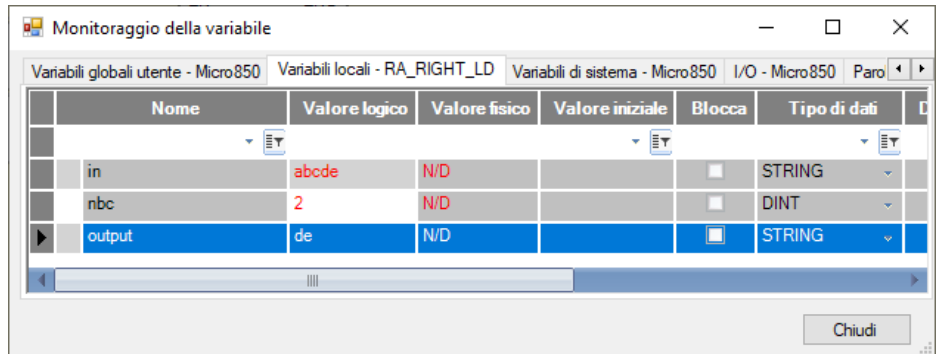
```
RIGHT (
  STRING RIGHT (STRING IN, DINT NbC)
  Estrai parte destra di una stringa
```

```
1 nbc := 2;
2 output := RIGHT(in, nbc);
```

(* Equivalenza ST: *)

stringa_completa := RIGHT ('12345678', 4), LEFT ('12345678', 4),5;
 (* la stringa_completa è '56781234'
 il valore restituito dalla chiamata a RIGHT è '5678'
 il valore restituito dalla chiamata a LEFT è '1234'
 *)

Risultati



Vedere anche

[Istruzioni di elaborazione della stringa](#) a pagina 657

REPLACE (sostituisce la sottostringa)

Sostituisce le parti di una stringa con nuovi set di caratteri.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.

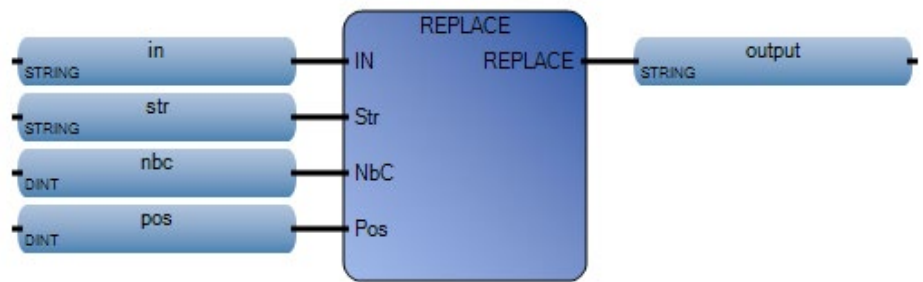


Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

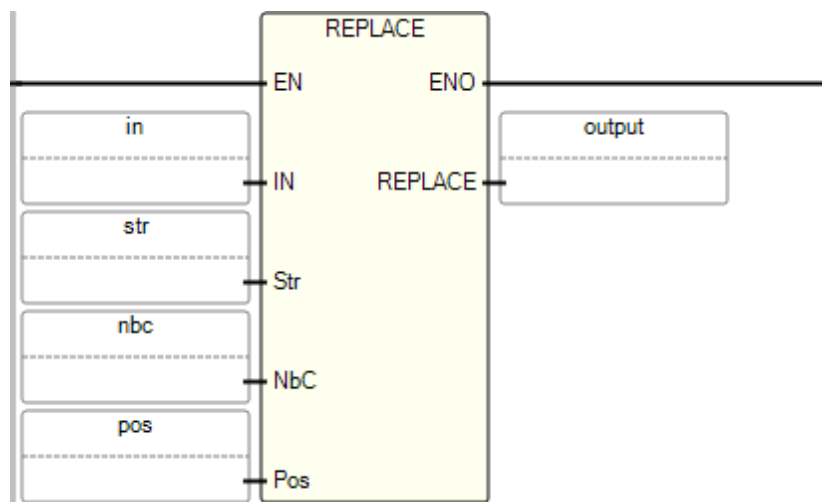
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitare funzione. TRUE: sostituisce parti di stringhe con nuovi caratteri. FALSE: nessuna operazione. Applicabile alle programmazioni in Ladder Diagram.

Parametro	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	STRING	Qualsiasi stringa.
Str	Ingresso	STRING	Stringa da inserire (per sostituire i caratteri NbC).
NbC	Ingresso	DINT	Numero di caratteri da eliminare.
Pos	Ingresso	DINT	Posizione del primo carattere modificato (la prima posizione valida è 1).
REPLACE	Uscita	STRING	Stringa modificata. I caratteri NbC sono eliminati alla posizione Pos, quando in questa posizione viene inserita la sottostringa Str. Può essere: <ul style="list-style-type: none"> • stringa vuota se Pos <= 0 • una concatenazione di stringhe (IN+Str) se Pos è maggiore della lunghezza della stringa IN • stringa iniziale IN se NbC <= 0
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.

Esempio di programmazione a blocchi funzionali REPLACE



Esempio di diagramma ladder REPLACE



Esempio di testo strutturato REPLACE

REPLACE (
 STRING **REPLACE**(STRING IN, STRING Str, DINT NbC, DINT Pos)
 Sostituisci stringa secondaria

```

1 nbc := 4;
2 pos := 2;
3 output := REPLACE(in, str, nbc, pos);
    
```

Sostituzione di una parte di una stringa con un nuovo set di caratteri.

(* Equivalenza ST: *)

MyName := REPLACE ("Mr X JONES, "Frank", 1, 4);

(* MyName è 'Mr Frank JONES' *)

Risultati

Nome	Valore logico	Valore fisico	Valore iniziale	Blocca	Tipo di dati	Dimensione
in	abcdef	N/D		<input type="checkbox"/>	STRING	
nbc	4	N/D		<input type="checkbox"/>	DINT	
output	aghif	N/D		<input checked="" type="checkbox"/>	STRING	
pos	2	N/D		<input type="checkbox"/>	DINT	
str	ghi	N/D		<input type="checkbox"/>	STRING	

Vedere anche

[Istruzioni di elaborazione della stringa](#) a pagina 657

Istruzioni del timer

Utilizzare istruzioni del Timer per controllare le operazioni in base al tempo.

Istruzione	Descrizione
TOF a pagina 678	Tempistica ritardo alla diseccitazione Incrementare un timer interno fino a un valore dato.
TON a pagina 680	Tempistica ritardo all'eccitazione Incrementare un timer interno fino a un valore dato.
TONOFF a pagina 682	Ritardo di accensione di un'uscita in caso di ramo True, quindi ritardo di spegnimento di un'uscita in caso di ramo False.
TP a pagina 685	Tempistica impulsi. Sul fronte di salita, incrementa un timer interno fino a un valore dato.
RTO a pagina 687	Tempo di ritrasmissione. Incrementa un timer interno quando l'ingresso è attivo, ma non ripristina il timer interno quando l'ingresso diventa inattivo.
DOY a pagina 689	Attiva un output se il valore del real-time clock è compreso nell'intervallo dell'impostazione Ora dell'anno.
TDF a pagina 691	Calcola la differenza di tempo tra TimeA e TimeB.
TOW a pagina 693	Attiva un'uscita se il valore del real-time clock è compreso nell'intervallo dell'impostazione Ora della settimana.

Vedere anche

[Istruzione impostate in ordine alfabetico](#) a pagina 18

Configurazione delle istruzioni del timer

La precisione temporale si riferisce al tempo trascorso tra l'abilitazione di un'istruzione del timer da parte del processore e il termine dell'intervallo temporizzato.

Il processore utilizza le seguenti informazioni provenienti dall'istruzione del timer:

- **Timer** - L'indirizzo di controllo del timer nell'area dell'archivio dati dedicata al timer.
- **Base di tempo** - Determina la modalità di funzionamento del timer.
- **Valore preimpostato** - Specifica il valore che il timer deve raggiungere prima che il processore imposti il bit Done.
- **Valore accumulato** - Il numero di incrementi temporali contati dall'istruzione. Quando abilitato, il timer aggiorna continuamente questo valore.

Vedere anche

[Istruzioni del timer](#) a pagina 677

TOF (timer, ritardo alla diseccitazione)

Incrementa un timer interno fino a un valore dato.

Dettagli operazione:

- Se si usa il parametro EN con questo blocco istruzione, il timer inizia a incrementare quando EN è impostato su TRUE e continua a incrementare anche se EN è impostato su FALSE.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

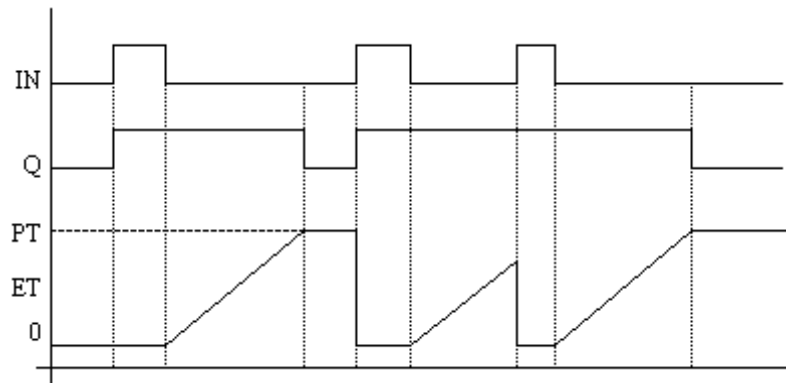
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



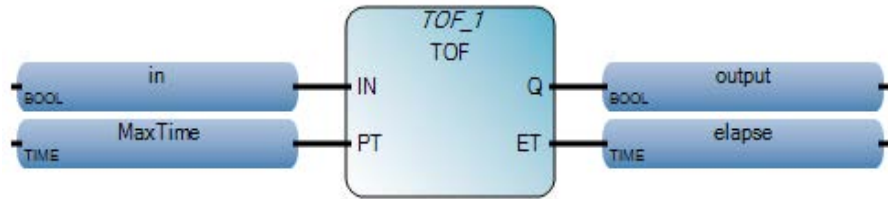
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Controllo ingresso. TRUE: fronte di discesa rilevato, il timer interno inizia ad incrementare. FALSE: fronte di salita rilevato, il timer interno si arresta e viene ripristinato.
PT	Ingresso	TIME	Tempo massimo programmato. Vedere Tipo di dati Durata.
Q	Uscita	BOOL	TRUE: il tempo totale non è trascorso. FALSE: il tempo totale è trascorso.
ET	Uscita	TIME	Tempo trascorso. I valori possibili variano da 0ms a 1193h2m47s294ms.

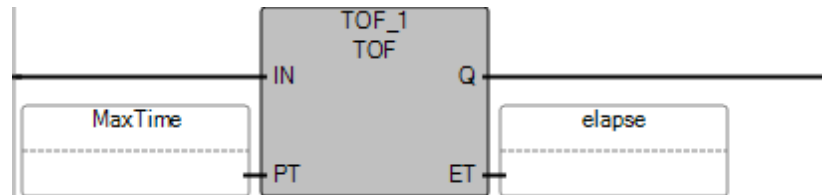
Diagramma di temporizzazione TOF



Esempio di Programmazione a blocchi funzionali TOF



Esempio di Diagramma ladder TOF



Esempio di Testo strutturato TOF

```

TOF_1 (
void TOF_1(BOOL IN, TIME PT)
Tipo: TOF, Tempo di ritardo alla disattivazione

1 MaxTime := T#3s;
2 TOF_1(in, MaxTime);
3 output := TOF_1.Q;
4 elapsed := TOF_1.ET;

```

Risultati

Monitoraggio della variabile					
Variabili globali utente - Micro870					
Variabili locali - RA_TOF_LD					
Variabili di sistema					
	Nome	Valore logico	Valore fisico	Blocca	Tipo di
+	TOF_1	<input type="checkbox"/>	TOF
	in	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
	MaxTime	T#3s	N/D	<input type="checkbox"/>	TIME
▶	elapsed	T#2s18ms	N/D	<input type="checkbox"/>	TIME
	output	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL

Vedere anche

[Configurazione delle istruzioni del timer](#) a pagina 677

TON (timer, ritardo all'eccitazione)

Incrementa un timer interno fino a un valore dato.

Dettagli operazione:

- Non usare un salto per superare un blocco istruzione TON in un Diagramma ladder (LD). Se si utilizza un jump, il timer TON continuerà dopo il tempo trascorso.
- Ad esempio: Ramo 1 contiene un salto; ramo 2 contiene un blocco istruzione TON con un tempo trascorso di 10 secondi; abilita il salto da ramo 1 a ramo 3; disabilita il salto dopo 30 secondi; il tempo trascorso sarà di 30 secondi, non 10 secondi come definito dal tempo trascorso.
- Se si usa il parametro EN con TON, il timer inizia a incrementare quando EN è impostato su TRUE e continua a incrementare anche se EN è impostato su FALSE.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

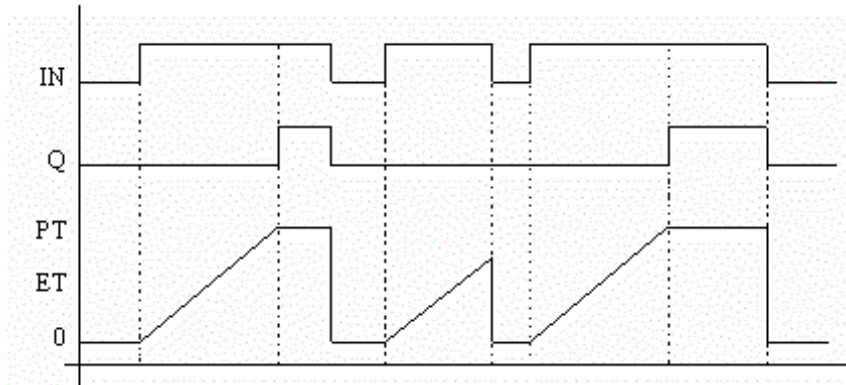
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Controllo ingresso. TRUE: se fronte di salita, il timer interno inizia a incrementare. FALSE: se fronte di discesa, il timer interno si arresta e viene ripristinato.
PT	Ingresso	TIME	Tempo massimo programmato definito utilizzando il tipo di dati Durata.
Q	Uscita	BOOL	TRUE: il tempo programmato è trascorso. FALSE: il tempo programmato non è trascorso.
ET	Uscita	TIME	Tempo trascorso. I valori possibili variano da 0ms a 1193h2m47s294ms.

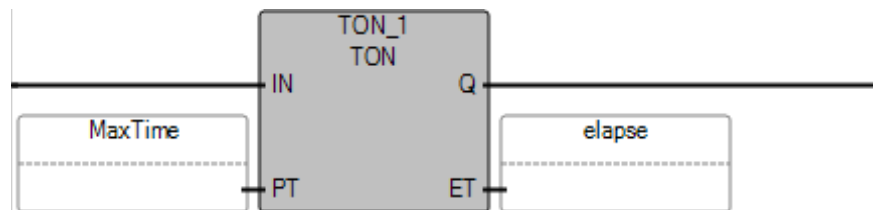
Diagramma di temporizzazione TON



Esempio di Programmazione a blocchi funzionali TON



Esempio di Diagramma ladder TON



Esempio di Testo strutturato TON

```

TON_1(
void TON_1(BOOL IN, TIME PT)
Tipo: TON, Tempo di ritardo all'eccitazione

1 MaxTime := T#3s;
2 TON_1(in, MaxTime);
3 output := TON_1.Q;
4 elapse := TON_1.ET;

```

Risultati

The screenshot shows a window titled 'Monitoraggio della variabile' with tabs for 'Variabili globali utente - Micro870', 'Variabili locali - RA_TON_LD', and 'Variabili di sistema'. The main table displays the following data:

Nome	Valore logico	Valore fisico	Blocca	Tipo di
TON_1	<input type="checkbox"/>	TON
in	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
MaxTIME	T#3s	N/D	<input type="checkbox"/>	TIME
output	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
elapse	T#1s493ms	N/D	<input type="checkbox"/>	TIME

Vedere anche

[Configurazione delle istruzioni del timer](#) a pagina 677

[Istruzioni del timer](#) a pagina 677

TONOFF (ritardo, on, off)

Ritarda l'attivazione di un'uscita su un ramo True, quindi ritarda la disattivazione dell'uscita su ramo False.

Dettagli operazione:

- Se si usa il parametro EN con TONOFF, il timer inizia a incrementare quando EN è impostato su TRUE e continua a incrementare anche se EN è impostato su FALSE.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

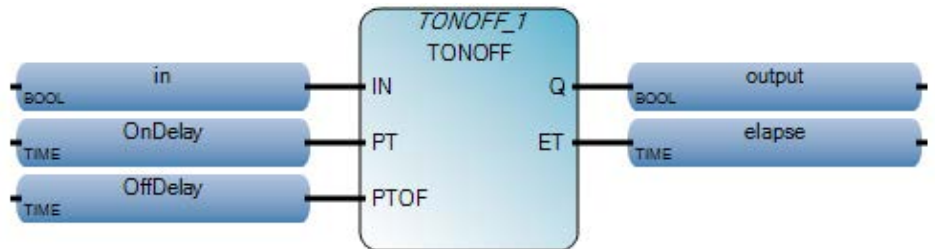
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



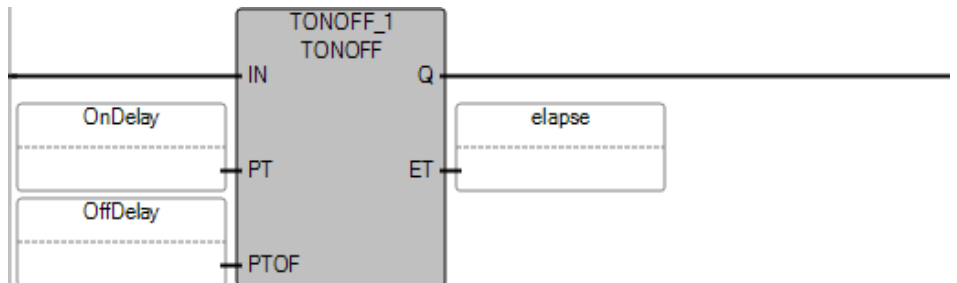
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Controllo ingresso. TRUE: fronte di salita rilevato (IN passa da 0 a 1): <ul style="list-style-type: none"> • avvia il timer Ritardo all'eccitazione (PT). • Se la durata ritardo off programmato (PTOF) non è trascorsa, riavviare il timer Ritardo all'eccitazione (PT). FALSE: fronte di discesa rilevato (IN passa da 1 a 0): <ul style="list-style-type: none"> • Se non è trascorso il tempo di Ritardo all'eccitazione (PT) programmato, arrestare il timer PT e ripristinare ET. • Se è trascorso il tempo di Ritardo all'eccitazione (PT) programmato, avvia il timer Ritardo alla diseccitazione (PTOF).
PT	Ingresso	TIME	L'impostazione del tempo di Ritardo all'eccitazione definito utilizzando il tipo di dati Time.
PTOF	Ingresso	TIME	L'impostazione del tempo di Ritardo alla diseccitazione definito utilizzando il tipo di dati Time.
Q	Uscita	BOOL	TRUE: il tempo del Ritardo all'eccitazione programmato è trascorso e il tempo del ritardo alla diseccitazione non è trascorso.
ET	Uscita	TIME	Tempo trascorso. I valori possibili variano da 0ms a 1193h2m47s294ms. Se il tempo del ritardo all'eccitazione programmato è trascorso e il timer del ritardo alla diseccitazione non è stato avviato, il tempo trascorso (ET) rimane al valore del ritardo all'eccitazione (PT). Se il tempo Off-delay programmato è trascorso e il timer Off-delay non parte, il tempo trascorso (ET) resta al valore di off-delay (PTOF) fino a che si verifica nuovamente il fronte di salita.

Esempio di Programmazione a blocchi funzionali TONOFF



Esempio di Diagramma ladder TONOFF



Esempio di Testo strutturato TONOFF

```
TONOFF_1 {
void TONOFF_1(BOOL IN, TIME PT, TIME PTOF)
Tipo: TONOFF, Ritardare un'uscita attiva (true), quindi ritardare un'uscita inattiva (false).

1 OnDelay := T#3s;
2 OffDelay := T#5s;
3 TONOFF_1(in, OnDelay, OffDelay);
4 output := TONOFF_1.Q;
5 elapse := TONOFF_1.ET;
```

Risultati

Nome	Valore logico	Valore fisico	Blocca	Tipo di dati
TONOFF_1	<input type="checkbox"/>	TONOFF
in	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
OnDelay	T#3s	N/D	<input type="checkbox"/>	TIME
OffDelay	T#5s	N/D	<input type="checkbox"/>	TIME
output	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
elapse	T#1s60ms	N/D	<input type="checkbox"/>	TIME

Nome	Valore logico	Valore fisico	Blocca	Tipo di dati
TONOFF_1	<input type="checkbox"/>	TONOFF
in	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
OnDelay	T#3s	N/D	<input type="checkbox"/>	TIME
OffDelay	T#5s	N/D	<input type="checkbox"/>	TIME
output	<input type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
elapse	T#1s446ms	N/D	<input type="checkbox"/>	TIME

Vedere anche

[Configurazione delle istruzioni del timer](#) a pagina 677

[Istruzioni del timer](#) a pagina 677

TP (tempo impulsi)

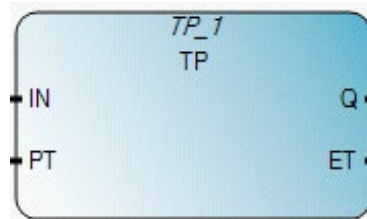
Sul fronte di salita, incrementa un timer interno fino a un valore dato. Se il timer è trascorso, il tempo interno viene ripristinato.

Dettagli operazione:

- Se si usa il parametro EN con TP, il timer inizia a incrementare quando EN è impostato su TRUE e continua a incrementare anche se EN è impostato su FALSE.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

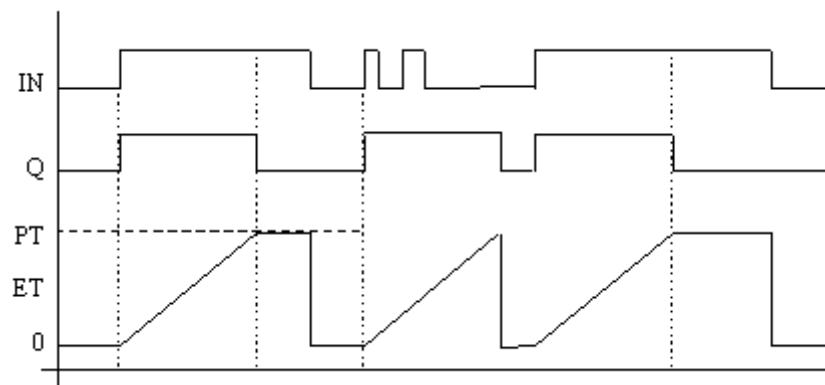
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



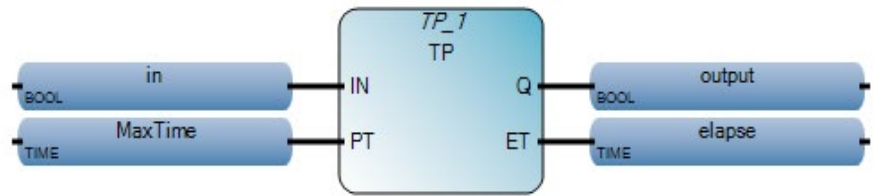
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	TRUE: se fronte di salita, il timer interno inizia a incrementare (se non sta già aumentando). FALSE: se il tempo del timer è trascorso, il timer interno viene ripristinato. Qualsiasi modifica a IN durante il conteggio non ha alcun effetto.
PT	Ingresso	TIME	Tempo massimo programmato definito utilizzando il tipo di dati Time.
Q	Uscita	BOOL	TRUE: il timer sta contando. FALSE: il timer non sta contando.
ET	Uscita	TIME	Tempo trascorso. I valori possibili variano da 0ms a 1193h2m47s294ms.

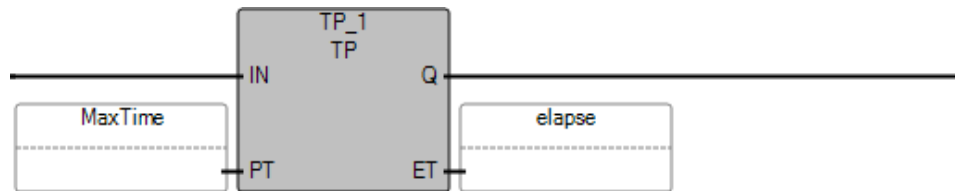
Diagramma di temporizzazione TP



Esempio di Programmazione a blocchi funzionali TP



Esempio di Diagramma ladder TP



Esempio di Testo strutturato TP

```

TP_1 (
void TP_1(BOOL IN, TIME PT)
Tipo: TP, Tempo di impulso

1 MaxTime := T#3s;
2 TP_1(in, MaxTime);
3 output := TP_1.Q;
4 elapse := TP_1.ET;

```

Risultati

Monitoraggio della variabile				
Variabili globali utente - Micro870 Variabili locali - RA_TP_LD Variabili di sistema - Micro870 I/O - Micro8				
Nome	Valore logico	Valore fisico	Blocca	Tipo di dati
TP_1	<input type="checkbox"/>	TP
in	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
MaxTime	T#3s	N/D	<input type="checkbox"/>	TIME
output	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL
elapse	T#593ms	N/D	<input type="checkbox"/>	TIME

Vedere anche

[Configurazione delle istruzioni del timer](#) a pagina 677

RTO (timer di ritrasmissione, Ritardo all'eccitazione)

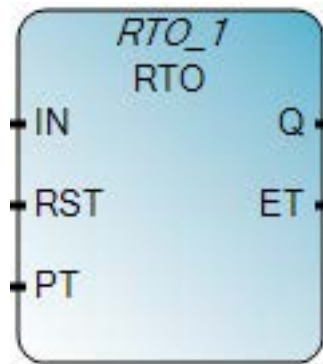
Incrementa un timer interno quando l'ingresso è attivo, ma non ripristina il timer interno quando l'ingresso diventa inattivo.

Dettagli operazione:

- Controllori <Micro810> o <Micro820>, il timer RTO interno non persiste per tutta la durata del ciclo di alimentazione. Per tenere attivo il timer interno, impostare il parametro di configurazione Memorizzato su True.
- Controllore <Micro830> o <Micro850>, il timer RTO interno persisterà per tutto il ciclo di alimentazione.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

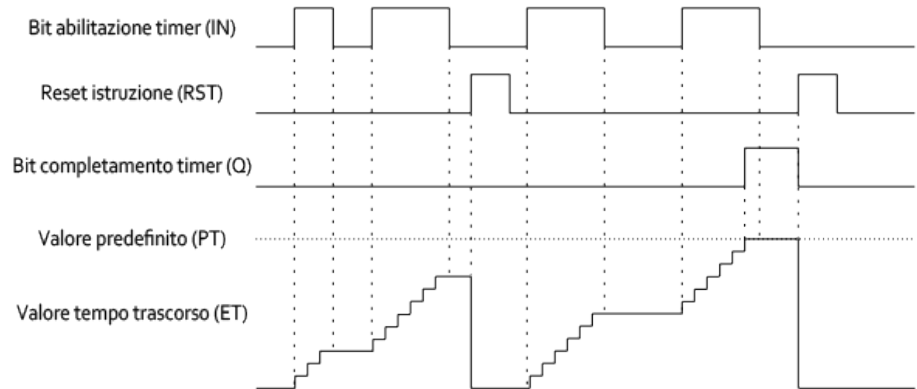
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



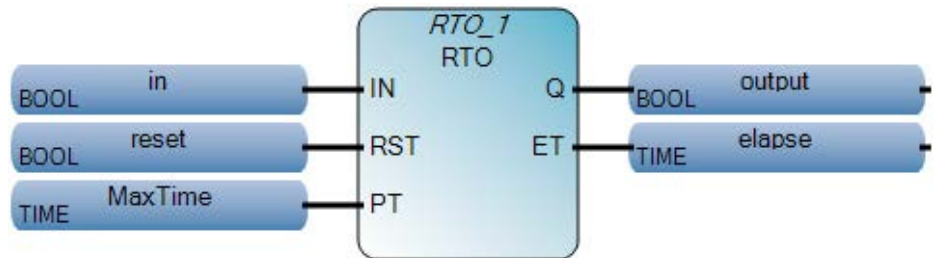
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
IN	Ingresso	BOOL	Controllo ingresso. TRUE: fronte di salita, inizia ad aumentare il timer interno. FALSE: fronte di discesa, si arresta e non viene eseguito il timer interno.
RST	Ingresso	BOOL	TRUE: fronte di salita, viene eseguito il reset del timer interno. FALSE: non viene eseguito il timer interno.
PT	Ingresso	TIME	Tempo di Ritardo all'eccitazione massimo programmato. PT è definito con i dati ora Durata.
Q	Uscita	BOOL	TRUE: il tempo del ritardo all'eccitazione programmato è trascorso. FALSE: il tempo del ritardo all'eccitazione del programma non è trascorso.
ET	Uscita	TIME	Tempo trascorso. I valori variano da 0 ms a 1193 h 2 m 47 s 294 ms. ET è definito con il tipo di dati Time.

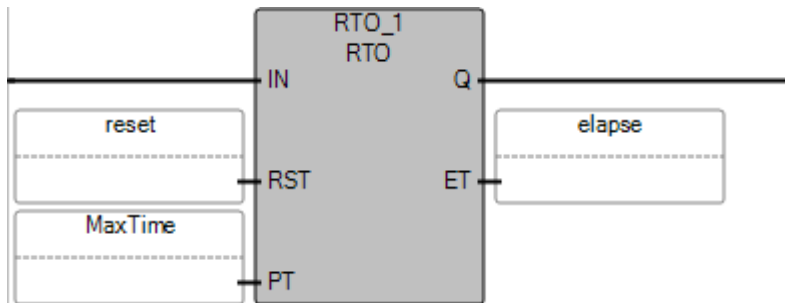
Esempio di diagramma di temporizzazione RTO



Esempio di Programmazione a blocchi funzionali RTO



Esempio di Diagramma ladder RTO



Esempio di Testo strutturato RTO

```
RTO_1 (
    void RTO_1(BOOL IN, BOOL RST, TIME PT)
    Tipo: RTO, Ritardare un'uscita attiva (true). Mantenere il tempo trascorso fino al reset.

    1 MaxTime := T#3s;
    2 RTO_1(in, reset, MaxTime);
    3 output := RTO_1.Q;
    4 elapse := RTO_1.ET;
```

Vedere anche

[Configurazione delle istruzioni del timer](#) a pagina 677

[Istruzioni del timer](#) a pagina 677

DOY (anno di controllo per Real-Time clock)

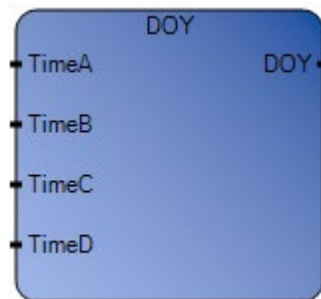
Attiva un'uscita se il valore Real-Time Clock (RTC) è compreso nell'intervallo dell'impostazione Ora dell'anno.

Dettagli operazione:

- Se non è presente un RTC, l'uscita è sempre OFF.
- Configurare i parametri di inserimento della durata con intervalli validi, secondo quanto specificato nella tabella Tipo di dati DOYDATA. Un valore non valido porta in errore il controllore se TimeX.Enable è impostato su TRUE ed è presente, e abilitato, un RTC.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

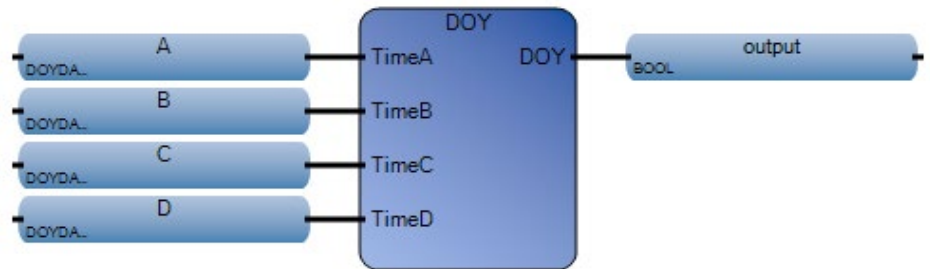
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue l'operazione. FALSE: non esegue l'operazione.
TimeA	Ingresso	DOYDATA	Impostazione di Year Time del Canale A: Per configurare TimeA, utilizzare il tipo di dati DOYDATA.
TimeB	Ingresso	DOYDATA	Impostazione di Year Time del Canale B. Per configurare TimeB, utilizzare il tipo di dati DOYDATA.
TimeC	Ingresso	DOYDATA	Impostazione di Year Time del Canale C. Per configurare TimeC, utilizzare il tipo di dati DOYDATA.
TimeD	Ingresso	DOYDATA	Impostazione di Year Time del Canale D. Per configurare TimeD, utilizzare il tipo di dati DOYDATA.
DOY	Uscita	BOOL	Se TRUE, il valore del Real Time Clock rientra nel range dell'impostazione Year Time di uno dei quattro canali.

Tipo di dati DOYDATA

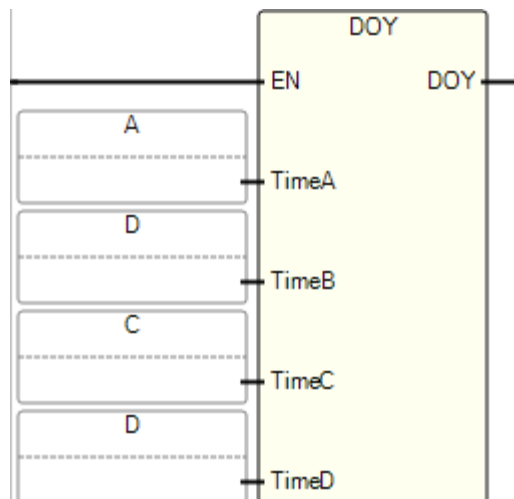
Nella tabella seguente è descritto il tipo di dati DOYDATA.

Parametri	Tipo di dati	Descrizione
Enable	BOOL	TRUE:Enable; FALSE:Disable
YearlyCenturial	BOOL	Tipo di timer (0:timer Yearly; 1:Centurial timer).
YearOn	UINT	Valore Year On (deve essere nel set [2000...2098]).
MonthOn	USINT	Valore Month On (deve essere nel set [1...12]).
DayOn	USINT	Valore Day On (deve essere nel set [1...31], determinato dal valore "MonthOn").
YearOff	UINT	Valore Year Off (deve essere nel set [2000...2098]).
MonthOff	USINT	Valore Month Off (deve essere nel set [1...12]).
DayOff	USINT	Valore Day Off (deve essere nel set [1...31], determinato dal valore "MonthOff").

Esempio di programmazione a blocchi funzionali DOY



Esempio di diagramma ladder DOY



Esempio di testo strutturato DOY

```
DOY(
  BOOL DOY(DOYDATA TimeA, DOYDATA TimeB, DOYDATA TimeC, DOYDATA TimeD)
  Attivare l'uscita quando il valore dell'orologio in tempo reale è all'interno dell'intervallo di anni.
```

```
1 output := DOY(A, B, C, D);
```

(* Equivalenza ST: *)

```
TESTOUTPUT := DOY(TIMEA1, TIMEB1, TIMEC1, TIMED1);
```

Vedere anche

[Configurazione delle istruzioni del timer](#) a pagina 677

[Istruzioni del timer](#) a pagina 677

TDF (differenza di tempo)

Calcola la differenza di tempo tra TimeA e TimeB.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

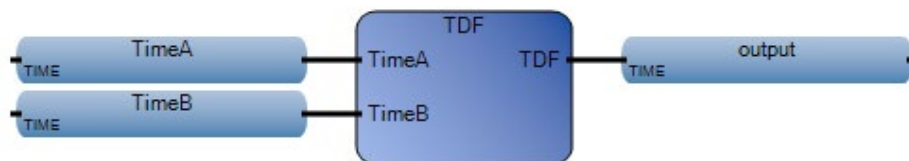
Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



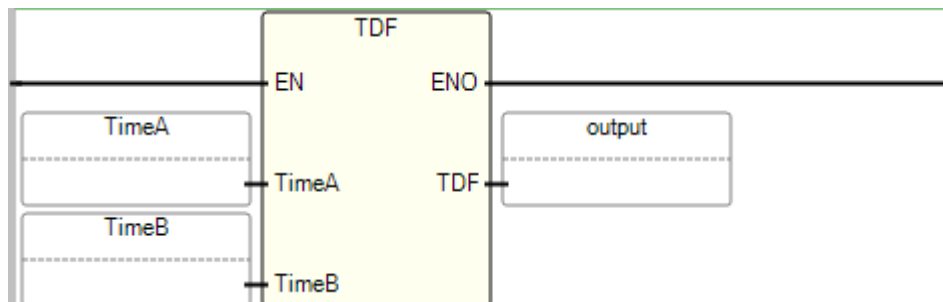
Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

Parametri	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. TRUE: esegue il calcolo corrente. FALSE: il calcolo non avviene. Applicabile alle programmazioni in Ladder Diagram.
TimeA	Ingresso	TIME	Il tempo di avvio per il calcolo della differenza di tempo.
TimeB	Ingresso	TIME	Il tempo di fine per il calcolo della differenza di tempo.
ENO	Uscita	BOOL	Abilita uscita. Applicabile alle programmazioni in Ladder Diagram.
TDF	Uscita	TIME	La differenza di tempo tra le due ore in ingresso. TDF è il nome o ID PIN

Esempio di programmazione a blocchi funzionali TDF



Esempio di diagramma ladder TDF



Esempio di testo strutturato TDF

```
TDF {
    TIME TDF(TIME TimeA, TIME TimeB)
    Calcolare la differenza di durata.

1 | TimeA := T#10s;
2 | TimeB := T#5s;
3 | output := TDF(TimeA, TimeB);
```

(* Equivalenza ST: *)

```
TESTTIMEDIFF := TDF(TESTTIME1, TESTTIME2);
```


Risultati

Nome	Valore logico	Valore fisico	Blocca	Tip
TimeA	T#10s	N/D	<input type="checkbox"/>	TIME
TimeB	T#5s	N/D	<input type="checkbox"/>	TIME
output	T#49d17h2m42s296ms	N/D	<input type="checkbox"/>	TIME

Vedere anche

[Configurazione delle istruzioni del timer](#) a pagina 677

[Istruzioni del timer](#) a pagina 677

TOW (settimana di controllo per Real Time Clock)

Attiva un'uscita se il valore del Real-Time Clock (RTC) è compreso nell'intervallo dell'impostazione Ora della settimana.

Dettagli operazione:

- Se non è presente un RTC, l'uscita è sempre OFF.
- Configurare i parametri di ingresso Time con intervalli validi, secondo quanto specificato nel Tipo di dati TOWDATA. Un valore non valido porta in errore il controllore se TimeX.Enable è impostato su TRUE ed è presente, e abilitato, un RTC.

Lingue supportate: Programmazione a blocchi funzionali, Diagramma ladder, Testo strutturato.

Questa istruzione vale per i controllori Micro810, Micro820, Micro830, Micro850, Micro870 e Micro800 Simulator.



Utilizzare questa tabella per determinare i valori del parametro per questa istruzione.

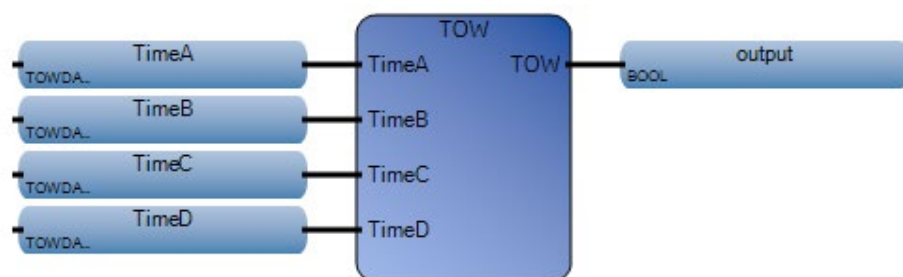
Parametro	Tipo di parametro	Tipo di dati	Descrizione
EN	Ingresso	BOOL	Abilitazione istruzioni. Se EN = TRUE esegue la funzione. Se EN = FALSE non esegue la funzione.
TimeA	Ingresso	TOWDATA	Impostazione dell'ora del giorno per il canale A. Utilizzare il tipo di dati TOWDATA per definire TimeA.
TimeB	Ingresso	TOWDATA	Impostazione dell'ora del giorno per il canale B. Utilizzare il tipo di dati TOWDATA per definire TimeB.
TimeC	Ingresso	TOWDATA	Impostazione dell'ora del giorno per il canale C. Utilizzare il tipo di dati TOWDATA per definire TimeC.
TimeD	Ingresso	TOWDATA	Impostazione dell'ora del giorno per il canale D. Utilizzare il tipo di dati TOWDATA per definire TimeD.
TOW	Uscita	BOOL	Se TRUE, il valore del Real Time Clock è compreso nell'intervallo dell'Ora del giorno impostata per uno qualsiasi dei quattro canali.

Tipo di dati TOWDATA

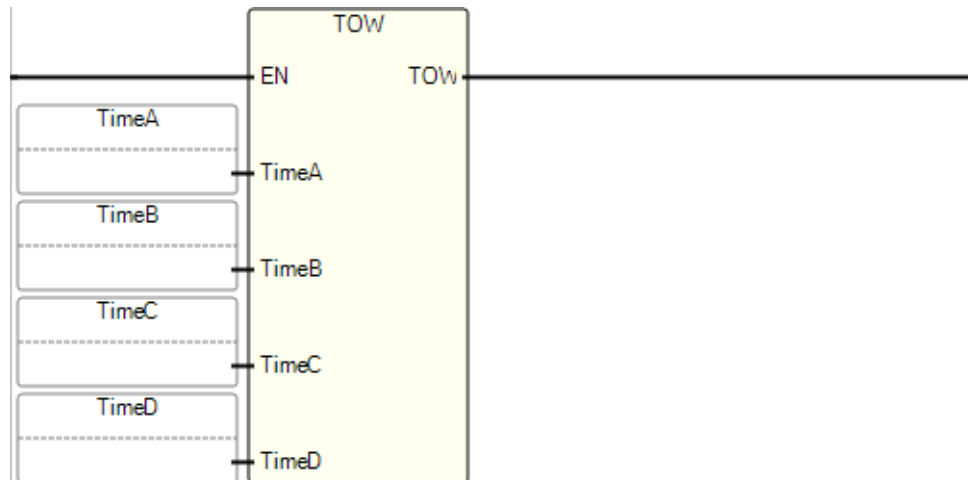
La seguente tabella descrive i tipi di dati TOWDATA:

Parametri	Tipo di dati	Descrizione
Enable	BOOL	TRUE: Abilita; FALSE: Disabilita.
DailyWeekly	BOOL	Tipo di timer (0:timer giornaliero; 1:timer settimanale).
DayOn	USINT	Valore Giorno della settimana il (deve essere in set [0 .. 6]).
HourOn	USINT	Valore Ora alle (deve essere in set [0 .. 23]).
MinOn	USINT	Valore Minuto alle (deve essere in set [0 .. 59]).
DayOff	USINT	Valore Giorno della settimana OFF (deve essere in set [0 .. 6]).
HourOff	USINT	Valore Ora OFF (deve essere in set [0 .. 23]).
MinOff	USINT	Valore Minuto OFF (deve essere in set [0 .. 59]).

Esempio di programmazione a blocchi funzionali TOW



Esempio di diagramma ladder TOW



Esempio di testo strutturato TOW

```
TOW(
  BOOL TOW(TOWDATA TimeA, TOWDATA TimeB, TOWDATA TimeC, TOWDATA TimeD)
  Attivare l'uscita quando il valore dell'orologio in tempo reale è all'interno dell'intervallo di settimane.
```

```
1| output := TOW(TimeA, TimeB, TimeC, TimeD);
(* Equivalenza ST: *)
TESTOUTPUT := TOW(TIMEA, TIMEB, TIMEC, TIMED) ;
```

Risultati

Variabili globali utente - Micro870		Variabili locali - RA_TOW_LD		Variabili di sistema - Micro870		I/O - Micro870		Parole definite	
Nome		Valore logico	Valore fisico	Blocca	Tipo di dati				
TimeA		<input type="checkbox"/>	TOWDATA				
	TimeA.Enable	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL				
	TimeA.DailyWeekly	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL				
	TimeA.DayOn	1	N/D	<input type="checkbox"/>	USINT				
	TimeA.HourOn	10	N/D	<input type="checkbox"/>	USINT				
	TimeA.MinOn	20	N/D	<input type="checkbox"/>	USINT				
	TimeA.DayOff	2	N/D	<input type="checkbox"/>	USINT				
	TimeA.HourOff	15	N/D	<input type="checkbox"/>	USINT				
	TimeA.MinOff	30	N/D	<input type="checkbox"/>	USINT				
TimeB		<input type="checkbox"/>	TOWDATA				
	TimeB.Enable	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL				
	TimeB.DailyWeekly	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL				
	TimeB.DayOn	2	N/D	<input type="checkbox"/>	USINT				
	TimeB.HourOn	15	N/D	<input type="checkbox"/>	USINT				
	TimeB.MinOn	20	N/D	<input type="checkbox"/>	USINT				
	TimeB.DayOff	3	N/D	<input type="checkbox"/>	USINT				
	TimeB.HourOff	10	N/D	<input type="checkbox"/>	USINT				
	TimeB.MinOff	30	N/D	<input type="checkbox"/>	USINT				
TimeC		<input type="checkbox"/>	TOWDATA				
	TimeC.Enable	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL				
	TimeC.DailyWeekly	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL				
	TimeC.DayOn	3	N/D	<input type="checkbox"/>	USINT				
	TimeC.HourOn	20	N/D	<input type="checkbox"/>	USINT				
	TimeC.MinOn	10	N/D	<input type="checkbox"/>	USINT				
	TimeC.DayOff	4	N/D	<input type="checkbox"/>	USINT				
	TimeC.HourOff	25	N/D	<input type="checkbox"/>	USINT				
	TimeC.MinOff	55	N/D	<input type="checkbox"/>	USINT				
TimeD		<input type="checkbox"/>	TOWDATA				
	TimeD.Enable	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL				
	TimeD.DailyWeekly	<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL				
	TimeD.DayOn	5	N/D	<input type="checkbox"/>	USINT				
	TimeD.HourOn	10	N/D	<input type="checkbox"/>	USINT				
	TimeD.MinOn	15	N/D	<input type="checkbox"/>	USINT				
	TimeD.DayOff	6	N/D	<input type="checkbox"/>	USINT				
	TimeD.HourOff	35	N/D	<input type="checkbox"/>	USINT				
	TimeD.MinOff	40	N/D	<input type="checkbox"/>	USINT				
output		<input checked="" type="checkbox"/>	N/D	<input type="checkbox"/>	BOOL				

Vedere anche

[Configurazione delle istruzioni del timer](#) a pagina 677

[Istruzioni del timer](#) a pagina 677

Assistenza Rockwell Automation

Utilizzare queste risorse per accedere alle informazioni di supporto.

Centro supporto tecnico	Trova assistenza per video, domande frequenti, chat, forum degli utenti e aggiornamenti delle notifiche dei prodotti.	rok.auto/supporto
Knowledge base	Articoli della Knowledge Base di Access.	rok.auto/knowledgebase
Numeri di telefono del supporto tecnico locale	Individua il numero di telefono per il tuo paese.	rok.auto/supporto telefonico
Literature Library	Trova istruzioni di installazione, manuali, brochure e pubblicazioni di dati tecnici.	rok.auto/letteratura
Area compatibilità e download prodotti (PCDC)	Ottieni assistenza per determinare il modo in cui i prodotti interagiscono, controllare le funzionalità e le funzionalità e trovare il firmware associato.	rok.auto/pcdc

Feedback documentazione

I commenti ci consentono di migliorare le esigenze della documentazione. In caso di suggerimenti su come migliorare questo documento, completare il modulo all'indirizzo rok.auto/docfeedback.

Rifiuti di apparecchiature elettriche ed elettroniche (WEEE)



Al termine del ciclo di vita, queste attrezzature dovrebbero essere raccolte separatamente da eventuali rifiuti urbani non utilizzati.

Rockwell Automation mantiene le attuali informazioni ambientali sul prodotto sul suo sito web all'indirizzo rok.auto/pec.

Allen-Bradley, expanding human possibility, Logix, Rockwell Automation e Rockwell Software sono marchi di Rockwell Automation, Inc.

Ethernet/IP è un marchio di ODVA, Inc.

I marchi che non appartengono a Rockwell Automation sono di proprietà delle rispettive aziende.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Contattaci su    

rockwellautomation.com

expanding human possibility®

AMERICHE: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

EUROPA/MEDIO ORIENTE/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgio, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

ASIA PACIFICO: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

ITALIA: Rockwell Automation S.r.l., Via Ludovico di Breme 13 A, 20156 Milano, Tel: +39 02 334471, Fax: +39 02 33447701, www.rockwellautomation.it

SVIZZERA: Rockwell Automation AG, Via Cantonale 27, 6928 Manno, Tel: 091 604 62 62, Fax: 091 604 62 64, Customer Service: Tel: 0848 000 279