



GE Fanuc Automation

Controllori programmabili

Uso del software PC Control

GFK-1424B-IT

Agosto 1998

Indicazioni Diffida, Avvertenze e note usate in questo manuale

Diffida

L'indicazione "Diffida" ha lo scopo di mettere in evidenza che in questa apparecchiatura esistono, o possono essere associate al suo impiego, tensioni, correnti e temperature pericolose, o altre condizioni in grado di causare lesioni personali.

Questa indicazione è utilizzata quando un'eventuale disattenzione rischia di provocare lesioni personali o danni materiali.

Avvertenza

Questa indicazione viene data quando il mancato rispetto delle necessarie precauzioni comporta il rischio di danni materiali.

Nota

Le note richiamano semplicemente l'attenzione su informazioni particolarmente importanti per la comprensione e il buon uso dell'apparecchiatura.

Questo documento è basato sulle informazioni disponibili al momento della sua pubblicazione. Per quanto sia stato fatto il massimo sforzo per rendere il suo contenuto il più accurato possibile, esso non pretende di coprire tutti i dettagli e le possibili variazioni di hardware e software, né di prevedere tutte le eventualità che possono verificarsi durante l'installazione, il funzionamento e la manutenzione dell'apparecchiatura alla quale si riferisce. E' possibile che alcune delle caratteristiche qui descritte non riguardino tutti i sistemi hardware e software. La GE Fanuc Automation non assume alcun obbligo di comunicare ai possessori di questo documento le modifiche successivamente apportatevi.

La GE Fanuc Automation non fa alcuna dichiarazione, né fornisce garanzie espresse, implicite o legali rispetto all'esattezza, alla completezza, alla sufficienza o all'utilità delle informazioni contenute in questo documento, né assume al riguardo alcuna responsabilità. Inoltre non viene data alcuna garanzia rispetto alla commerciabilità o all'adeguatezza allo scopo.

I seguenti sono marchi depositati della GE Fanuc Automation North America, Inc.

Alarm Master	Genet	PowerMotion	Series One
CIMPLICITY	Genius	ProLoop	Series Six
CIMPLICITY PowerTRAC	Genius PowerTRAC	PROMACRO	Series Three
CIMPLICITY 90-ADS	Helpmate	Series Five	VuMaster
CIMSTAR	Logicmaster	Series 90	Workmaster
Field Control	Modelmaster		

Revisioni di questo manuale

La release 2.0 del software PC Control fornisce le seguenti nuove prestazioni:

Notificazione errori di runtime — pagina 2-6

Autorizzazione di emergenza con chiave criptata — pagina 2-14

Spegnimento intelligente con monitor UPS — pagine 2-19 e 2-30

Nuova finestra di dialogo per le opzioni del sistema— pagina 2-26

Enumerazione strutture e blocchi funzionali— pagina 2-29

Visualizzazione riferimento IEC — pagina 2-29

Importazione/esportazione di simboli mediante un file .CSV, per simboli globali, e GE Fanuc PCIM e variabili PCIM — pagina 3-41

Miglioramento della gestione degli errori di compilazione — pagina 4-102

Editazione in linea senza interferenza con l'esecuzione — pagina 4-115

Linguaggi di programmazione ST e IL per programmi standalone — sezioni 4 e 5 del capitolo 4

Edito ST pilotato da menu — sezione 4 del capitolo 4

Interfaccia ActiveX — pagina 6-33

Supporto al driver PCIF2 per l'interfaccia con gli I/O della Serie 90-30 — Appendice C

Configurazione APM e DSM; creazione e scaricamento del programma 0 — pagina E-2 e aiuto in linea per la configurazione della scheda Motion

Programmazione Motion Control e controlli dell'interfaccia con l'operatore — Appendice E

Supporto per l'indirizzamento indiretto (puntatori) — Appendice F

Funzione Change MMI — "Funzioni miscellanee" nell'appendice G

Tipi aggiuntivi di temporizzatori — "Temporizzatori estesi" nell'appendice G

Nuove istruzioni di conversione e temporizzatori estesi — Appendice G

Supporto dei blocchi funzionali C definiti dall'utente — fare riferimento alla nota applicativa APP981R1 fornita sul CD del software PC Control per i dettagli relativi all'uso dell'editor CFB

Contenuto di questo manuale

Capitolo 1. Introduzione: Fornisce i requisiti e le procedure per l'installazione del PC Control sul personal computer e spiega come usare la documentazione in linea.

Capitolo 2. Come iniziare: Fornisce una descrizione generale del funzionamento del PC Control e delle fasi principali in cui si suddivide la creazione di un'applicazione di comando. Spiega come creare un nuovo progetto destinato a contenere tutti i file associati.

Capitolo 3. Configurazione degli I/O: Spiega come configurare il GENIUS® I/O e gli I/O della Serie 90™-30 nell'ambito del PC Control.

Capitolo 4. Creazione di programmi applicativi: Descrive la creazione di programmi SFC (Sequential Function Chart), RLL (Relay Ladder Logic) , ST (Structured Text) e IL (Instruction List).

Capitolo 5. Esecuzione dei programmi applicativi: Spiega come avviare i sistemi di runtime ed eseguire un programma applicativo.

Capitolo 6. Creazione di applicazioni di interfaccia con l'operatore: Spiega come creare uno schermo interfaccia operatore mediante l'editor GUI integrato nel PC Control.

Appendice A. Modulo di interfaccia con Personal Computer per Genius I/O: Questa appendice descrive l'installazione del PCIM sul personal computer e la configurazione del modulo mediante l'utilità di configurazione del PCIM del software PC Control.

Appendice B. Personal Computer Interface (PCIF) per I/O della Serie 90-30: Questa appendice descrive l'installazione del PCIF sul personal computer.

Appendice C. Personal Computer Interface (PCIF2) per I/O della Serie 90-30: Questa appendice descrive l'installazione del PCIF sul personal computer.

Appendice D. Esempi di applicazioni: Fornisce passo per passo le istruzioni relative alla creazione di un programma applicativo mediante gli editor SFC e RLL. Viene anche costruito uno schermo interfaccia operatore per il monitoraggio e il controllo del programma applicativo.

Appendice E. Motion Control: Spiega come usare il set di istruzioni testuali conforme a RS-274D per le operazioni Motion Control.

Appendice F. Puntatori: (Per utenti avanzati) Descrive l'uso di puntatori per le operazioni di indirizzamento indiretto nel linguaggio ST.

Appendice G. Sommario del set di istruzioni

Appendice H. Glossario dei termini

Pubblicazioni correlate

GEK-90486-1	<i>Genius® I/O System User's Manual</i>
GFK-0898	<i>Series 90™-30 Programmable Controller I/O Module Specifications</i>
GFK-0826	<i>Field Control Distributed I/O and Control System User's Manual</i>
GFK-0356	<i>Series 90™-30 Programmable Controller Installation Manual</i>
GFK-1180	<i>CIMPLICITY® HMI for Window NT® and Windows® 95 Base System User's Manual</i>

Chapter 1	Introduzione	1-1
	Sezione 1: Installazione	1-2
	Prima di iniziare	1-2
	Aggiornamento della versione precedente	1-4
	Installazione del software PC Control	1-4
	Esecuzione e autorizzazione del software PC Control	1-5
	Sezione 2: Uso della documentazione in linea	1-7
	Aiuto in linea	1-7
Chapter 2	Come iniziare.....	2-1
	Sezione 1: Descrizione del Software.....	2-2
	Sottosistemi software	2-2
	Lo standard IEC 1131	2-3
	Linguaggi di programmazione IEC 1131-3	2-3
	Partenza veloce.....	2-4
	Sottosistemi di runtime del software PC Control	2-6
	Program Editor	2-7
	Livelli di accesso.....	2-7
	Tasti di scelta rapida dell'interfaccia operatore.....	2-9
	Descrizione dei menu.....	2-10
	Barre degli strumenti e barra di stato.....	2-16
	Risposte alle domande più frequenti	2-18
	Sezione 2: Progetti e applicazioni	2-20
	Gestione progetti.....	2-20
	Creazione di un nuovo progetto.....	2-20
	Apertura di un progetto.....	2-20
	Copia di un progetto.....	2-21
	Rinominazione di un progetto	2-21
	Attivazione di una configurazione.....	2-21
	Programmi applicativi.....	2-22
	Creazione di un nuovo programma.....	2-22
	Apertura di un programma.....	2-22
	Salvataggio di un file	2-23
	Stampa di un programma.....	2-23
	Stampa della Cross-Reference del programma.....	2-24
	Gestione dei programmi applicativi.....	2-24
	Chiusura dei programmi applicativi	2-24
	Visualizzazione dei programmi.....	2-25

Dimensionamento in scala di un programma per adeguarlo alla finestra	2-25
Attivazione/disattivazione della visualizzazione dei commenti	2-25
Impostazione delle opzioni del sistema.....	2-26
Localizzazione dei simboli secondo IEC.....	2-29
Configurazione dell'UPS	2-30
Sezione 3: Descrizione generale del funzionamento del programma.....	2-32
Attivazione di una configurazione	2-32
Prima scansione con la configurazione attiva.....	2-32
Sequenza di spegnimento.....	2-32
Funzionamento normale.....	2-33
Nomi dei file.....	2-34
Chapter 3 Configurazione degli I/O	3-1
Sezione 1: Descrizione generale.....	3-2
Conflitti hardware.....	3-2
Porte I/O.....	3-3
Indirizzo di memoria.....	3-3
Interrupt – IRQ.....	3-3
I file di configurazione	3-4
Elementi di una configurazione.....	3-4
Creazione di una nuova configurazione.....	3-4
Editazione di un file di configurazione esistente.....	3-4
Attivazione di un file di configurazione.....	3-4
Navigazione all'interno dell'utilità di configurazione	3-5
La finestra di dialogo System Configuration.....	3-5
Intervallo di scansione degli I/O.....	3-5
La finestra di dialogo Define Board.....	3-6
Sezione 2: Configurazione di GENIUS I/O	3-7
Il modulo interfaccia PC (PCIM = PC Interface Module)	3-7
Finestra di dialogo Global Data Setup	3-11
Finestra di dialogo Global Data: Device	3-11
Finestra di dialogo Global Data: Device, Word.....	3-11
Definizione degli indirizzi sul bus Genius	3-11
Finestra di dialogo Genius Bus Address Definitions	3-12
Finestra di dialogo Discrete Point Information	3-13
Trasmissione dei datagrammi con il driver PCIM	3-15

	Sezione 3: Configurazione degli I/O della Serie 90-30	3-16
	L'interfaccia PC (PCIF = PC Interface).....	3-16
	Finestra di dialogo PCIF Board Definition	3-16
	Finestra di dialogo Rack Definition	3-18
	Finestra di dialogo Module	3-18
	Definizione delle connessioni delle porte digitali.....	3-19
	Definizione delle connessioni delle porte analogiche.....	3-19
	Sezione 4: Configurazione di altri bus di campo.....	3-20
	Configurazione degli I/O DeviceNet	3-20
	Prestazioni	3-20
	Installazione e configurazione delle unità sulla rete DeviceNet.....	3-20
	Configurazione degli I/O Profibus.....	3-22
	Sezione 5: Scambio dinamico dei dati (DDE = Dynamic Data Exchange).....	3-38
	L'interfaccia DDE	3-38
	Comunicazioni DDE con il software Microsoft Excel.....	3-38
	Trasferimento di dati ad Excel	3-38
	Trasferimento di dati al sistema di comando	3-39
	Trasferimento di valori al sistema di comando su richiesta	3-39
	Sezione 6: Importazione/esportazione di configurazioni.....	3-41
Chapter 4	Creazione di programmi applicativi.....	4-1
	Sezione 1: Configurazione dei simboli	4-2
	Che cosa sono i simboli.....	4-2
	Visibilità dei simboli.....	4-2
	Identificatori.....	4-3
	Costanti letterali	4-4
	Costanti letterali numeriche	4-4
	Stringhe di caratteri letterali	4-4
	Costanti letterali di durata.....	4-5
	Costanti letterali che rappresentano ora e data.....	4-6
	Tipi di dati.....	4-7
	BOOL (Booleano).....	4-7
	BYTE	4-7
	DATE.....	4-8
	DINT (doppio intero).....	4-8
	DWORD (doppia parola)	4-8
	INT (intero)	4-8

REAL	4-9
STRING.....	4-9
TIME.....	4-9
TOD (TIME_OF_DAY).....	4-10
UINT (Intero senza segno).....	4-10
WORD.....	4-10
Tipi di dati generici.....	4-11
Tipi di dati definiti dall'utente	4-11
Array.....	4-12
Puntatori	4-12
Symbol Manager.....	4-12
Apertura del Symbol Manager.....	4-12
Creazione di un simbolo.....	4-14
Editazione di un simbolo.....	4-14
Copia di un simbolo	4-16
Cancellazione di un simbolo	4-16
Denominazione di un bit in un simbolo.....	4-16
Editazione dei tipi di dati definiti dall'utente.....	4-17
Uso dei simboli.....	4-19
Trascinamento	4-19
Enumerazioni	4-19
Simboli di sistema.....	4-20
Simboli di sistema predefiniti.....	4-20
Simboli di Run-Time	4-21
Parole riservate	4-22
Esportazione di simboli per CIMPPLICITY HMI.....	4-25
Sezione 2: Programmazione RLL	4-26
Descrizione generale dei diagrammi RLL (Relay Ladder Logic)	4-26
Come vengono risolti i programmi applicativi RLL	4-27
Come viene risolta la semplice logica a relè	4-27
Come viene risolta la logica RLL quando si usano i blocchi funzionali.....	4-28
Creazione di un programma RLL.....	4-28
Sezione 3: Programmazione SFC.....	4-41
Descrizione generale dei diagrammi funzionali sequenziali.....	4-41
Che cosa sono i passi.....	4-41
Uso dei simboli di sistema del passo.....	4-43
Che cosa sono le azioni	4-44
Gestore delle azioni.....	4-50
Che cosa sono le transizioni	4-50

Che cosa sono le divergenze.....	4-51
Mezzi di controllo del flusso del programma.....	4-53
Come vengono risolti gli SFC	4-56
Come sono valutate le transizioni.....	4-57
Uso delle divergenze simultanee.....	4-58
Uso dei passi macro	4-58
Estensioni dello standard IEC 1131-3	4-59
Creazione di diagrammi funzionali sequenziali.....	4-60
Creazione di un programma SFC	4-60
Uso della barra del menu e degli strumenti SFC	4-62
Operazioni sui passi	4-63
Operazioni sulle transizioni	4-66
Operazioni sui passi macro	4-67
Operazioni sulle azioni	4-68
Aggiunta dei controlli di flusso del programma SFC	4-71
Integrazione di testo strutturato in un SFC	4-80
Documentazione di un programma SFC.....	4-81
Sezione 4: Programmazione con il linguaggio ST	4-82
Introduzione	4-82
Apertura di un documento in testo strutturato.....	4-82
Editazione di testo strutturato in un passo SFC.....	4-82
Introduzione delle istruzioni.....	4-83
Editazione del testo strutturato	4-83
Descrizione del linguaggio	4-84
Espressioni.....	4-84
Operatori	4-84
Operatori per i puntatori	4-85
Sintassi del testo strutturato.....	4-85
Frase di assegnazione.....	4-86
Frase BREAK	4-87
Frase CASE	4-87
Commenti	4-88
Frase EXIT	4-89
Frase IF.....	4-90
Frase INCLUDE	4-91
Frase FOR.....	4-91
Richiamo di funzione	4-93
ETICHETTA	4-93
Frase REPEAT.....	4-94
Frase SCAN.....	4-95
Frase WHILE.....	4-95

Operatori del linguaggio ST.....	4-97	
Sommario dei tipi di frasi.....	4-98	
Uso di Pid in un programma applicativo	4-98	
Controllo del flusso dei programmi applicativi RLL e ST	4-100	
Monitoraggio e prova dei programmi applicativi e dei simboli	4-102	
Sezione 5: Programmazione con il linguaggio IL	4-105	
Introduzione	4-105	
Apertura di un documento IL	4-105	
Introduzione delle istruzioni.....	4-105	
Editazione delle istruzioni.....	4-106	
Descrizione del linguaggio	4-107	
Sintassi IL.....	4-107	
Operatori	4-108	
Funzioni e blocchi funzionali	4-110	
Esempi di programmazione.....	4-112	
Esempi di uso degli operatori.....	4-113	
Esempio di richiamo di blocco funzionale	4-114	
Sezione 6: Editazione in linea.....	4-115	
Che cosa è l'editazione in linea	4-115	
Regole.....	4-116	
Regola generale.....	4-116	
Simboli	4-116	
I/O	4-116	
Programmi RLL.....	4-117	
Programmi SFC	4-117	
Operazioni su file.....	4-117	
Programmi ST	4-118	
Programmi IL	4-118	
Chapter 5	Esecuzione dei programmi applicativi.....	5-1
Sottosistemi di runtime	5-2	
Esecuzione di un singolo programma.....	5-2	
Esecuzione del programma attivo.....	5-2	
Cancellazione dell'esecuzione di un programma	5-3	
Configurazione dei programmi per l'esecuzione automatica	5-4	
Avvio dei programmi con un file batch	5-4	
Avvio di programmi con il comando Run With Restart	5-5	
Monitoraggio del flusso di corrente.....	5-5	
Programmi RLL attivi.....	5-5	
Programmi SFC attivi.....	5-5	

	Visualizzazione dello stato dei programmi applicativi.....	5-6
Chapter 6	Creazione di applicazioni di interfaccia con l'operatore.....	6-1
	Sezione 1: Descrizione generale dell'Operator Interface Editor	6-2
	Avvio del software Operator Interface	6-2
	Livelli di accesso.....	6-2
	Introduzione del codice di accesso.....	6-3
	Variazione dei codici di accesso	6-3
	Modi attivazione e edit	6-4
	Commutazione dei modi del software Operator Interface	6-4
	Controllo di programmi RLL mediante l'interfaccia con l'operatore	6-5
	Sezione 2: Schermi dell'interfaccia con l'operatore	6-6
	Funzioni del software Operator Interface	6-6
	Creazione di un nuovo file interfaccia operatore.....	6-6
	Apertura di un file interfaccia operatore.....	6-6
	Salvataggio di un file interfaccia operatore.....	6-6
	Operazioni relative agli schermi	6-7
	Creazione di un nuovo schermo di interfaccia con l'operatore	6-7
	Cancellazione di uno schermo.....	6-7
	Copia di uno schermo	6-7
	Rinominazione di uno schermo	6-8
	Selezione dello schermo iniziale.....	6-8
	Selezione dello schermo da editare.....	6-8
	Sezione 3: Operazioni relative ai controlli.....	6-9
	Aggiunta di controlli.....	6-9
	Editazione di controlli.....	6-9
	Selezione di controlli	6-10
	Spostamento di controlli.....	6-10
	Dimensionamento di controlli	6-11
	Copia, taglia e incolla.....	6-11
	Cancellazione di controlli.....	6-12
	Allineamento di controlli.....	6-12
	Spostamento dei controlli davanti/dietro	6-12
	Sezione 4: Operazioni sui simboli	6-13
	Editazione di simboli.....	6-13
	Attivazione di configurazioni	6-13

	Sezione 5: Creazione di applicazioni di interfaccia con l'operatore.....	6-14
	Controlli standard	6-14
	Introduzione.....	6-14
	Bar.....	6-16
	Bitmap.....	6-18
	Box	6-19
	Click Button.....	6-20
	Continuous Button	6-23
	Gauge	6-25
	Indicator	6-27
	Numeric Display	6-29
	Pannello Selected Program Status	6-30
	Slide	6-30
	Text	6-32
	Controlli ActiveX	6-33
	Introduzione.....	6-33
	Limitazioni dei controlli ActiveX.....	6-33
	Controlli ActiveX e controlli standard.....	6-34
	Fornitori di controlli ActiveX.....	6-34
	Registrazione di controlli ActiveX	6-35
	Inserimento di controlli ActiveX	6-36
	Editazione di controlli ActiveX.....	6-37
Appendix A	Modulo di interfaccia con Personal Computer per Genius I/O..	A-1
	Descrizione.....	A-1
	Installazione e configurazione.....	A-3
	Connessione del PCIM al bus.....	A-12
	Rimozione del PCIM dal Bus	A-13
	Specifiche	A-14
	Caratteristiche elettriche del PCIM	A-15
	Ricerca guasti	A-16
Appendix B	Interfaccia con Personal Computer per I/O della Serie 90-30 (PCIF)	B-1
	Installazione della scheda PCIF-30.....	B-1
	Configurazione della scheda PCIF-30	B-4
Appendix C	Interfaccia con Personal Computer per I/O della Serie 90-30 (PCIF2)	C-1
	Introduzione	C-1

	Compatibilità	C-2
	Descrizione dell'hardware.....	C-2
	Ponticelli.....	C-3
	Connettori.....	C-4
	Microinterruttori.....	C-6
	Guida all'avviamento rapido	C-7
Appendix D	Esempi di applicazioni.....	D-1
	Esercizio 1: Creazione di un nuovo progetto.....	D-1
	Esercizio 2: Creazione di una nuova configurazione del sistema	D-2
	Esercizio 3: Creazione di un SFC.....	D-5
	Esercizio 4: Creazione dei simboli.....	D-9
	Esercizio 5: Aggiunta della logica delle transizioni	D-11
	Esercizio 6: Introduzione dei comandi in testo strutturato	D-13
	Esercizio 7: Aggiunta ed editazione dei blocchi delle azioni.....	D-14
	Esercizio 8: Esecuzione del programma dell'esempio	D-19
	Esercizio 9: Creazione dello schermo di interfaccia con l'operatore	D-20
Appendix E	Motion Control.....	E-1
	Configurazione per Motion Control	E-2
	Programmazione Motion Control.....	E-3
	Aggiunta di codice Motion Control a un SFC.....	E-3
	Miglioramenti del software PC CONTROL allo standard RS-274D.....	E-3
	Uso delle frasi Motion Control	E-4
	Simboli Motion Control predefiniti	E-11
	Configurazione delle opzioni di Motion Control.....	E-18
	Controllo del flusso del programma in applicazioni Motion	E-19
	Uso del richiamo macro G65 con Motion Control	E-22
	Monitoraggio ed esecuzione di programmi applicativi Motion Control.....	E-24
	Inclusione di Testo strutturato nel codice Motion Control.....	E-27
	Funzioni Motion Control del linguaggio Testo strutturato	E-29
	Controlli Motion dell'interfaccia con l'operatore.....	E-31
	Pannello jog.....	E-31
	Pannello di stato monoasse.....	E-32
	Pannello di stato multiasse.....	E-33
	Finestra dei blocchi RS274.....	E-34

Indice

Appendix F	Uso dei puntatori nel linguaggio ST	F-1
	Indirizzamento	F-2
	Operatori riservati ai puntatori	F-2
	Definizione dei simboli di tipo puntatore	F-3
	Note sui puntatori	F-3
	Puntatori ad array	F-5
Appendix G	Sommario del set di istruzioni.....	G-1
	Set di istruzioni del linguaggio RLL.....	G-1
	Set di istruzioni del linguaggio SFC	G-7
	Set di istruzioni del linguaggio Testo strutturato	G-10
	Set di istruzioni del linguaggio Lista di istruzioni.....	G-17
Appendix H	Glossario dei termini	H-1

Benvenuti in PC Control. PC Control, è il nuovo pacchetto software predisposto per l'ambiente Windows NT[®] 4.0 dalla GE Fanuc per supportare il collegamento a Genius[®], Field Control[™], DeviceNet[™], PROFIBUS, e agli I/O Serie 90[™]-30. PC Control è un pacchetto integrato che include le funzioni di programmazione e configurazione, l'interfaccia grafica con l'operatore (GUI : graphical user interface), e il software di runtime.

Con il software PC Control potete:

- Creare un progetto applicativo
- Configurare l'hardware degli I/O
- Creare ed editare i simboli necessari
- Creare ed editare un programma applicativo
- Eseguire il programma sotto Windows NT

Questo capitolo contiene le seguenti informazioni:

- Requisiti e procedure per l'installazione del software PC Control sul vostro computer
- Linee di guida per l'esecuzione del software PC Control sotto Windows[®] 95
- Uso dell'aiuto in linea del prodotto

Windows NT e Windows[®] 95 sono marchi registrati Microsoft Corporation.
DeviceNet è un marchio registrato della Open DeviceNet Vendor Association, Inc.

Sezione 1: Installazione

Questa sezione fornisce i requisiti raccomandati per l'esecuzione sul vostro computer del software PC Control e ne descrive la procedura di installazione.

Prima di iniziare

Per Windows NT

I requisiti minimi del vostro computer per l'installazione e l'esecuzione di PC Control sotto Windows NT sono i seguenti.

Windows NT 4.0 richiede il Service Pack 3.

	<i>Raccomandati</i>
CPU	Pentium, 100MHz
RAM	32 MB (preferibilmente 64 MB)
Disco fisso	100 MB liberi
Unità CD-ROM	Sì
Monitor	VGA minimo SVGA raccomandato
Tastiera e dispositivo di puntamento	Sì
Sistema UPS	Raccomandato

Sono richieste appropriate interfacce I/O. E' possibile che siano richieste altre risorse per la contemporanea esecuzione di altre applicazioni.

Controllate anche il documento *Important Product Information*, spedito insieme alla vostra release, per informarvi sulle eventuali modifiche dell'ultimo minuto.

Per Windows 95

Diffida

Windows 95 non è un sistema operativo in modo protetto. Non raccomandiamo di controllare alcun processo con il software PC Control eseguito sotto Windows 95. Usare il sistema operativo Windows 95 solo per lo sviluppo dei programmi.

Il software PC Control supporta lo sviluppo off line e il funzionamento in modo demo sotto Windows 95, con simulazione degli I/O. Per eseguire PC Control sotto Windows 95, il vostro sistema deve soddisfare almeno i seguenti requisiti minimi.

	<i>Raccomandato</i>
CPU	Pentium, 100MHz
RAM	32 MB (preferibilmente 64 MB)
Disco fisso	100 MB liberi
Unità CD-ROM	Sì
Monitor	VGA minimo SVGA raccomandato
Tastiera e dispositivo di puntamento	Sì
Sistema UPS	Raccomandato

Controllate anche il documento *Important Product Information*, spedito insieme alla vostra release, per informarvi sulle eventuali modifiche dell'ultimo minuto.

Aggiornamento della versione precedente

Se sul vostro sistema è installata una precedente versione autorizzata di PC Control, prima di installare la Versione 2.0 eseguite la seguente procedura

1. Dal pannello di controllo, selezionare “Installazione applicazioni”..
2. Se non lo è già, selezionare la pagina “Installa/Rimuovi”.
3. Evidenziare PC e cliccare sul pulsante Aggiungi/Rimuovi.
4. Cliccare su “Sì” per rimuovere la versione precedente.
5. Usare Explorer per cancellare la cartella BIN nel percorso
\\PCCONTROL\BIN
6. Seguire le istruzioni per l’installazione e l’autorizzazione del software PC Control Software.

Installazione del software PC Control

Prima di iniziare, leggete il documento “ Important Product Information”, speditovi insieme alla vostra release per informarvi su eventuali modifiche dell’ultimo minuto o speciali note operative riguardanti la procedura di installazione

Le presenti istruzioni per l’installazione assumono che l’unità CD-Rom sia d:. Se il nome dell’unità CD-ROM è diverso, eseguite le sostituzioni necessarie nel testo di queste istruzioni.

Il processo di installazione può essere tralasciato in qualsiasi momento. Per installare PC Control sotto Windows NT:

1. Inserire il CD-Rom nell’unità CD-ROM. Setup è invocato automaticamente. Fare doppio click sull’icona a fianco di Setup.Exe, Comparirà lo schermo PC Control Setup.
2. Leggere le istruzioni. Cliccare su Next per proseguire.
3. Comparirà la finestra di dialogo per la selezione della cartella di destinazione. Il percorso selezionato per difetto è C:\CIMPLICITY\PCCONTROL. Per cambiare questa selezione cliccare sul pulsante Browse e selezionare la cartella desiderata. Per accettare la cartella selezionata cliccare su Next.
4. I file di PC Control saranno installati nella cartella di destinazione. Il completamento di questo processo può richiedere alcuni minuti.

Esecuzione e autorizzazione del software PC Control

Dopo aver installato PC Control, potete eseguire il software nel modo Demo, oppure potete registrarlo per operare nel modo Autorizzato. Il modo Demo è limitato a 32 punti di I/O e a 2 ore di funzionamento.

Esecuzione di PC Control nel modo Demo:

1. Localizzare l'icona Control Program Editor del gruppo programmi:



Comparirà la finestra di dialogo “Key Validation”.

2. Cliccare su **Demo**. Comparirà la finestra di dialogo “License Information”.
3. Leggere il testo della convenzione e cliccare su **I Agree** per accettarla e continuare. Comparirà la finestra di dialogo “Enter Password”.
4. Introdurre la password standard 4, 5, 6, 7 e cliccare su **Enter**.

A questo punto inizierà l'esecuzione di PC Control nel modo Demo.

Esecuzione di PC Control nel modo Autorizzato:

L'esecuzione nel modo Autorizzato è possibile solo per il software acquistato. I CD demo non possono essere autorizzati.

1. Localizzare l'icona Control Program Editor del gruppo programmi:



Comparirà la finestra di dialogo “Key Validation”.

2. Cliccare su **Authorize**. Comparirà la finestra di dialogo “License Information”.
3. Leggere il testo della convenzione e cliccare su **I Agree** per accettarla e continuare. Comparirà la finestra di dialogo Authorization, che contiene il codice sito.
4. Per autorizzare il software, seguire le istruzioni visualizzate.

Digitare il codice sito e cliccare su **OK**. Comparirà la finestra di dialogo “Enter Password”.

5. Introdurre la password standard 4, 5, 6, 7 e cliccare su **Enter**.

A questo punto inizierà l’esecuzione di PC Control nel modo Autorizzato.

Sezione 2: Uso della documentazione in linea

Quando usate PC Control la vostra principale fonte di informazioni è l'aiuto in linea. L'aiuto in linea include una demo, che vi sarà di aiuto nella prima fase di uso del software, e specifici capitoli che descrivono le prestazioni del prodotto e il modo di usarle.

Aiuto in linea

L'aiuto in linea è stato concepito per fornirvi facilmente e rapidamente le necessarie informazioni su PC Control. Usate l'aiuto per sapere come eseguire una certa procedura, o per trovare la definizione di un termine che non vi è familiare. All'aiuto è possibile accedere in tre modi: visualizzazione del suo contenuto, ricerca di un argomento, ricorso all'aiuto in base al contesto.

Contenuto

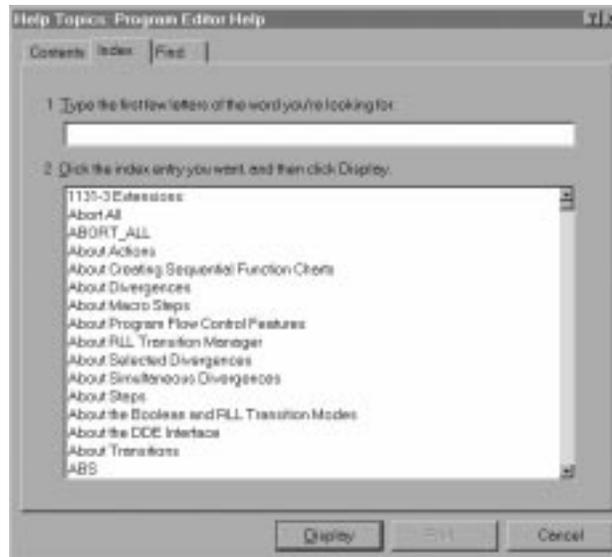
La pagina Contents contiene un elenco dei vari argomenti dell'aiuto. Per accedere a questa pagina selezionare Help sul menu principale e cliccare sulla linguetta Contents.



Un doppio click sul nome dell'argomento o sull'icona corrispondente, apre il libro e visualizza l'elenco dei sottoargomenti. Per visualizzarne il contenuto, fare un doppio click sul nome del sottoargomento.

Ricerca

La pagina Index dell'aiuto vi permette di specificare l'argomento che vi interessa o di scorrere lungo una lista di argomenti in ordine alfabetico. Per accedere a questa pagina, scegliere Help sul menu principale e selezionare Search for Help On (oppure cliccare sul pulsante  della barra degli strumenti Help Search:



Per iniziare una ricerca, introduce il nome dell'argomento nell'apposito campo o fate scorrere la lista. Per accedere all'aiuto relativo ad un argomento fate un doppio click sul suo nome.

Aiuto sensibile al contesto (F1)

L'aiuto sensibile al contesto vi permette di accedere rapidamente alle informazioni relative alla finestra o alla casella di dialogo attive, premendo il tasto F1.

Capitolo 2

Come iniziare

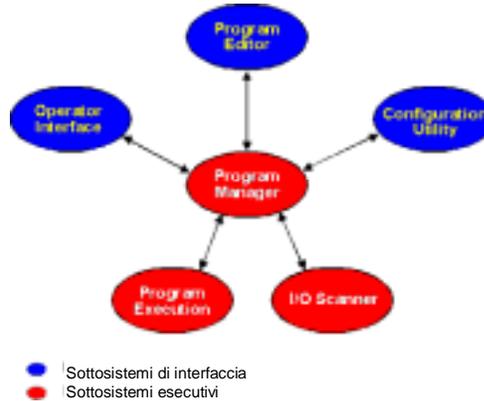
Per iniziare ad usare il software PC Control, aprire l'utilità Program Editor. Questa utilità permette di: creare un nuovo progetto che conterrà il programma applicativo e i file di configurazione, creare i programmi applicativi, accedere all'utilità di configurazione degli I/O (I/O Configuration).

Questo capitolo:

- Fornisce una descrizione generale dei sottosistemi che compongono il software PC Control
- Fornisce una descrizione generale dello standard IEC 1131
- Spiega come cambiare i livelli e i codici di accesso
- Spiega come creare e gestire i progetti attraverso il Program Editor
- Spiega come creare e gestire i programmi applicativi
- Fornisce una descrizione generale del funzionamento del programma

Sezione 1: Descrizione del Software

Il software PC Control comprende i seguenti sottosistemi:



Sottosistemi software

Utilità di configurazione L'utilità di configurazione permette di definire la struttura degli I/O e di assegnare dei nomi ai punti di I/O e alle porte di I/O.

Interfaccia operatore (PC Control GUI) L'utilità interfaccia operatore permette di manipolare l'ambiente dell'interfaccia con l'operatore, l'insieme di schermi, messaggi, e finestre di cui dispone l'operatore per comandare e controllare il funzionamento di una macchina o di un processo.

Program Editor Il Program Editor permette di creare programmi nei formati Ladder Diagram o Relay Ladder Logic (*linguaggio a contatti*), Sequential Function Chart (SFC+) (*diagramma funzionale sequenziale*), Structured Text (*testo strutturato*), e Instruction List (*lista di istruzioni*) e di avviarne l'esecuzione.

Program Manager Il Program Manager prepara i programmi per l'esecuzione e gestisce l'interfaccia con i sottosistemi Program Execution e I/O Scanner.

Program Execution Il sottosistema Program Execution esegue i programmi creati nei formati Relay Ladder Logic, Sequential Function Chart (SFC+), Structured Text e Instruction List. Questo sottosistema software è eseguito con la priorità REAL_TIME process di Windows NT.

I/O Scanner Il sottosistema I/O Scanner scandisce le unità I/O fisiche e rende disponibili le informazioni al sottosistema Program Execution. Esso è eseguito con la priorità REAL_TIME process. Lo I/O Scanner acquisisce tutti gli ingressi dell'interfaccia I/O e le trasmette tutte le uscite.

Lo standard IEC 1131

L'IEC 1131-3 è uno standard internazionale predisposto dalla International Electrotechnical Commission. Lo standard IEC 1131-3 specifica la sintassi e la semantica dei linguaggi di programmazione per controllori programmabili.

Linguaggi di programmazione IEC 1131-3

I linguaggi IEC-1131-3 si suddividono in linguaggi testuali e linguaggi grafici. Questi linguaggi possono essere usati insieme in un ambiente integrato di programmazione. Per i miglioramenti apportati dal software PC Control allo standard IEC 1131-3, fare riferimento all'aiuto in linea.

Linguaggi testuali	
Instruction List (IL)	Il formato Instruction List (<i>lista di istruzioni</i>) è simile al linguaggio assembler.
Structured Text (ST)	Il formato Structured Text (<i>testo strutturato</i>) è adatto per algoritmi complessi e per operazioni su stringhe e file. I linguaggi procedurali (in forma testuale) agevolano la manipolazione di strutture di dati. Questo formato è conveniente per chi ha familiarità con i linguaggi strutturati di alto livello, come BASIC, Pascal, C, ecc.
Linguaggi grafici	
Ladder Diagram (RLL)	Il formato Ladder Diagram (<i>linguaggio a contatti</i>) è comunemente usato sui controllori programmabili per costruire programmi a logica discreta. Un diagramma ladder è simile allo schema elettrico di un circuito logico a relè equivalente. Un diagramma ladder contiene due linee verticali che rappresentano le barre di alimentazione. La barra di sinistra è una sorgente di corrente elettrica eccitata quando il programma è in esecuzione. La barra di destra assorbe il flusso di corrente proveniente dalla barra di sinistra. Le due barre sono collegate da linee orizzontali (che ricordano i pioli di una scala (ladder), in inglese rung) sulle quali sono posizionate le istruzioni che formano la logica del programma.
Sequential Function Chart (SFC)	Il formato SFC (<i>diagramma funzionale sequenziale</i>) è adatto per il controllo delle sequenze di macchina e per applicazioni di controllo caratterizzate da più modi operativi, ad esempio manuale ed automatico. Un SFC rappresenta il programma applicativo come una serie di passi sequenziali. I passi sono collegati tramite archi orientati e il controllo passa dall'uno all'altro mediante transizioni che si verificano durante l'esecuzione del programma. La logica di controllo può essere inserita nel passo o nell'azione ad esso associata. Questa logica viene eseguita quando il passo diventa attivo. Il controllo va al passo successivo quando è soddisfatta la condizione alla quale è subordinata la transizione.
Function Block Diagram	Il formato Function block diagram (<i>diagramma a blocchi funzionali</i>) è adatto per il trattamento di segnali analogici e per qualsiasi processo di controllo o algoritmo che sia eseguito con continuità. Nel linguaggio FBD, le funzioni e i blocchi funzionali sono rappresentati graficamente come blocchi etichettati con il nome della funzione, gli ingressi e le uscite. Un diagramma a blocchi funzionali consiste in una rete di funzioni e blocchi funzionali collegati da segnali. Questo formato non è attualmente supportato dal software PC Control.

Partenza veloce

Nota

Per installare o aggiornare il software PC Control, dovete aver eseguito il log in come “administrator”.

1. Installare tutto l'hardware del PC e le schede degli I/O.
2. Avviare Windows NT.
3. Installare il software PC Control e i driver degli I/O.
4. Usare l'utilità Program Editor per creare il nuovo progetto, che conterrà il programma applicativo e i file di configurazione.
5. Usare l'utilità Configuration per configurare l'hardware del sistema e definire i nomi simbolici degli I/O usati dai programmi applicativi. L'utilità Configuration si esegue dal menu File del Program Editor.
6. Usare il Program Editor per creare il programma applicativo.
7. Usare l'Operator Interface's Screen Editor (nel menu Tools) per costruire gli schermi dell'interfaccia uomo/macchina.
8. Usare Operator Interface e Program Editor per provare i programmi applicativi.
9. Se lo si desidera, configurare Windows NT per avviare automaticamente il software CIMPPLICITY® PC Control e i programmi applicativi all'accensione del sistema.

Avvio del software PC Control



Per un sistema di comando che include il software di interfaccia con l'operatore:

Nel Program manager, aprire il gruppo PC Control Applications. Fare doppio click sull'icona Operator Interface.

Oppure, se il Program Editor è in esecuzione, selezionare *Operator Interface* nel menu *Tools* del Program Editor..

Nota

Se i sottosistemi di runtime del PC Control non sono attivi, avere la possibilità di avviarli.

Avvio dei sottosistemi di runtime del PC Control



1. Dal Program Manager, aprire il gruppo PC Control Applications.
2. Fare doppio click sull'icona PC Control Runtime. Quando i sottosistemi sono attivi, nella parte bassa dello schermo compare l'icona Runtime minimizzata.

Quando sono attivi i sottosistemi di runtime, l'icona Runtime può comparire nella barra delle applicazioni o nell'area messaggi.

Se l'icona Runtime è nell'area messaggi, per chiudere i sottosistemi di runtime occorre usare il mouse. Se la barra delle applicazioni contiene il pulsante dell'applicazione Runtime, questa può essere chiusa usando la tastiera. Per far comparire Runtime nella barra delle applicazioni, accedere a *System Options* dal menu *Tools* del Program Editor e selezionare questa opzione nella *Display Properties*.

Chiusura dei sottosistemi di runtime di PC Control

1. Cliccare con il pulsante destro sull'icona Runtime visualizzata nella barra delle applicazioni.
2. Selezionare Shutdown Runtime.

Avvio del Program Editor



1. Dal Program Manager, aprire il gruppo PC Control Applications.
2. Cliccare sull'icona Program Editor.

Sottosistemi di runtime del software PC Control

L'icona PC Control Runtime rappresenta i sottosistemi Program Manager, Program Execution e I/O Scanner. Il log degli eventi (Event Log) ha la propria icona. I sottosistemi consistono in quanto segue:

Program Manager

Il Program Manager prepara i programmi per l'esecuzione e gestisce l'interfaccia esterna con i sottosistemi Program Execution e I/O Scanner.

Program Execution

I sottosistemi Program Execution hanno priorità real-time process; questo significa che il tempo di CPU viene dato a questo sottosistema prima che a tutte le altre applicazioni normali, inclusi gli aggiornamenti del mouse e gli accessi al disco.

I/O Scanner

I sottosistemi I/O Scanner hanno priorità real-time process; questo significa che il tempo di CPU viene dato a questo sottosistema prima che a tutte le altre applicazioni normali, inclusi gli aggiornamenti del mouse e gli accessi al disco.

All'avvio l'I/O Scanner cerca le schede I/O. Se non ne rileva, visualizza un messaggio. Per ignorare l'hardware mancante ed eseguire PC Control nel modo simulazione, premere il pulsante Simulate. Nel modo simulazione gli ingressi e le uscite risiedono in una tabella in memoria, ma i loro valori non sono letti o scritti sui dispositivi I/O. Il software di simulazione scritto dall'utente può simulare gli I/O scrivendo nella mappa degli ingressi e leggendo la mappa delle uscite.

Dall'interno del sottosistema Operator Interface, è possibile attivare il Program Editor premendo il pulsante appropriato o selezionando la corrispondente voce del menu Tools.

Event Log

Il sottosistema Event Log memorizza gli eventi, i messaggi e gli errori, registrando il momento in cui hanno luogo. L'Event Log è avviato e fermato automaticamente dai sottosistemi di runtime. Perché funzioni l'invio dei messaggi alla Output Window è necessario che l'Event Log sia in esecuzione.

Nota

Per configurare un sistema in modo che sia di solo runtime, copiare l'intera cartella del progetto da un sistema di sviluppo (fare attenzione che i due sistemi abbiano gli stessi percorsi per il programma e per i file di progetto). Un modo alternativo consiste nel dare temporaneamente al sistema la licenza di sviluppo, attivare il progetto desiderato e ritornare alla licenza di runtime. Un sistema di solo runtime non può modificare i progetti.

Program Editor

Il Program Editor permette di organizzare il proprio lavoro progetto per progetto. All'interno del Program Editor è possibile creare e gestire qualsiasi numero di progetti. In un dato momento può essere aperto un solo progetto. Questo è il progetto attivo. A ciascun progetto è associata una configurazione attiva. Quando viene aperto un progetto, si attiva la configurazione ad esso associata. Quando viene aperto un nuovo, progetto, esso è vuoto.

Livelli di accesso

I livelli di accesso e i codici di accesso (password) servono per controllare l'accesso ai programmi applicativi, agli schermi dell'interfaccia operatore e ai dati di configurazione. Esistono cinque livelli di accesso, a ciascuno dei quali corrisponde un diverso codice di accesso. La sottostante tabella descrive i privilegi di ciascun livello. Il livello di accesso più basso è il livello 0; ciascun livello di accesso successivo ha i privilegi dei livelli precedenti.

Livello di accesso	Privilegi
0	Attiva gli schermi dell'operatore e usa i controlli. Seleziona ed esegue i programmi SFC dall'interfaccia operatore.
1	Apri, visualizza, esegue e arresta i programmi.
2	Edita i programmi nei formati SFC, testo strutturato e lista di istruzioni. Esegue le funzioni di gestione progetto.
3	Edita i programmi RLL. Modifica l'interfaccia operatore. Modifica i file di configurazione del sistema.
4	Cambia le password per l'accesso ai livelli da 1 a 4..

I codice di accesso assunto per difetto è 4567.

Nota

Se il Program Editor è aperto la variazione del livello di accesso all'interno dell'interfaccia operatore non cambia il livello di accesso del Program Editor.

Introduzione del codice di accesso

I sottosistemi Program Editor e Operator Interface richiedono i codici di accesso. Quando viene avviato, il Program Editor presenta automaticamente la tastiera per l'introduzione del codice di accesso.

Per introdurre il codice di accesso:

Da:	Operazione:
Program Editor	Sulla tastiera visualizzata cliccare sui numeri corrispondenti al proprio codice di accesso e cliccare su <i>OK</i> .
Operator Interface	<ol style="list-style-type: none"> 1. Selezionare <i>Access</i> sulla barra del menu e selezionare <i>Password</i>. Comparirà la tastiera. 2. Cliccare sui numeri corrispondenti al proprio codice di accesso e cliccare su <i>OK</i>.

Per azzerare il codice di accesso:

Cliccare sul pulsante *Cancel* della tastiera visualizzata.

Variazione dei codici di accesso

I codici di accesso possono essere cambiati solo dall'interno dell'interfaccia operatore, specificando un codice di accesso al livello 4. Esiste un codice di accesso per ciascun livello.

Per cambiare la password di un livello di accesso:

1. Sulla barra del menu selezionare *Access - Password*. Comparirà una tastiera.
2. Cliccare quattro volte sul carattere '*'. Comparirà il messaggio: "Enter Access Level to Change".
3. Cliccare sul numero del livello di accesso per il quale si desidera eseguire la modifica e cliccare su *OK*.
4. Cliccare il nuovo numero di quattro cifre. Comparirà il messaggio: "Enter new password again".
5. Cliccare di nuovo la nuova password e cliccare su *OK*. Se la password cliccata corrisponde a quella specificata nel passo 4, essa diventa la nuova password del livello di accesso selezionato.

Nota

Per azzerare il codice di accesso cliccare sul pulsante *CANCEL* della tastiera visualizzata.

Se, nel modo operator interface edit, ai controlli dell'operatore si associano le funzioni di controllo, è possibile specificare un livello di accesso per controllare l'uso di ciascun controllo dell'operatore.

Tasti di scelta rapida dell'interfaccia operatore

Tasti	Azioni
<ESCAPE>	Mette fine alla maggior parte delle operazioni e delle finestre di dialogo.
<ENTER>	Edita il controllo selezionato.
<DELETE>	Cancella il o i controlli selezionati.
<CONTROL> o <SHIFT>	Durante la selezione dei controlli, aggiunge il controllo al gruppo selezionato se non lo è o lo rimuove dal gruppo se è già selezionato.
un tasto freccia	Sposta di un pixel il o i controlli selezionati nella direzione indicata dalla freccia.

Descrizione dei menu

Menu File

Voce	Pulsante	Descrizione
New Editor		Crea un nuovo file programma.
Open Editor		Apri un file programma esistente.
Close	-	Chiude tutte le finestre associate al file programma attivo.
Save		Salva il file programma attivo.
Save As	-	Salva il programma attivo con un altro nome.
Save All	-	Salva tutti i file programma aperti e le relative informazioni di progetto.
Print...		Stampa un programma.
Print XRef...	-	Stampa l'elenco delle variabili di un programma con l'indicazione dei punti e della frequenza con cui sono usate dal programma. Questa opzione compare solo quando è aperto un file programma.
Print Setup	-	Permette di modificare le opzioni di stampa.
New Config	-	Apri l'utilità Configuration con un nuovo file di configurazione I/O.
Open Config	-	Apri un file di configurazione I/O esistente.
Save Config		Salva la configurazione attiva.
Save Config As		Salva la configurazione attiva con un nuovo nome o percorso.
Import CSV to Config		Importa un file di configurazione delimitato da virgole.
Export Config to CSV		Esporta la configurazione attiva in un file delimitato da virgole, che può essere letto da un editore di testi o da un foglio elettronico.
Recently Used File List	-	Lista degli ultimi quattro file usati dal Program Editor.
Exit	-	Chiude il software PC CONTROL. Se i programmi aperti non sono stati salvati, il sistema chiede se lo si vuole fare prima di chiudere.

Menu Edit

Voce	Pulsante	Descrizione
Undo		Annulla l'ultima azione.
Redo		Ripete l'azione precedentemente annullata.
Cut		Rimuove l'oggetto selezionato e lo inserisce negli appunti.
Copy		Copia l'oggetto selezionato e lo inserisce negli appunti.
Paste		Incolla il contenuto degli appunti.
Delete	-	Cancella l'oggetto selezionato.
Select All		Seleziona tutto il testo contenuto in un documento ST o IL.

Voce	Pulsante	Descrizione
Edit Element...	-	Aprire la finestra di dialogo per l'elemento di programma selezionato.
New Element	-	Inserisce un elemento SFC o RLL nel punto selezionato.
New Function Block	-	Inserisce un blocco funzionale nel punto selezionato. E' disponibile solo se un programma RLL è aperto per la modifica.
New SFC Element	-	Inserisce un nuovo elemento SFC.
Step Properties	-	Modifica un passo SFC.
Insert ST Statement	-	Inserisce una frase ST.
Insert ST Function Calls	-	Inserisce una funzione o un blocco funzionale ST.
Insert IL Statement	-	Inserisce una frase IL.
Insert IL Function Calls	-	Inserisce una funzione o un blocco funzionale IL.
Find...		Cerca il testo specificato.
Find Next		Cerca la ricorrenza successiva del testo specificato.
Find Prev.		Cerca la ricorrenza precedente del testo specificato.
Replace...	-	Sostituisce il testo specificato con un nuovo testo.
Go To...	-	Salta al numero di rung specificato. Disponibile solo quando è aperto per la modifica un programma RLL.
Trace...		Cerca la ricorrenza successiva di una bobina di uscita con lo stesso nome simbolico del contatto selezionato. Disponibile solo quando è aperto per la modifica un programma RLL.
Insert Mode		Inserisce nuovi elementi prima del punto selezionato.
Append Mode		Appende nuovi elementi dopo il punto selezionato.
Boolean Transitions	-	Durante la modifica di un SFC, definisce, per tutti i programmi SFC aperti, come booleana, anziché di tipo RLL, la transizione per difetto. Non cambia le transizioni esistenti. Disponibile solo quando è aperto per la modifica un programma SFC.
Lock Algorithms	-	Durante la modifica di un SFC, permettere di proteggere con una password gli algoritmi dei passi. Disponibile solo quando è aperto per la modifica un programma SFC.

Menu View

Voce	Pulsante	Descrizione
Toolbar	-	Visualizza come icone i comandi dell'editor.
Status Bar	-	Visualizza la barra di stato della finestra dell'editor.
Contact/Coil Bar	-	Durante la modifica di un programma RLL, la barra Contatti/Bobine visualizza gli elementi utilizzabili nel programma. Disponibile solo quando è aperto per la modifica un programma RLL.
SFC Bar	-	Durante la modifica di un programma SFC, la barra degli strumenti SGC visualizza gli elementi utilizzabili nel programma. Disponibile solo quando è aperto per la modifica un programma SFC.

Voce	Pulsante	Descrizione
Function Block Palette	-	Durante la modifica di un programma RLL, la paletta dei blocchi funzionali visualizza i blocchi funzionali utilizzabili. Disponibile solo quando è aperto per la modifica un programma RLL.
Application Icon Palette		Durante l'editazione di un programma SFC, la paletta delle icone visualizza le icone utilizzabili. Disponibile solo quando è aperto per la modifica un programma SFC.
Accessory Bar		Visualizza la barra accessoria del testo strutturato e della lista di istruzioni, che inserisce automaticamente le frasi del programma.
Watch/Force Variables		Durante il runtime, questa finestra visualizza le variabili e il loro valore corrente. Come aiuto per la messa a punto del programma, è possibile specificare (forzare) differenti da quelli visualizzati. Disponibile solo quando un programma è aperto per la modifica.
I/O Faults		Visualizza la finestra degli errori I/O..
Output Window	-	In questa finestra compaiono i messaggi di runtime.
Program Status	-	Apri la finestra dello stato del programma.
Show Parse Errors		Visualizza la finestra degli errori di compilazione.
Program Comments		Visualizza o nasconde i commenti inseriti nel programma. Disponibile solo quando un programma è aperto per la modifica.
Scale to Fit Window		Ridimensiona il programma in modo che stia nella finestra visualizzata. Disponibile solo quando un programma è aperto per la modifica.
RLL Warnings	-	Attiva/disattiva la notificazione degli avvertimenti che si verificano durante la compilazione di un programma.
Disp. Symbol Location		Visualizza il numero di rack/slot/punto degli I/O simbolici.
Toggle FB Details		Attiva/disattiva la visualizzazione dei nomi simbolici e dei dati in tempo reale nei blocchi funzionali RLL. Disponibile solo quando è aperto per la modifica un programma RLL.
All Steps	-	Durante la modifica di un programma SFC, questa opzione permette di visualizzare il nome, la descrizione e l'icona di un passo o il corrispondente codice di programma. Disponibile solo quando è aperto per la modifica un programma SFC.
Show Symbol Enumerations		Attiva le enumerazioni dei simboli selezionate nella finestra di dialogo System Options.

Menu Project

Voce	Descrizione
New	Crea un nuovo progetto per la memorizzazione dei programmi applicativi e delle corrispondenti configurazioni.
Open	Apri un progetto esistente.
Copy	Copia un progetto esistente (inclusi tutti i programmi applicativi e tutti i file di configurazione) in un nuovo progetto.
Rename	Cambia il nome di un progetto esistente. Non ha effetti sui programmi applicativi e sui file di configurazione contenuti nel progetto.
Activate Config.	Attiva il file di configurazione corrente. Il file di configurazione corrente è visualizzato nella barra di stato, nella parte bassa dello schermo.

Menu Execute

Voce	Descrizione
Parse	Compila il programma
Run	Compila ed esegue il programma.
Run with Debug	Esegue il programma fino a una frase BREAK in testo strutturato; poi ferma l'esecuzione e mette il programma nello stato di "break". Se viene dato il comando RUN, le frasi BREAK sono ignorate. Disponibile solo quando è aperto per la modifica un programma SFC.
Single Step	Esegue un comando per volta. Disponibile solo quando è aperto per la modifica un programma SFC.
Run with Restart	Contrassegna il programma per l'esecuzione ogni volta che viene avviato il runtime. Dopo essere stato interrotto o aver dato luogo ad errori, un programma non è più contrassegnato per l'esecuzione alla ripartenza.
Stop	Ferma l'esecuzione di un programma al suo punto di arresto. Quando il programma viene fatto ripartire, la sua esecuzione riprende dal punto di arresto. Disponibile solo quando è aperto per la modifica un programma SFC.
Abort	Interrompe l'esecuzione di un programma. Quando il programma viene fatto ripartire, la sua esecuzione riprende dall'inizio.
Reset Estop and clear I/O Faults	Resetta i driver degli I/O e ristabilisce le comunicazioni con l'hardware I/O.
Startup Runtime Subsystems	Avvia i programmi dei sottosistemi di runtime.

Menu Icon

Voce	Descrizione
New App Icon	Accede alla Application Icon Step Library permettendo di creare passi predefiniti in libreria.
Edit App Icon	Permette di modificare i passi predefiniti nella libreria Application Icon.
Delete App Icon	Permette di cancellare dalla libreria Application uno o più passi predefiniti.

Menu Tools

Voce	Pulsante	Descrizione
Operator Interface	-	Avvia l'interfaccia operatore integrata.
Symbol Manager		Accede al Symbol Manager, che permette di creare e di modificare le variabili.
Action Manager	-	Accede all'Action Manager, che permette di rinominare e cancellare le azioni SFC.
RLL Transition Manager	-	Accede al Transition Manager, che permette di rinominare e cancellare le transizioni RLL, SFC.
Export Symbols	-	Esporta tutti i simboli globali scrivendoli in un file *.SNF. Disponibile solo con l'interfaccia CIMPPLICITY HMI.
System Options	-	Apri una finestra di dialogo che permette impostare una serie di parametri che influenzano il comportamento e l'aspetto del sistema (dimensione Heap, dimensione tabella dati, modalità di visualizzazione, enumerazioni dei simboli, ecc.).

Menu License

Voce	Descrizione
Authorize	Questa opzione si usa per creare una “licenza” software per il PC CONTROL. Comunicate, tramite fax o telefono, alla GE Fanuc Automation il vostro “Site Code” e riceverete una “Site Key” per autorizzare il software. Questa opzione è attiva solo quando il PC Control è stato avviato nel modo demo.
Upgrade Authorization	Quando siete pronti ad accrescere la funzionalità del PC CONTROL, potete usare questa opzione per generare un nuovo “Site Code”. Questa opzione è abilitata solo dopo che il software è stato autorizzato.
Register Transfer	Usate questa opzione per iniziare il trasferimento di una licenza software da un computer ad un altro. Inserite un dischetto vuoto nel computer non autorizzato e selezionate questa opzione per scrivere uno specifico file di identificazione del computer. Questa opzione è attiva solo quando il PC Control è stato avviato nel modo demo.
Transfer Out	Questo è il secondo passo del processo di trasferimento di una licenza software da un computer ad un altro. Dopo aver completato il processo Register Transfer sul computer non autorizzato, inserite il dischetto in quello autorizzato e selezionate questa opzione. La licenza software verrà rimossa dal computer e scritta sul dischetto. Questa licenza potrà essere registrata solo sul computer che ha scritto il file originale di identificazione sul dischetto. Questa opzione è attiva solo su un computer autorizzato.
Transfer In	Transfer In è l’ultimo passo del processo di trasferimento di una licenza software da un computer ad un altro. Inserire il dischetto nel computer che deve ricevere la licenza. Selezionare Transfer In per scrivere la licenza. Questo può essere fatto solo sul computer usato per scrivere sul dischetto il file di identificazione.
Temporary Authorization	Questa opzione vi permette di dare temporaneamente una completa autorizzazione di sviluppo a un sistema esclusivamente di runtime. Ciascun sistema di solo runtime viene consegnato con 30 minuti di autorizzazione temporanea gratuita; è possibile acquistare tempo addizionale. Questa opzione è valida per i sistemi di solo runtime. Per ottenere una copia gratuita dell’utilità Temporary Authorization, telefonate al numero che compare nella finestra Authorization. Quando viene selezionata questa opzione, compaiono i pulsanti relativi all’autorizzazione temporanea e il tempo residuo di autorizzazione temporanea disponibile: <i>OK</i> – attiva l’autorizzazione temporanea e avvia il corrispondente temporizzatore (purché esista del tempo residua disponibile). <i>Cancel</i> – ritorna al modo solo runtime. <i>Add More Minutes</i> – permette di acquistare tempo addizionale di autorizzazione temporanea per questo sistema. Telefonate al numero che compare nella finestra di dialogo Authorization e fornire il codice che compare nel campo <i>Code</i> . Vi sarà fornita una chiave da introdurre nel campo <i>Key</i> per poter aggiungere altro tempo di autorizzazione temporanea.
Reset Original Authorization	Disattiva l’autorizzazione temporanea e ferma il corrispondente temporizzatore. Questa opzione è valida per i sistemi di solo runtime.
Emergency Authorization	Dà una piena autorizzazione per tutte le prestazioni per quattro giorni. Dopo quattro giorni, il PC CONTROL ritorna al modo demo. Quando viene selezionata, il sistema avverte che si tratta di una autorizzazione valida una sola volta, finché non viene formattato il disco fisso. Se si continua, sul disco fisso viene scritta una identificazione.

Barre degli strumenti e barra di stato

Le barre degli strumenti contengono i pulsanti delle funzioni più comuni.

Barra degli strumenti standard

Pulsante	Strumento
	Nuovo
	Apri
	Salva
	Annulla
	Ripeti
	Taglia
	Copia
	Incolla
	Inserisci prima
	Inserisci dopo
	Stampa
	Informazioni
	Aiuto nel contesto
	Visualizza commenti
	Adeguo alla finestra
	Gestore simboli
	Finestra visualizzazione
	Modo transizioni booleane

Barra degli strumenti RLL

Pulsante	Strumento
	Selezione
	Contatto
	Etichetta
	Bobina
	Bobina di salto
	Bobina di transizione
	Ramo
	Nuovo rung

Barra degli strumenti SFC

Pulsante	Strumento
	Selezione
	Passo
	Passo macro
	Azione
	Transizione
	Etichetta
	Salto
	Ciclo
	Divergenza selettiva
	Divergenza simultanea
	Icona applicazione
	Commento

Barra di stato

La barra di stato si trova nella parte inferiore della finestra del Program Editor. Essa indica il nome della configurazione corrente e lo stato di:

- NUM lock
- SCROLL lock
- CAPS lock
- tasti della tastiera
- modi insert e append

Risposte alle domande più frequenti

PC Control richiede un PLC?

- PC Control non richiede un'unità PLC per eseguire la logica.
- PC Control esegue la logica nel PC senza alcuna necessità di PLC.
- PC Control può usare schede di scansione per PC per controllare rack di I/O, moduli e altri dispositivi di tipo PLC.

PC Control esegue il controllo di parità?

I PC possono essere acquistati e configurati per fornire il controllo di parità su tutta la memoria di sistema.

Il software PC Control non esegue il controllo di parità sui programmi applicativi, ma esegue in background i calcoli di checksum per tutti i programmi applicativi in esecuzione. Se viene rilevato un errore, il programma applicativo viene fermato in errore e l'operatore può decidere che cosa fare per farlo ripartire.

Che cosa succede nel caso di un errore di parità?

Se nel sistema rileva un errore di parità, Windows NT si ferma e visualizza uno schermo di errore con sfondo blu, che indica un Data Bus Error e la posizione nella quale è stato rilevato l'errore. Per continuare è necessario rieseguire il boot del sistema.

Il sistema PC Control fa uso di temporizzatori Watchdog e delle prestazioni Safe State I/O disponibili su alcuni sistemi I/O per rilevare la perdita di attività del controllore PC Control, e per mettere le uscite in una condizione predeterminata di sicurezza.

Per ulteriori informazioni contattare la GE Fanuc Automation.

Che cosa succede se manca la corrente?

Il PC e il sistema I/O vanno nello stato di perdita di alimentazione. Quando l'alimentazione elettrica viene ristabilita, il sistema I/O assume lo stato che assume normalmente o quello che, in sede di configurazione del sistema, è stato selezionato per il caso di perdita dell'alimentazione o del controllo. Se il sistema dispone di memoria non volatile o è protetto da una batteria di alimentazione della memoria, una applicazione PC CONTROL può riavviare gli I/O e i programmi applicativi dal punto di interruzione.

E' possibile avviare automaticamente il software PC Control?

Windows NT può essere configurato in modo da avviare il software PC Control e il programma applicativo automaticamente, al termine del caricamento del sistema.

Per ulteriori informazioni su questo argomento fare riferimento a "Configurazione dei programmi per l'esecuzione automatica" nel capitolo 5.

Il software PC Control può lavorare con una alimentazione elettrica non interrompibile?

Windows NT può essere configurato in modo da lavorare con una alimentazione elettrica non interrompibile (UPS = Uninterruptible Power Supply), che può fornire l'alimentazione elettrica al sistema in caso di caduta della linea in CA. In questo caso, la caduta della linea in CA viene notificata a Windows NT da un interrupt al quale Windows NT può rispondere attuando le azioni appropriate.

PC CONTROL può anche utilizzare i segnali di perdita di alimentazione e di batteria bassa ricevuti dall'UPS per eseguire le azioni programmate dall'utente. Per ulteriori informazioni fare riferimento a "Configurazione dell'UPS", a pagina 2-30.

Sezione 2: Progetti e applicazioni

Gestione progetti

La gestione progetti permette di organizzare il lavoro in blocchi logici detti progetti. Ciascun progetto contiene un gruppo file di programma, configurazione e interfaccia operatore, usati per uno scopo comune.

Per editare i programmi applicativi nei formati RLL, SFC, Testo strutturato e Lista di istruzioni si usa il Program Editor, attraverso il quale si eseguono tutte le funzioni di gestione del progetto. Per i progetti valgono le seguenti note:

- Il Program Editor può avere un solo progetto aperto per volta.
- Il nome di un progetto dovrebbe indicare lo scopo comune dei file che lo compongono.
- Il nome del progetto attivo è visualizzato nella barra del titolo della finestra del Program Editor.

Creazione di un nuovo progetto

1. Avviare il program Editor.
2. Cliccare su *Project* nella barra del menu e selezionare *New*. Compare la finestra di dialogo New Project.
3. Digitare il nome del nuovo progetto e cliccare su *OK*.

Nota

Se quando viene eseguita questa procedura esiste un progetto attivo aperto, esso viene chiuso e il nuovo progetto diventa il progetto attivo.

Apertura di un progetto

1. Avviare il Program Editor.
2. Cliccare su *Project* nella barra del menu e selezionare *Open*. Compare la finestra di dialogo Open Project.
3. Cliccare sul progetto che si intende aprire e cliccare su *OK*.

Quando viene aperto un progetto, esso diventa il progetto attivo. Il precedente progetto attivo viene chiuso insieme ai file di programma eventualmente aperti. All'apertura di un progetto, vengono riaperti tutti i file di programma che erano precedentemente aperti nel progetto e viene attivato l'ultimo file di configurazione attivo per quel progetto.

Copia di un progetto

1. Avviare il Program Editor.
2. Cliccare su *Project* nella barra del menu e selezionare *Copy*. Compare la finestra di dialogo Copy Project From.
3. Cliccare sul progetto che si intende copiare e cliccare *OK*. Compare la finestra di dialogo Copy Project To.
4. Digitare il nome della copia e cliccare *OK*.

Quando viene copiato un progetto, tutti i suoi file di programma, interfaccia operatore e configurazione vengono copiati nel nuovo progetto, che diventa il progetto attivo. Dopo la copia, i file di programma aperti nel progetto originale sono aperti anche nel nuovo progetto. Allo stesso modo, il file di configurazione attivo nel progetto originale lo è anche nel nuovo progetto.

Rinominazione di un progetto

1. Avviare il Program Editor.
2. Cliccare su *Project* nella barra del menu e selezionare *Rename*. Compare la finestra di dialogo Rename Project From.
3. Cliccare sul progetto che si vuole rinominare e cliccare su *OK*. Compare la finestra di dialogo Rename Project To.
4. Digitare il nuovo nome del progetto e cliccare su *OK*.

Dopo essere stato rinominato, il progetto diventa il progetto attivo.

Attivazione di una configurazione

1. Selezionare *Project – Activate*. Compare la finestra di dialogo Select Global Configuration File, che contiene l'elenco dei file di configurazione definiti per il progetto corrente.
2. Selezionare la configurazione che si intende attivare e cliccare *OK*.

Programmi applicativi

La funzione Program Management permette di creare e di salvare programmi nei formati RLL (Logica a relè), SFC (Diagramma funzionale sequenziale), ST (Testo strutturato) e IL (Lista di istruzioni). E' possibile aprire molti programmi contemporaneamente. Per passare da un programma aperto ad un altro, usare il menu Window o usare il mouse per cliccare su un programma parzialmente visibile e portarlo in primo piano. I nomi dei programmi attivi sono visualizzati nella barra del titolo del Program Editor.

Creazione di un nuovo programma

Per creare un nuovo programma in una nuova finestra:

1. Cliccare su *File* e selezionare *New*, oppure cliccare sul pulsante nuovo programma della barra degli strumenti dell'editor.



Compare una finestra di dialogo che permette di specificare un nuovo programma RLL, SFC, ST o IL.

2. Selezionare il tipo di programma desiderato e cliccare *OK*. Si aprirà una nuova finestra del tipo appropriato. Al programma viene dato un nome standard (ad esempio, *RLLI*).

Apertura di un programma

Per aprire un programma esistente in una nuova finestra:

Cliccare su *File* e selezionare *Open*, oppure cliccare sul pulsante appropriato della barra degli strumenti dell'editor.



E' possibile aprire molti programmi contemporaneamente. Per passare da un programma aperto ad un altro, usare il menu Window o usare il mouse per cliccare su un programma parzialmente visibile e portarlo in primo piano. Se il comando di apertura viene dato quando non è attivo un progetto, compaiono la finestra di dialogo New Project o la finestra di dialogo Open Project.

Salvataggio di un file

Per salvare il programma attivo nella cartella e con il nome attuali:

Cliccare su *File* e selezionare *Save*, oppure cliccare sul pulsante salva programma della barra degli strumenti dell'editor.



Quando un programma viene salvato per la prima volta, il Program Editor visualizza la finestra di dialogo *Save As* ed è possibile attribuire un nome al programma. Se si desidera cambiare il nome o la cartella del programma attivo prima di salvarlo, usare il comando *Save As*.

Per salvare il file con un nuovo nome

Cliccare su *File* e selezionare *Save As*. Il *Program Editor* visualizza la finestra di dialogo *Save As*, che permette di cambiare il nome del programma.

Stampa di un programma

Per stampare un programma SFC, RLL, ST, o IL:

Cliccare su *File* e selezionare *Print*, oppure cliccare sul pulsante stampa programma della barra degli strumenti dell'editor.



Per cambiare l'impostazione della stampante, cliccare su *File* e selezionare *Print Setup*, oppure premere il pulsante *Setup* nella finestra di dialogo della stampa.

Quando viene stampato un programma SFC, il diagramma SFC viene stampato per primo. Le azioni in esso incluse vengono stampate in ordine alfabetico dopo il diagramma SFC. Infine vengono stampate le eventuali transizioni in logica a relè (transizioni RLL).

Stampa della Cross-Reference del programma

Questa funzione produce una lista dei simboli usati dal programma. Questa lista contiene tutti i simboli di una configurazione, la frequenza con cui sono usati e la loro posizione nel programma. Ad esempio, numero di rung e tipo di bobina o contatto, per un programma RLL.

Per stampare la cross-reference

- Selezionare *Print Xref* nel menu *File*.

Gestione dei programmi applicativi

La gestione programmi permette di creare e di salvare programmi RLL e SFC. E' possibile aprire molti programmi contemporaneamente. I nomi dei programmi attivi compaiono nella barra del titolo del Program Editor.

Per accedere ad un programma aperto nel Program Editor:

1. Cliccare su Window nella barra del menu.
2. Cliccare sul programma desiderato.

Nota

E' anche possibile usare il mouse per cliccare su un programma parzialmente visibile, portandolo così in primo piano.

I comandi eseguiti tramite i menu o i pulsanti della barra degli strumenti sono eseguiti sul programma attivo.

Chiusura dei programmi applicativi

Per chiudere un programma:

Cliccare su *File* nella barra del menu e selezionare *Close*.

oppure

Cliccare sul pulsante *Close* nell'angolo in alto a destra della finestra.

Visualizzazione dei programmi

Quando un programma è aperto, il suo contenuto è visualizzato in una finestra. E' possibile aprire più finestre dello stesso programma RLL, SFC, IL o ST.

Per accedere ad un programma aperto nel Program Editor:

- Selezionare il programma nella lista visualizzata cliccando sul menu *Window*.
- Cliccare su un programma parzialmente visibile, portandolo così in primo piano.

Dimensionamento in scala di un programma per adeguarlo alla finestra

Cliccare su *View* nella barra del menu e selezionare *Scale to Fit Window*.

oppure

Commutare il tipo di visualizzazione cliccando sull'apposito pulsante della barra degli strumenti.



Questa funzione può anche essere usata per visualizzare in scala i programmi RLL che definiscono azioni o transizioni in un programma SFC.

Per navigare con la funzione Scale To Fit Window

1. Visualizzare in scala il programma in modo da adeguarlo alla finestra.
2. Selezionare l'elemento desiderato.
3. Ritornare alla visualizzazione normale. Se quella visualizzata non è l'ultima vista attiva normale, l'elemento selezionato verrà a trovarsi al centro dello schermo.

Attivazione/disattivazione della visualizzazione dei commenti

E' possibile aggiungere dei commenti ad un programma applicativo, per documentarne le funzioni. Il modo Visualizzazione commenti è valido per tutti i programmi aperti.

Per commutare il modo visualizzazione commenti:

Cliccare su *View* nella barra del menu e selezionare *Program Comments*.

oppure

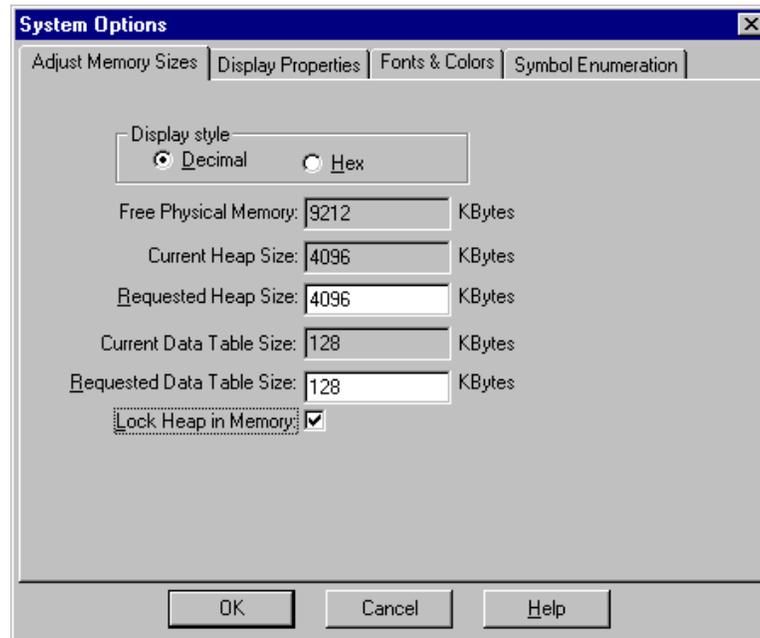
Cliccare sul pulsante appropriato della barra degli strumenti.

Note

Quando i commenti sono visibili, la voce Program Comments del menu View è contrassegnata e il pulsante Visualizza commenti appare premuto.

Impostazione delle opzioni del sistema

La finestra di dialogo System Options permette di impostare vari parametri che influenzano il comportamento e l'aspetto del sistema di comando.



Pagina Adjust Memory Sizes

Campo	Descrizione
Display Style	Visualizza le opzioni relative alla dimensione della memoria in formato decimale o in formato esadecimale.
Free Physical Memory	Memoria fisica libera attualmente disponibile nel computer. Dato non modificabile: dipende dal tipo e dal numero di applicazioni Windows aperte.
Current Heap Size	Impostazione corrente dell'heap, dato non modificabile. L'heap è la memoria allocata (allo start-up) per il sistema di comando. L'heap è utilizzato dai programmi applicativi aperti e dai simboli. Può essere necessario aumentarne la dimensione, tuttavia occorre tenere presente che più grande è l'heap, minore è la quantità di memoria disponibile per le altre applicazioni Windows.
Requested Heap Size	Se è necessario ingrandire l'heap (o se è possibile ridurlo), introdurre la nuova dimensione in questo campo. L'heap può essere più grande della memoria fisica in quanto usa la memoria virtuale fornita dallo swap file. Se durante una sessione di editazione la memoria allocata al sistema di comando si rivela insufficiente, compare un messaggio che suggerisce di aumentare la dimensione dell'heap.
Current Data Table Size	Questa è la memoria usata per i simboli e dipende dal tipo di dati dei simboli. Ad esempio, un array di grandi dimensioni usa molta memoria della tabella dati. Se durante una sessione di editazione la memoria della tabella dati si rivela insufficiente, compare un messaggio che suggerisce di aumentare la dimensione della tabella dati.
Requested Data Table Size	Se necessario, introdurre la nuova dimensione della tabella dati.

Campo	Descrizione
Lock Heap in Memory	Questo campo, se è selezionato, migliora il determinismo del sistema di comando. All'avvio del sistema di comando verrà fatto un tentativo di bloccare l'heap nella RAM fisica; altrimenti il sistema potrà usare la memoria virtuale. Il risultato del tentativo di usare esclusivamente la memoria fisica non è visualizzato; di conseguenza è opportuno avviare il sistema di runtime del sistema di comando prima di qualsiasi altra applicazione, in modo da rendere disponibile per l'heap la massima quantità di memoria fisica possibile.

Pagina Display Properties

Campo	Descrizione
Runtime Tray Icon	Se questo campo è selezionato, l'icona Runtime compare solo nell'area dei messaggi; altrimenti essa compare anche nella barra delle applicazioni. Se il Runtime compare anche nella barra delle applicazioni, è possibile accedervi tramite la tastiera; invece, se compare solo nell'area messaggi, per accedere al Runtime è necessario usare il mouse.
Number of Decimal Places	Questo campo controlla il numero delle posizioni decimali visualizzate nel dettaglio dei blocchi funzionali di un programma RLL. Più posizioni decimali sono visualizzate, maggiore è lo spazio utilizzato.
Number of FB Symbol Characters	Questo campo controlla la lunghezza dello spazio usato per visualizzare i caratteri nel dettaglio dei blocchi funzionali di un programma RLL (non indica necessariamente il numero dei caratteri effettivamente visualizzati. Lo spazio è calcolato considerando la dimensione della X maiuscola. Se i caratteri richiedono meno spazio (come ad esempio la I minuscola) compariranno più caratteri.
Editor Display Update Time	Questo campo controlla il periodo di aggiornamento del Program Editor; ad esempio, ogni quanto tempo vengono aggiornati i valori della finestra di visualizzazione.
Show IEC Style Locations	Se è selezionato questo campo, i punti di I/O sono visualizzati usando la sintassi IEC 1131-3 invece dei loro nomi simbolici.

Pagina Fonts & Colors

Campo	Descrizione
Fonts	Seleziona il tipo di carattere usato dal Program Editor.
Colors	Seleziona vari colori per l'editor.
Reset to Default Colors	Ripristina i colori standard.

Pagina Symbol Enumeration

Il termine enumerazione si riferisce agli elementi di un simbolo complesso. Un simbolo è complesso quando ha degli elementi ai quali è possibile accedere individualmente: strutture, blocchi funzionali, ecc. Le seguenti caselle di controllo funzionano congiuntamente con il comando *Show Symbol Enumerations* del menu *View*. Se una casella è selezionata e il comando *Show Symbol Enumerations* è abilitato, le enumerazioni corrispondenti alla casella sono visibili nella lista dei simboli. Se la casella è deselezionata o il comando *Show Symbol Enumeration* è disabilitato, le enumerazioni non sono visibili nella lista dei simboli.

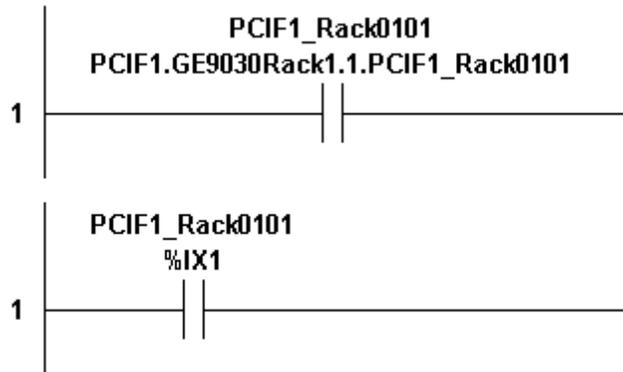
La visualizzazione delle enumerazioni di simboli può influenzare le prestazioni dell'editor (in base al numero di simboli) in quanto sarà necessario caricare più elementi nella lista dei simboli.

Nota: Anche se le enumerazioni sono disabilitate, se inserite un simbolo trascinandolo dal Symbol Manager, compare una lista delle enumerazioni di simboli tra le quali potete selezionare l'enumerazione che desiderate utilizzare.

Campo	Descrizione
Show Function Block Enumerations	Visualizza gli ingressi e le uscite di un simbolo di tipo blocco funzionale.
Show User Type Enumerations	Visualizza gli elementi di un simbolo di tipo definito dall'utente.
Show Axis Enumerations	Visualizza i simboli degli elementi di tipo asse.
Show Axis Group Enumerations	Visualizza i simboli degli elementi di tipo gruppo di assi.
Show System Object Enumerations	Visualizza le variabili associate a temporizzatori (TMR), PID e PRGCB.

Localizzazione dei simboli secondo IEC

La sottostante figura mostra un esempio di localizzazione dei punti di I/O normale e secondo IEC 1131-3 (deve essere abilitato *Display Symbol Locations*). Se un simbolo rappresenta una variabile mappata in memoria, come posizione del simbolo si indica **Memory**.



La sintassi IEC 1131-3 per la localizzazione dei simboli è la seguente:

%	Variabile rappresentata direttamente
I	Punto di ingresso
Q	Punto di uscita
X	BOOL
B	BYTE
W	WORD
D	DWORD

Configurazione dell'UPS

Il sistema di comando può lavorare con una UPS. Esso può ricevere dall'UPS dei segnali in modo da predisporre uno spegnimento controllato. I segnali compaiono nel Symbol Manager come segue:

RT_POWER_FAIL	Rilevazione caduta alimentazione elettrica.
RT_LOW_BATTERY	Rilevazione batteria scarica.

Per usare l'UPS

- Aprire il menu *PC Control Applications* dal menu Avvio di Windows e scegliere *UPS Monitor*.

Nell'area dei messaggi compare l'icona UPS Monitor . Per configurare l'UPS fare doppio click sull'icona. Comparirà la finestra di dialogo *PC Control UPS Monitor*.



Campo	Descrizione
UPS is installed on:	Selezionare la porta seriale alla quale è collegata l'UPS.
UPS Configuration	Per specificare questi dati fare riferimento alla documentazione dell'UPS.
Power failure signal UPS Interface Voltage	Selezionare questa opzione se l'UPS fornisce un segnale in caso di caduta dell'alimentazione. In questo caso selezionare anche la polarità della tensione.
Low battery signal UPS Interface Voltage	Selezionare questa opzione se l'UPS fornisce un segnale di batteria scarica. In questo caso selezionare anche la polarità della tensione.
UPS Characteristics	Se l'UPS non fornisce un segnale di batteria scarica, si possono specificare le sue caratteristiche facendo riferimento alla documentazione ad essa relativa.
UPS Service	Queste preferenze possono essere specificate in base alle necessità del sistema..

Per arrestare l'UPS

- Cliccare con il pulsante destro del mouse sull'icona dell'UPS e scegliere *Shutdown Memory UPS*.

Salvataggio e ripristino delle variabili

Quando si verifica l'evento corrispondente, il monitor dell'UPS setta le variabili RT_POWER_FAIL o RT_LOW_BATTERY. E' possibile includere nel programma applicativo la logica necessaria per salvare sul disco fisso i dati che interessano e successivamente leggere quei dati allo start-up per ripristinarli

Esempio di logica per il salvataggio delle variabili:

```
IF RT_LOW_BATTERY THEN

    savedata.int1:= ValoreImportante;

    (* ripetere per tutti i dati da salvare *)

    OPENFILE(FCBVar, FILE:= "Backup.dat");

    WRITEFILE(FCBVar, IN:= savedata );

    CLOSEFILE(FCBVar);

END_IF;
```

Sezione 3: Descrizione generale del funzionamento del programma

Attivazione di una configurazione

All'attivazione di una configurazione il sistema esegue le seguenti azioni:

- Inizializza la memoria globale.
- Se le uscite sono configurate come disabilitate, abilita l'interfaccia, mette le uscite nello stato disabilitato (force non avrà effetti) e inizia la scansione.
- Se le uscite sono abilitate, stabilisce lo stato standard e inizia la scansione.

Prima scansione con la configurazione attiva

Alla prima scansione si verifica quanto segue:

- Vengono inizializzate le variabili dei blocchi funzionali e le variabili locali.
- Per la prima scansione del primo programma eseguito, RT_FIRST_SCAN è alto.
- Viene risolta la logica.
- Vengono aggiornati gli I/O e la sequenza si ripete.

Sequenza di spegnimento

All'ultima scansione si verifica quanto segue:

- Il programma non riceve alcuna notifica anticipata di un prossimo spegnimento e non può eseguire alcuna azione specifica.
- I moduli di interfaccia stabiliscono lo stato disabilitato degli I/O in base alle loro specifiche caratteristiche.

Funzionamento normale

Durante il funzionamento normale si verifica quanto segue:

- Tutte le azioni associate a un passo vengono risolte ad ogni scansione nella quale il passo è attivo (però dipende dal qualificatore dell'azione).
- La soluzione delle azioni precede quella del testo strutturato contenuto nel passo.
- Se una transizione associata ad un passo diventa TRUE (vera), vengono riaggiornati gli I/O, dopo di che vengono risolte tutte le azioni appropriate (non impulsive, ritardate o limitate (inattive)), mettendo nello stato off tutte le bobine di uscita salvo quelle ritenute (latched). I blocchi funzionali non sono risolti. Gli I/O vengono aggiornati e i passi attivi successivi vengono risolti nella scansione successiva.
- Se un passo contiene testo strutturato, la transizione non è valutata finché tutte le frasi in testo strutturato non sono state completate.
- Tutte le frasi in testo strutturato all'interno di un ciclo FOR sono eseguite a una iterazione per scansione, a meno che non venga incontrata una frase NOWAIT, che causa l'esecuzione di tutte le iterazioni in una sola scansione.
- Tutte le frasi in testo strutturato contenute in un passo sono eseguite una volta nella prima scansione durante la quale il passo è attivo e non sono ripetute nelle scansioni successive.
- Tutte le frasi contenute in un ciclo WHILE continuano ad essere eseguite finché la condizione di WHILE non diventa FALSE..
- La normale sequenza di esecuzione è: aggiornamento degli I/O, soluzione della logica, poi ripetizione.

Nomi dei file

La sottostante tabella descrive i tipi di file contenuti nella cartella di un progetto.

_gprog.bbn	Configurazione globale in formato binario
_gprog.st	Configurazione globale in testo strutturato.
CFG	File di configurazione
CSV	File di variabili separato da virgole
DAT	Parametri di ambiente per i file di programma
IL	Estensione dei programmi IL (lista di istruzioni)
OPI	File dell'interfaccia operatore
R**	Estensione dei programmi RLL (linguaggio a contatti)
S**	Estensione dei programmi SFC (diagramma funzionale sequenziale)
SBN	File binario eseguito dal motore di runtime
SFR	Informazioni di "redo" (ripeti) memorizzate
SFU	Informazioni di "undo" (annulla) memorizzate
SFX	Informazioni per la cancellazione delle modifiche online
SNF	File nomi condivisi
SST	Versione in testo strutturato di un programma SFC
ST	Estensione dei programmi ST (testo strutturato)
SWC	Nomi dei simboli aggiunti alla finestra di controllo

Nota

I file con le estensioni RLL, OPI, SFC, CFG, ST e IL contengono i programmi sorgente. Per trasferire un'applicazione è sufficiente copiare questi file.

Capitolo 3

Configurazione degli I/O

Il passo successivo alla creazione di un progetto è, normalmente, la creazione della configurazione del sistema (System Configuration) per quel progetto. La System Configuration contiene una collezione di informazioni relative all'hardware insieme alle variabili globali usate in tutti i programmi di controllo del progetto.

Idealmente, prima di configurare gli I/O del sistema, si installeranno le schede di interfaccia I/O e gli I/O stessi. E' tuttavia possibile configurare il sistema prima di installare questo hardware. Per farlo, cliccare sul pulsante Simulate nella finestra di dialogo a livello scheda per le schede di interfaccia I/O utilizzate.

Le informazioni relative all'installazione e alla configurazione delle schede di interfaccia si trovano nelle seguenti appendici:

Sistema I/O	Scheda	Numero di catalogo	Appendice
Genius	PCIM	IC660ELB921 (una porta) IC660ELB922 (due porte)	Appendice A
Serie 90-30	PCIF1	IC693PIF301	Appendice B
Serie 90-30	PCIF2	IC693PIF400	Appendice C

Questo capitolo fornisce le seguenti informazioni:

- Descrizione generale della configurazione del sistema I/O
- Configurazione I/O Genius
- Configurazione I/O Serie 90-30
- Configurazione di altri bus di campo
- Scambio dinamico dei dati (DDE = Dynamic Data Exchange)
- Importazione/esportazione della configurazione

Sezione 1: Descrizione generale

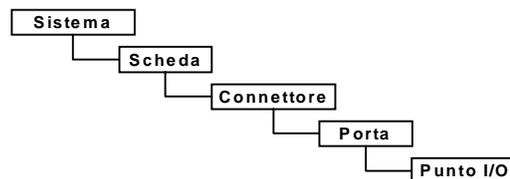
La finestra di dialogo System Configuration contiene una collezione di informazioni relative all'hardware insieme alle variabili globali usate in tutti i programmi di controllo di un dato progetto.

La configurazione dice al software quali schede di interfaccia sono installate sul computer e quali I/O sono collegati a quelle schede. Mediante l'editor della configurazione è possibile assegnare nomi simbolici ai punti di I/O e inserire commenti relativi alla funzione di particolari elementi.

L'editor della configurazione vi conduce passo passo lungo tutto il processo di configurazione. Inizierete attribuendo un nome al sistema (il computer) che state utilizzando e definendo il tempo di scansione del programma. Seguiranno gli slot del PC e le schede del sistema di comando (non le schede base del computer) installate in ciascuno slot, poi sarà la volta dei moduli I/O connessi alla scheda e finalmente dei punti di I/O.

Se la scheda è una scheda per le comunicazioni o un I/O Scanner, la configurazione va ancor più in profondità per descrivere rack, slot, schede, ecc. remoti.

La sottostante figura presenta i livelli di configurazione per una semplice scheda I/O.



Conflitti hardware

Per comunicare con la maggior parte delle schede di interfaccia è necessario effettuare un certo numero di impostazioni. Le più comuni sono: indirizzo della porta, indirizzo base di memoria, interrupt. Perché la scheda di interfaccia funzioni correttamente occorre che le impostazioni tramite ponticelli fatte sulla scheda corrispondano a quelle specificate nella finestra di dialogo Define Board **E CHE** non esistano conflitti con altro hardware del PC.

IMPORTANTE:

Il modo migliore di determinare quali risorse del sistema siano assorbite dai dispositivi hardware consiste nel visualizzare la diagnostica NT nel gruppo programmi degli strumenti amministrativi. La pagina Risorse contiene la maggior parte delle risorse hardware usate dal sistema. Tuttavia, alcuni dispositivi non comunicano alla diagnostica NT le risorse che consumano e i dispositivi al momento non utilizzati non sono indicati, ma POSSONO ugualmente causare conflitti. La diagnostica NT è semplicemente un buon punto di partenza.

Per ulteriori informazioni sulla prevenzione dei conflitti fare riferimento a:

Sistema I/O	Scheda	Numero di catalogo	Appendice
Genius	PCIM	IC660ELB921 (una porta) IC660ELB922 (due porte)	Appendice A
Serie 90-30	PCIF1	IC693PIF301	Appendice B
Serie 90-30	PCIF2	IC693PIF400	Appendice C

Porte I/O

Per comunicare con la scheda, il device driver usa frequentemente una porta di I/O dei dati.

Gli indirizzi di alcune porte sono standard su molti PC. La lista che segue identifica le porte più comunemente usate:

COM1: 3F8-3FF
 COM2: 2F8-2FF
 COM3: 3E8-3EF
 COM4: 2E8-2EF
 LPT1: 378-37A
 Floppy: 3F0-3F7
 Video: 3B0-3BB e 3C0-3DF

Indirizzo di memoria

L'indirizzo di memoria si usa per predisporre un'area della memoria condivisa per la scheda di interfaccia. Normalmente è disponibile la memoria da C8000 a DFFFF. Quest'area di memoria è normalmente usata per le schede relative a funzioni speciali. Ad esempio, se la scheda usa 4000 (esadecimale) byte di memoria condivisa a partire dall'indirizzo D0000, essa userà la memoria da D0000 a D3FFF.

Interrupt – IRQ

Per comunicare con la scheda si usa l'interrupt. I valori validi di interrupt vanno da 0 a 11.

Alcuni livelli di interrupt sono standard per la maggior parte dei PC e non devono essere usati. Una situazione comune è la seguente:

COM1 e COM3: IRQ4
 COM2 e COM4: IRQ3
 Floppy: IRQ6
 LPT1: IRQ7

IRQ1 e 2 sono sempre utilizzati dall'hardware del sistema.

I file di configurazione

Elementi di una configurazione

Una configurazione vi permette di definire la struttura degli I/O e di assegnare i nomi di identificazione ai punti e alle porte I/O. Tutte le configurazioni hanno come nome del file *.cfg.

Tutte le operazioni relative alla configurazione si eseguono all'interno del Program Editor.

Creazione di una nuova configurazione

1. Avviare il Program Editor.
2. Cliccare su *File* e selezionare *New Config*. Comparirà la finestra di dialogo per la configurazione.



Editazione di un file di configurazione esistente

1. Avviare il Program Editor.
2. Cliccare su *File* e selezionare *Open Config*. Comparirà la finestra di dialogo per la configurazione.
3. Editare i campi necessari e cliccare su *Close* e *Save*.

Attivazione di un file di configurazione

Per attivare un nuovo file di configurazione per il progetto attivo, cliccare su *Project* e selezionare *Activate Config*. Il nome del file di configurazione attivo compare nella barra di stato inferiore del Program Editor.

Il file di configurazione attivo è usato dall'ambiente di runtime per determinare la struttura dei sistemi di I/O e i nomi dei simboli globali. Lo stesso file è usato dal Program Editor per mettere a disposizione del programmatore i nomi simbolici degli ingressi, delle uscite e delle variabili residenti in memoria, al livello di sistema.

Quando viene attivato un nuovo file di configurazione è possibile che i programmi attivi non siano più in grado di funzionare, se la configurazione precedentemente attiva conteneva informazioni non disponibili nel nuovo file. L'attivazione di un file di configurazione è registrata nel progetto attivo..

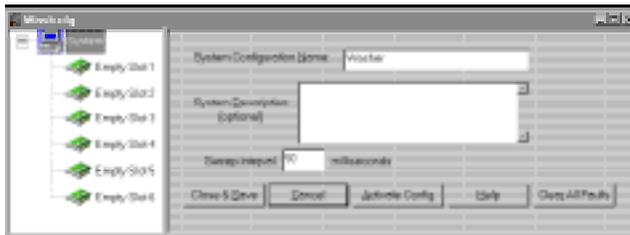
Ogni volta che viene aperto un progetto, viene attivato il file di configurazione selezionato per quel progetto.

Navigazione all'interno dell'utilità di configurazione

La finestra di dialogo System Configuration

Mediante la finestra di dialogo System Configuration è possibile attribuire alla configurazione un nome di sistema. Il nome del file deve essere un nome DOS valido di otto caratteri. Non sono ammessi spazi. Nell'esempio che segue il nome di sistema è "Washer".

La cartella standard dei file di configurazione è la cartella del progetto corrente.



Intervallo di scansione degli I/O

L'intervallo di scansione degli I/O si specifica nella finestra di dialogo System Configuration. Questo è l'intervallo con il quale i programmi di controllo aggiornano gli I/O ed eseguono la logica programmata. Le applicazioni Windows a più bassa priorità sono interrotte dalla scansione degli I/O. I programmi di scansione degli I/O sono eseguiti dall'inizio alla fine senza interruzioni.

Prima della soluzione della logica programmata, l'I/O Scanner legge gli ingressi. Dopo l'esecuzione della logica programmata vengono scritte le uscite. Il tempo di esecuzione della logica programmata è variabile, essendo variabile il numero e il tipo delle istruzioni attive in ciascun intervallo di scansione degli I/O. Terminato l'aggiornamento delle uscite, l'applicazione di controllo viene sospesa e riprende l'esecuzione delle altre applicazioni NT a più bassa priorità.

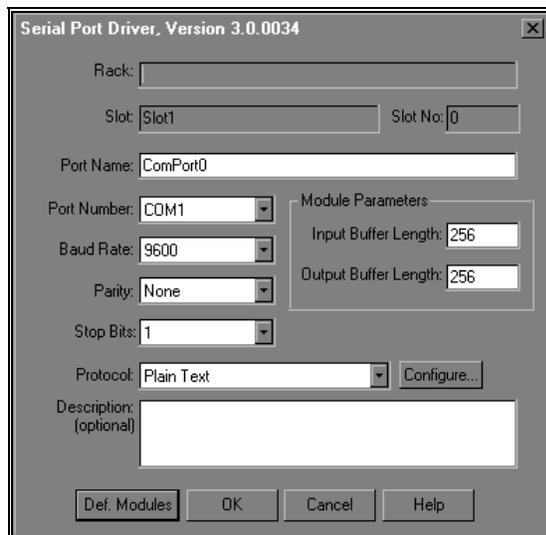
Nel definire l'intervallo di scansione considerate:

- Ampiezza minima di un impulso in ingresso – il tempo minimo per il quale un ingresso deve mantenere un certo stato per essere riconosciuto dal software.
- Tempo di risposta minimo – il tempo minimo entro il quale il sistema di comando genera un cambio di stato di un'uscita in risposta a un cambio di stato di un ingresso.
- Tempo di risposta massimo - il tempo massimo entro il quale il sistema di comando genera un cambio di stato di un'uscita in risposta a un cambio di stato di un ingresso.

La finestra di dialogo Define Board

Esistono piccole differenze tra le finestre di dialogo per la configurazione dei punti di I/O dei vari tipi di schede di interfaccia. Per avere informazioni più specifiche riguardo alla configurazione dei punti di I/O, fare riferimento al file di aiuto del driver della scheda utilizzata.

1. Nella finestra di dialogo per la configurazione, cliccare su Define Board.



2. Impostare appropriatamente gli attributi della scheda e cliccare su *OK*.

Sezione 2: Configurazione di GENIUS I/O

Il modulo interfaccia PC (PCIM = PC Interface Module)

La connessione fra un PC e un sistema Genius I/O si realizza tramite una scheda PCIM (PC Interface Module). Esistono due tipi di schede PCIM: con una o con due porte. Ciascuna porta di una scheda PCIM è utilizzata per comunicare su un dato bus Genius I/O. Un bus Genius I/O può supportare fino a 30 blocchi di I/O.

Per definire/configurare l'hardware della scheda PCIM e i blocchi di I/O presenti sul o sui bus Genius I/O ad essa connessi si usa la finestra di dialogo PCIM Board Definition.

Configurazione del PCIM

1. Se il PCIM non è ancora stato configurato (mediante la PCIM Configuration Utility (PCU), cliccare sul pulsante Run PCIM Configuration Utility. In questo modo si avvia un'utilità separata che permette di specificare vari parametri di configurazione del PCIM. L'aiuto in linea di questa utilità fornisce le istruzioni per l'installazione e la configurazione del PCIM.
2. Dopo aver configurato il PCIM ed essere usciti dalla PCU, cliccare sul pulsante Read PCIM Params From Registry. Questo apre una finestra contenente una lista delle schede PCIM configurate.
3. Nella lista dei numeri di modello (Model No), selezionare la scheda PCIM appropriata.
4. Per testare la configurazione del PCIM cliccare sul pulsante Test PCIM Configuration. Devono accendersi entrambe le spie della scheda CPIM.

Finestra di dialogo PCIM Board Definition

Campo Name

Il campo Name si usa per specificare il nome della scheda PCIM. Questo nome serve per descrivere la scheda nella localizzazione dei punti di I/O nel resto del software.

Campo Model No

Esistono due modelli validi: con una o con due porte (Single o Dual Port). Il modello con una porta supporta un solo bus Genius I/O, mentre il modello con due porte supporta due bus. Il dialogo di definizione della scheda permette di specificare gli I/O e la configurazione hardware per la porta 1 su una scheda con una porta. Su una scheda con due porte è possibile configurare la porta 1 e la porta 2. Se il tipo di scheda è NULL, non è possibile definire gli I/O.

Campo **Board OK Bit**

Se in questo campo viene introdotto un nome, viene creata una variabile booleana con quel nome. Questa variabile è TRUE quando nessuno dei moduli della scheda è guasto. Se uno o più moduli sono guasti, questa variabile diventa FALSE.

Campo **Cfg Mismatch**

Se in questo campo viene introdotto un nome, viene creata una variabile booleana con quel nome. Questa variabile è TRUE se i moduli effettivamente collegati al bus non corrispondono ai blocchi specificati nella configurazione.

Campo **Description**

Questo campo permette di specificare una descrizione in testo libero della scheda. E' un campo facoltativo.

Casella di controllo **Disable outputs when no programs are running**

Se questa casella è selezionata, quando non ci sono programmi in esecuzione tutte le uscite sono a 0.

Porta 1/Porta 2

Casella di controllo **Simulate**

La casella Simulate causa la disabilitazione della porta. Le informazioni relative ai guasti e alla configurazione saranno ignorate. Questo è utile quando gli I/O collegati alla porta non sono utilizzati. Se l'I/O non è simulato, qualsiasi indicazione di errore generata dai blocchi di Genius I/O indurrà lo I/O Scanner a fermare in emergenza i programmi attivi.

Campo **Network Number**

Deve essere selezionato un numero arbitrario di rete compreso fra 1 e 9. Il numero di rete serve per collegare il bus fisico di I/O desiderato alla porta logica definita mediante questo dialogo.

Avvertenza

Il driver PCIM I/O non può supportare più di 9 bus I/O. I numeri di rete specificati in una configurazione devono essere unici.

Campo **Interrupt Number**

Il driver PCIM I/O usa l'interrupt specificato in questo campo per gestire la segnalazione degli errori e la notificazione delle variazioni di configurazione per lo scanner degli I/O. Il valore specificato qui deve corrispondere al valore specificato per la scheda.

Campo **Shared RAM Address**

Questo campo specifica la posizione del buffer di 16K usato per comunicare con la porta specificata della scheda PCIM. I valori possibili di questo campo vengono presentati per la selezione. Ciascuna posizione specificata in una configurazione deve essere unica, lo stesso buffer non può essere usato da due porte distinte.

L'intero address space della RAM condivisa non deve essere usato da nessun altro dispositivo del PC. Possono verificarsi conflitti che impedirebbero alla scheda PCIM di funzionare correttamente. Il valore specificato qui deve corrispondere al valore specificato con l'impostazione dei microinterruttori della scheda PCIM a una porta o nel programma di configurazione per la scheda a due porte.

Nota

L'indirizzo della RAM condivisa si imposta con l'utilità di configurazione PCIM (PCU). Qui è semplicemente visualizzato. Eventuali modifiche devono essere fatte ricorrendo alla PCU.

Campo **I/O Port Address**

Questo campo specifica la posizione iniziale dei quattro byte usati per configurare e controllare la porta. I secondi due byte sono quelli specificati con i microinterruttori sulla scheda a due porte (ad esempio, un valore di 0x222 sarebbe mappato a 0x220).

L'indirizzo della porta I/O deve essere uguale al valore specificato con i microinterruttori per la scheda a una porta e deve essere 2 meno del valore specificato con i microinterruttori per la scheda a due porte.

Campo **PCIM Bus Address**

Questo campo specifica l'indirizzo del bus seriale per questa porta.

Pulsante Test Port Configuration

Questo pulsante si usa per convalidare i valori specificati i valori introdotti nei campi relativi al numero di interrupt, all'indirizzo della RAM condivisa e all'indirizzo della porta I/O, tentando di avviare la scheda PCIM. Questa operazione ha significato solo se è installata la scheda..

Pulsante Auto Configure Network Devices

Questo pulsante avvia la scansione del bus Genius I/O da parte del driver PCIM, per la rilevazione dei dispositivi in linea. I dispositivi rilevati sono inseriti automaticamente nella configurazione.

Pulsante Setup Global Data

Questo pulsante permette di impostare i dati globali per la porta. Esso apre la finestra di dialogo Global Data Setup.

Pulsante Define PCIM I/O

Questo pulsante apre la finestra di dialogo Genius Bus Address Definitions, che si usa per specificare i dispositivi collegati al bus Genius I/O.

Pulsante OK

Il pulsante OK salva tutte le modifiche eseguite e mette fine alla sessione di specifica della configurazione, ritornando al menu principale dell'utilità di configurazione.

Pulsante Cancel

Questo pulsante annulla le modifiche eseguite e mette fine all'INTERA sessione di specifica della configurazione, ritornando al menu principale dell'utilità di configurazione.

Finestra di dialogo **Global Data Setup**

Sul bus Genius, ciascuna unità può trasmettere fino a 128 byte di dati ad ogni scansione e può ricevere 128 byte di dati da ciascuna delle altre unità collegate allo stesso bus. In questa finestra di dialogo, selezionare la casella di controllo “Enable Global Data” per ciascuna delle unità dalle quali si intende ricevere dati globali. Se il PCIM dovrà trasmettere dei dati globali ad altre unità, selezionare la casella “Enable Global Data” per l’unità corrispondente all’indirizzo di bus del PCIM.

Pulsante **Setup Global Data**

Premere questo pulsante per predisporre le etichette di I/O da associare ai dati globali per questo indirizzo di bus. Comparirà la finestra di dialogo “Global Data: Device” relativa all’unità selezionata.

Finestra di dialogo **Global Data: Device**

Questa finestra di dialogo permette di etichettare i dati globali che possono essere trasmessi e ricevuti attraverso il PCIM. E’ possibile etichettare ciascuna parola entro i 128 byte che costituiscono l’area dei dati globali e ciascun bit di ciascuna di queste parole. Inizialmente compaiono sullo schermo i nomi attuali delle prime 16 parole. Per passare alle altre parole, usare i pulsanti “Next 16 Words” e “Previous 16 Words”. A fianco dei nomi di etichetta si trova una casella di controllo. Per etichettare i singoli bit delle parole selezionare la casella di controllo “Name Bits” accanto alla parola della quale si intendono etichettare i bit, poi cliccare sul pulsante “Edit Bit Names”. Comparirà la finestra di dialogo “Global Data: Device, Word”.

Finestra di dialogo **Global Data: Device, Word**

Questa finestra di dialogo permette di etichettare i singoli bit della parola selezionata..

Definizione degli indirizzi sul bus Genius

La finestra di dialogo “Genius Bus Address Definition” si usa per specificare e modificare i blocchi di I/O definiti su un bus Genius I/O. Ciascun blocco di I/O è definito ad un certo indirizzo del bus. Gli indirizzi validi vanno da 0 a 31. I blocchi definiti sono elencati nella regione di scorrimento.

Finestra di dialogo **Genius Bus Address Definitions**

Enable Device Configuration Table for this port

Abilita la tabella di configurazione.

Enable Datagrams for this port

Abilita i datagrammi per questa porta. Per ulteriori informazioni fare riferimento al paragrafo “Trasmissione dei datagrammi con il driver PCIM”.

Bus

Visualizza l'identificatore del bus (ID).

Block Type

Questo campo visualizza il tipo del blocco di I/O definito. (Il tipo di blocco standard per l'ID 31 è PCIM e non può essere modificato.)

Block Name

Questo campo visualizza il nome del blocco di I/O..

Select I/O Device

Questo pulsante apre la finestra di dialogo “Block Information”, che permette di cambiare il tipo del blocco di I/O all'indirizzo selezionato. Per rimuovere dal bus un blocco di I/O , selezionare il blocco, premere questo pulsante e selezionare NULL come nuovo tipo del blocco.

Pulsante Define Ports

Questo pulsante apre la finestra di dialogo associata al blocco di I/O selezionato. In questa finestra è possibile editare i punti di I/O del blocco specificato. Fare doppio click su un blocco è come selezionare quel blocco e cliccare su questo pulsante.

Pulsante Back to Board

Cliccare su questo pulsante per ritornare alla finestra di dialogo per la definizione della scheda PCIM.

Pulsante OK

Salva le modifiche eseguite e mette fine all'intera sessione di configurazione, ritornando al menu principale dell'utilità di configurazione.

Pulsante **Cancel**

Visualizza la finestra di dialogo “Confirm Abort”. Se si risponde sì, la sessione di edit termina senza salvare le modifiche

Finestra di dialogo **Discrete Point Information**

Il dialogo “Discrete Point Information” si usa per definire i punti di I/O associati a un blocco di I/O Genius.

Campo **Status Bit Name**

Se in questo campo viene inserito un nome, viene creata una variabile booleana con quel nome. Se il blocco funziona correttamente questa variabile è TRUE. Se il blocco è guasto, la variabile è FALSE.

Campi **Whole Block Input/Whole Block Output**

Questi campi possono comparire entrambi o può comparirne uno soltanto. Se viene inserito un nome, viene creata una variabile del tipo e della dimensione appropriati a coprire l'intero blocco. Ad esempio, se il blocco è un ingresso discreto a 16 bit, viene creata una variabile WORD di ingresso. Per i moduli con punti a direzione configurabile, leggere solo i bit del simbolo “Whole Block Input” corrispondenti agli ingressi e scrivere solo i bit del simbolo “ Whole Block Output ” corrispondenti alle uscite.

Bit

Visualizza i numeri dei bit del blocco di I/O. I valori vanno da 1 a 16. Questi valori non possono essere modificati. Essi servono solo da riferimento.

Campo **Name**

Questo campo contiene il nome specificato dall'utente per il punto di I/O. Il nome deve rispettare le convenzioni di denominazione definite dallo standard IEC-1131. Il nome deve iniziare con una lettera e può contenere i caratteri alfanumerici e il carattere di sottolineatura. Il nome di un punto di I/O non può contenere spazi. Inoltre, i nomi fanno differenza fra maiuscole e minuscole.

Pulsante **Direction**

Se il tipo di blocco ammette la configurazione della direzione, un click su questo pulsante ne causa la commutazione fra “Input” e “Output”.

Pulsante **Back to Bus**

Un click su questo pulsante causa il salvataggio delle modifiche eseguite e il ritorno al dialogo “Genius Bus Address Definition”.

Finestra di dialogo **Analog Point Information**

Questa finestra di dialogo si usa per definire i punti di I/O associati al blocco di I/O Genius.

Campo **Status Bit Name**

Se in questo campo viene inserito un nome, viene creata una variabile booleana con quel nome. Se il blocco funziona correttamente questa variabile è TRUE. Se il blocco è guasto, la variabile è FALSE.

Campo **Name**

Questo campo contiene il nome specificato dall’utente per il punto di I/O. Il nome deve rispettare le convenzioni di denominazione definite dallo standard IEC-1131. Il nome deve iniziare con una lettera e può contenere i caratteri alfanumerici e il carattere di sottolineatura. Il nome di un punto di I/O non può contenere spazi. Inoltre, i nomi fanno differenza fra maiuscole e minuscole.

Pulsante **Direction**

Questo pulsante ha solo funzione informativa. La direzione dei punti analogici è determinata dal tipo di blocco I/O.

Pulsante **Back to Bus**

Un click su questo pulsante causa il salvataggio delle modifiche eseguite e il ritorno al dialogo “Genius Bus Address Definition”.

Trasmissione dei datagrammi con il driver PCIM

Il dialogo “Bus Address Definition” contiene una casella di controllo denominata “Enable Datagrams for this port”.

Se si seleziona questa casella, nel file .cfg vengono creati 11 simboli. Essi avranno tutti il seguente prefisso:

<Nome della scheda>_Network<Numero network>_

Questi simboli corrispondono al blocco di comando della scheda μ Geni della scheda PCIM, descritti nel capitolo 10 *μ Geni Board User's Manual*, GFK-0845. I soli comandi da emettere per questa via sono Trasmissione datagramma e Trasmissione datagramma con risposta.

Perché la variabile DgramStatus venga messa a 1 devono essere riempiti tutti i campi. A questo punto DGramStatus si comporta come è descritto nel *μ Geni Board User's Manual*. Per le informazioni relative al formato dei dati dei datagrammi, si rinvia al *Genius I/O System and Communications User's Manual*, GEK-908486-1.

Simbolo	Descrizione
DgramStatus	Stato
DgramCmdType	Tipo di comando (02=TrasmissioneDatagramma, 03=Trasmissione con risposta)
DgramDestAddr	Indirizzo bus di destinazione
DGramFunc	Codice funzione
DgramSubFunc	Codice sottofunzione
DgramHdrByte5	Vedere la sottostante descrizione.
DgramHdrByte6	Vedere la sottostante descrizione.
DgramHdrByte7	Vedere la sottostante descrizione.
DgramHdrByte8	Vedere la sottostante descrizione.
DgramData	Array di BYTE (Dati in uscita)
DgramReplyData	Array di BYTE (Dati della risposta)

Quando viene dato un comando Trasmissione datagramma, HdrByte5 è la priorità e HdrByte6 è la lunghezza.

Quando viene dato un comando Trasmissione datagramma con risposta, HdrByte5 è il codice di sottofunzione della risposta, HdrByte6 è la priorità, HdrByte7 è la lunghezza del buffer per la trasmissione e HdrByte8 è la lunghezza del buffer per la risposta (impostata dalla scheda μ Geni).

Sezione 3: Configurazione degli I/O della serie 90-30

Il software PC Control supporta le seguenti schede I/O della serie 90-30:

Designazione in PC Control	Numero di catalogo	Installazione
PCIF1	IC693PIF301	Appendice B
PCIF2	IC693PIF400	Appendice C

L'interfaccia PC (PCIF = PC Interface)

Per stabilire una connessione fra un PC e un sistema I/O della serie 90-30 si utilizza una scheda PCIF. Per definire la configurazione dell'hardware della scheda PCIF e degli I/O della serie 90/30 si usa la finestra di dialogo "PCIF Board Definition".

Finestra di dialogo PCIF Board Definition

Campo Board Name

Inserire il nome della scheda. E' possibile conservare il nome standard o inserirne uno diverso. Questo nome è visualizzato nel Symbol Manager come aiuto nella identificazione della posizione dei punti di I/O.

Campi Port Address e Board Number

Usare questi campi per specificare l'indirizzo I/O base e il numero di scheda della scheda PCIF nel PC. L'indirizzo base deve corrispondere all'impostazione dei microinterruttori della scheda. Il numero di scheda di ciascuna PCIF deve essere unico. La numerazione parte da 0. Per ulteriori informazioni sull'installazione della scheda PCIF e sulla selezione di un indirizzo I/O base appropriato, fare riferimento all'appendice B (PCIF1) o all'appendice C (PCIF2), o cliccare sul pulsante aiuto a fianco del campo "Port Address".

Casella di controllo Simulate

Selezionare questa casella per eseguire i programmi applicativi senza accedere al sistema I/O. Questo è utile per l'esecuzione dei test e per lo sviluppo quando la scheda PCIF o i rack di I/O non sono disponibili.

Casella di controllo **Disable Outputs When No Programs are Running**

Selezionare questa casella per azzerare tutte le uscite quando non vi sono programmi in esecuzione. Questo può essere un importante fattore di sicurezza. Deselezionare questa casella se si intendono lasciare le uscite dopo il completamento dei programmi o se si vuole poter forzare l'attivazione di un'uscita senza programmi in esecuzione.

Casella di controllo **Abort Program Upon Error**

Se questa casella è selezionata, gli eventuali errori saranno visualizzati e i bit di stato saranno impostati correttamente, ma i programmi in esecuzione non saranno interrotti. Se questa casella non è selezionata, i programmi in esecuzione verranno interrotti in caso di errore.

Campo **Cfg Mismatch Bit Name**

Se in questo campo viene inserito un nome, viene creata una variabile booleana con quel nome. Questa variabile sarà TRUE se la configurazione specificata non corrisponde all'hardware effettivamente installato.

Campo **Status Bit Name**

Se in questo campo viene inserito un nome, viene creata una variabile booleana con quel nome. Questa variabile sarà TRUE se tutti i moduli della scheda funzionano correttamente e sarà FALSE in caso contrario.

Campo **Description: (facoltativo)**

Questo spazio è reso disponibile per una descrizione facoltativa della scheda I/O.

Pulsanti **Racks**

Questi pulsanti aprono i dialoghi di configurazione dei moduli I/O montati in ciascun rack. PCIF1 supporta quattro rack; PCIF2 supporta sette rack. I pulsanti sono grigi finché non è definito ciascun rack.

Pulsante **Define 90-30 Racks**

Questo pulsante apre una finestra di dialogo per la definizione dei rack I/O connessi alla scheda PCIF.

Pulsante **Auto Configure**

Questo pulsante attiva la funzione di configurazione automatica. Questa funzione può essere usata quando i rack e i moduli I/O sono montati e accesi e il sottosistema Runtime non è attivo. Quando si clicca su questo pulsante, il software attiva la scheda PCIF e legge le informazioni relative ai rack e ai moduli montati.

Pulsante **Set and Test Board**

Questo pulsante comunica con la scheda PCIF per verificare se l'installazione è corretta (Il sottosistema Runtime non deve essere attivo.) Dovrebbe apparire una finestra contenente il messaggio "PCIF2 Board found, shared RAM access successfully found". Se questo messaggio non compare, tentare di configurare un diverso blocco di RAM condivisa.

Finestra di dialogo **Rack Definition**

Questo dialogo permette di configurare manualmente il rack connesso alla scheda PCIF. Introdurre un nome per il rack e selezionare il tipo di rack.

I pulsanti relativi ai moduli aprono i dialoghi di definizione dei moduli per il rack associato.

Finestra di dialogo **Module**

Campo **Module Name**

Introdurre un nome descrittivo di questo modulo. Questo nome viene visualizzato dal Symbol Manager come aiuto nella localizzazione di un particolare simbolo I/O.

Campo **Module Position**

Questo è il numero dello slot fisico del rack. Usare i pulsanti "Next Module" e "Prev. Module" per passare da un modulo all'altro dello stesso rack.

Campo **Module ID Code**

Introdurre il codice del modulo montato in questo slot. Premere il pulsante "Lookup" per visualizzare i numeri di modello dei moduli nel campo descrizione.

Pulsante **Ports**

Questo pulsante apre un dialogo per la definizione delle singole porte di questo modulo.

Definizione delle connessioni delle porte digitali

Introdurre il nome simbolico di ciascun bit di I/O utilizzato. Usare i pulsanti “Next Conn” e “Prev Conn” per passare da un gruppo di I/O all’altro di questo modulo.

La finestra nella parte superiore del dialogo identifica la direzione (Input o Output).

Campo **Module Health Bit**

Se in questo campo viene specificato un nome, viene creata una variabile booleana. Questa variabile sarà True se il modulo funziona correttamente e diventerà FALSE se il modulo è guasto.

Definizione delle connessioni delle porte analogiche

Introdurre un nome descrittivo per ciascuna delle porte analogiche listate. Ciascun nome deve essere unico nell’intero database dei simboli globali.

Campo **Module Health Bit**

Se in questo campo viene specificato un nome, viene creata una variabile booleana. Questa variabile sarà True se il modulo funziona correttamente e diventerà FALSE se il modulo è guasto.

Sezione 4: Configurazione di altri bus di campo

Il software PC Control supporta i collegamenti a diversi tipi largamente diffusi di bus di campo. Questa sezione del manuale descrive la configurazione dei parametri per il collegamento a DeviceNet e Profibus.

Configurazione degli I/O DeviceNet

Il driver DeviceNet permette di comunicare con un massimo di 63 nodi su una rete DeviceNet. Il driver ha un'interfaccia a configurazione aperta che supporta tutte le unità con approvazione ODVA. Esiste un'ampia gamma di unità I/O in grado di soddisfare varie esigenze di controllo.

Il driver DeviceNet usa una scheda S-S Technologies 5136-DN. Questa scheda esegue la scansione delle unità I/O sulla rete.

Prestazioni

DeviceNet supporta fino a 64 nodi. Con un MAC ID riservato per la scheda 5136-DN, restano 63 nodi a disposizione dell'applicazione. Il driver supporta sia le connessioni Polled I/O che le connessioni Bit Strobed I/O. Non è supportato lo scambio esplicito di messaggi. Sono supportate velocità di trasmissione di 125, 250 e 500 Kbaud.

Il driver configura la scheda scanner 5136-DN come DeviceNet Master. Un DeviceNet Master "possiede" gli slave i cui MAC ID compaiono nella sua lista di scansione. Salvo che per il controllo della duplicazione di MAC ID, uno Slave non può iniziare una comunicazione senza un comando del Master. La scheda 5136-DN è impostata come un I/O scanner, al servizio di tutte le unità abilitate nella configurazione del driver.

Installazione e configurazione delle unità sulla rete DeviceNet

Per le istruzioni relative all'installazione di una unità sulla rete DeviceNet, fare riferimento al manuale dell'operatore dell'unità considerata. Fare attenzione a impostare il MAC ID di ciascuna unità in modo da evitare conflitti di indirizzamento. Molte unità semplici sono configurabili mediante microinterruttori. Tuttavia, esistono unità più complesse, che devono essere configurate in linea attraverso la rete. Per essere configurate correttamente, tali unità richiedono una utilità di gestione DeviceNet. Per la loro configurazione la GE Fanuc raccomanda di utilizzare un pacchetto software con approvazione ODVA.. Contattare il distributore DeviceNet per ottenere un elenco dei fornitori di software per la gestione della rete.

Esistono varie finestre di dialogo dedicate alla configurazione di una rete DeviceNet. Questa sezione contiene un sommario di queste finestre di dialogo e delle loro funzioni. Per le informazioni dettagliate su ciascuna finestra di dialogo si rinvia all'aiuto in linea.

Dialogo:	Descrizione
Board	Questa finestra di dialogo permette di configurare la scheda scanner 5136-DN per la rete DeviceNet.
Fault Detection	Questa finestra di dialogo permette di scegliere gli errori di rete e unità significativi.
Edit DeviceNet Nodes	Questa finestra di dialogo permette di selezionare e di editare ciascuna unità della rete.
Device Edit	Questa finestra di dialogo permette di editare l'unità DeviceNet al MAC ID specificato.
I/O Stream Editor	Questa finestra di dialogo permette di definire l'esatto formato dei flussi in ingresso e in uscita per l'unità selezionata. Permette anche di associare informazioni simboliche a ciascun componente del flusso.
Stream Component Edit	Questa finestra di dialogo permette di assegnare un nome simbolico al componente selezionato del flusso e di definirne il tipo.
Variabili globali	Quando nella configurazione del DeviceNet sono definiti determinati simboli, il driver fornisce automaticamente un insieme di variabili globali contenenti informazioni di stato e di errore. Se è definito un nome di scheda, il driver creerà un insieme di variabili globali che forniranno informazioni di stato e di errore sul bus CAN. Se è definito un nome di unità, il driver fornirà una variabile che riporterà lo stato dell'unità.
Variabili globali bus CAN	Se nel dialogo Board è specificato un nome di scheda, il driver creerà automaticamente un insieme di variabili globali che forniranno informazioni di stato e di errore sul bus CAN. Per creare queste variabili, il driver appende un simbolo al nome della scheda. La cancellazione del nome della scheda nel dialogo Board disabilita la creazione di queste variabili.
Variabili globali stato unità	Se nel campo Symbol Name del dialogo Device Edit è definito un nome di unità, il driver creerà automaticamente una variabile di stato dell'unità. Per creare questa variabile, il driver appende un simbolo al nome dell'unità. Se il campo Symbol Name del dialogo Device Edit è nullo, la creazione di questa variabile è disabilitata.

Configurazione degli I/O Profibus

Il software PC Control comunica con un rete Profibus mediante una scheda Hilscher CIF30/CIF104, CIF30_FMS, o CIF30_PB.

La configurazione del driver Profibus è un processo a due stadi. Prima è necessario configurarlo con il software SyconDP Configuration, creato Synergetic Microsystem. Poi tutti i dati di configurazione devono essere completati all'interno del PC Control in modo similare. Il processo di configurazione della rete Profibus all'interno del software SyconDP è descritto dettagliatamente qui sotto.

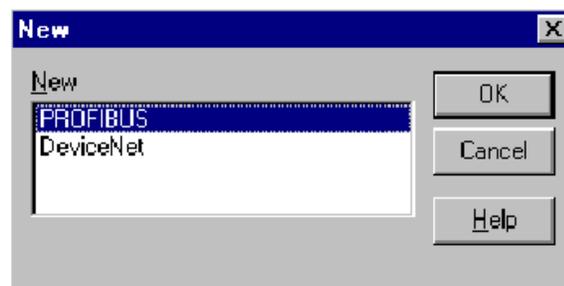
Prima di iniziare, accertarsi che i file .GSD da utilizzare siano stati installati nella cartella appropriata.

Program Files\Hilscher
GmbH\SyCon\Fieldbus\Profibus\GSD

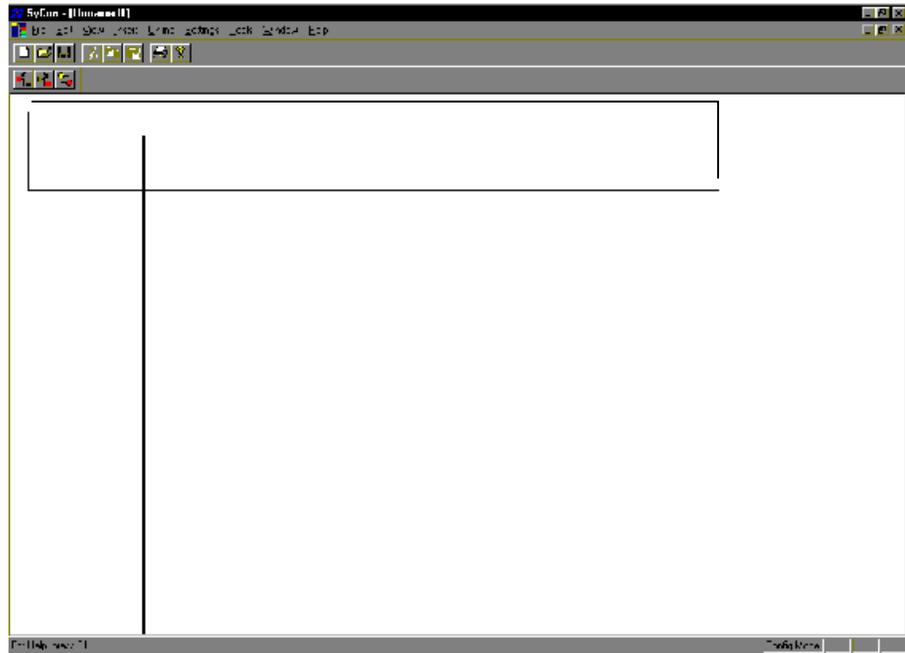
I file necessari sono:

gef_0534.gsd	Hil_7506.gsd
Hil_7501.gsd	Hil_7507.gsd
Hil_7502.gsd	Hms_1002.gsd
Hil_7503.gsd	UN_COMBI.gsd
Hil_7504.gsd	UN_DP.gsd
Hil_7505.gsd	VMXM0534.gsd

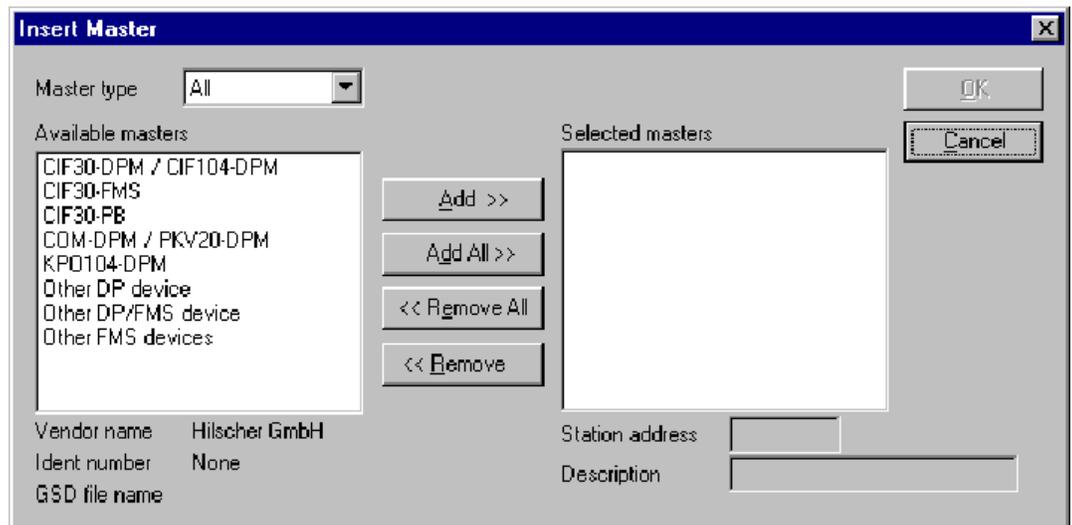
1. Avviare il software SyCon (SYstem CONfigurator) e creare un nuovo file. Comparirà la finestra di dialogo "New



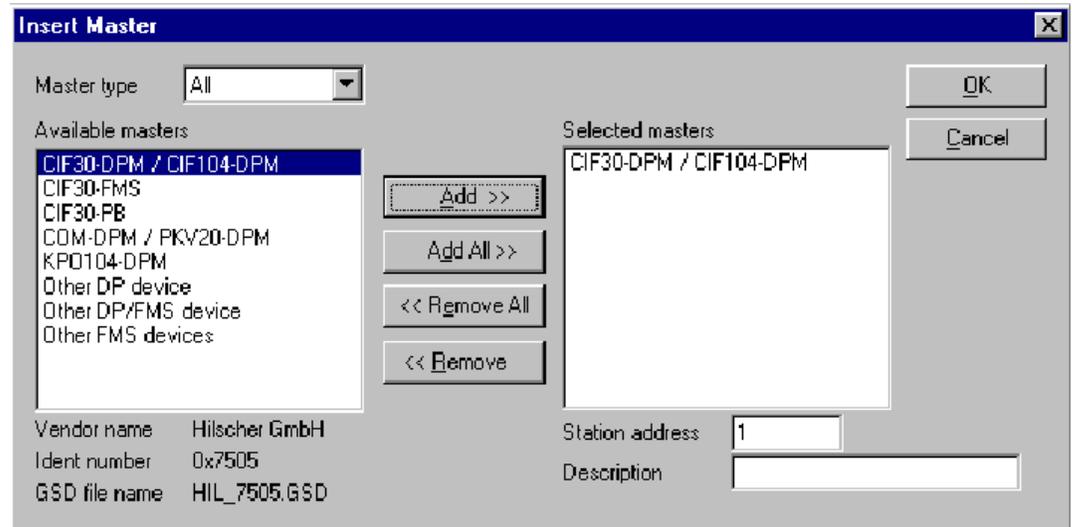
- Evidenziare la scelta PROFIBUS e premere OK. Comparirà lo schermo principale.



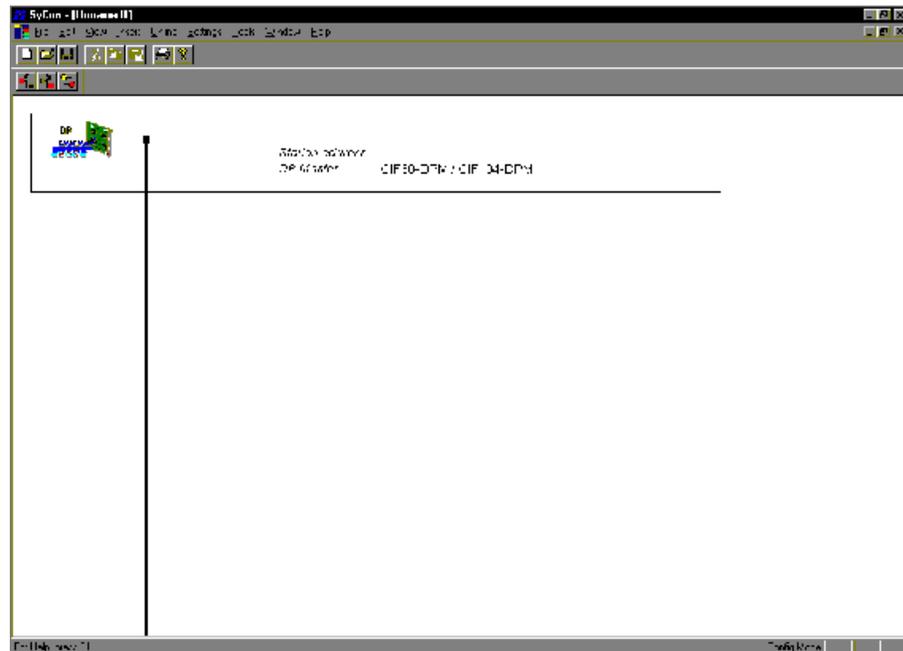
- Selezionare il pulsante di estrema sinistra della barra degli strumenti (Insert Master), oppure scegliere Insert e selezionare Master sulla lista che compare. Posizionare il cursore sulla finestra rettangolare e cliccare con il pulsante sinistro del mouse. Comparirà la finestra di dialogo Insert Master.



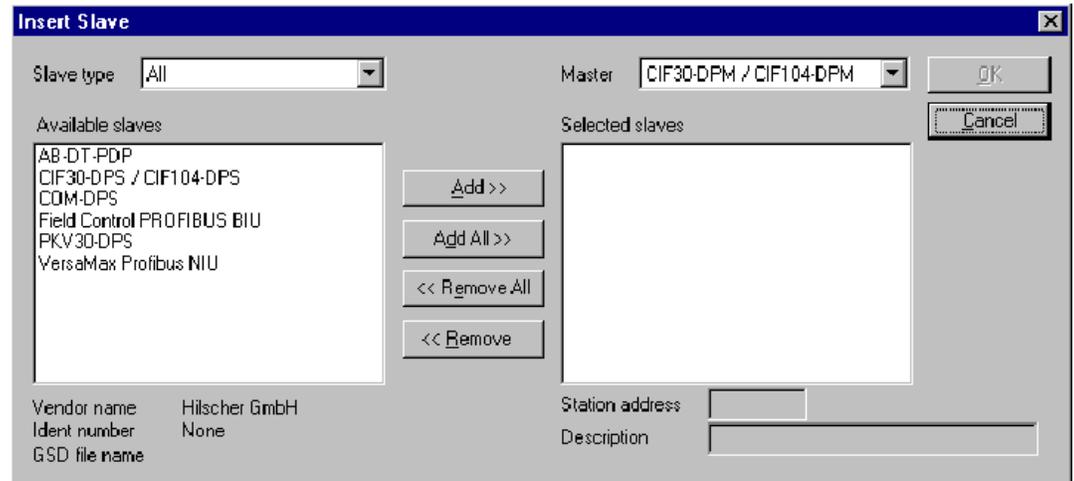
4. Selezionare la scheda master utilizzata e cliccare sul pulsante Add, poi cliccare su OK.



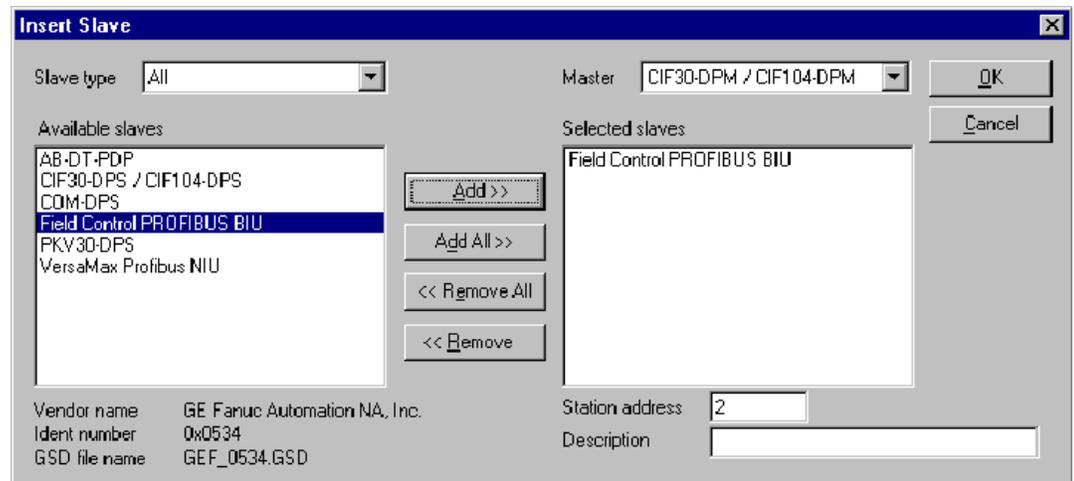
Dovrebbe ora essere visibile il seguente schermo di rete, contenente la scheda Master selezionata nel passo precedente.



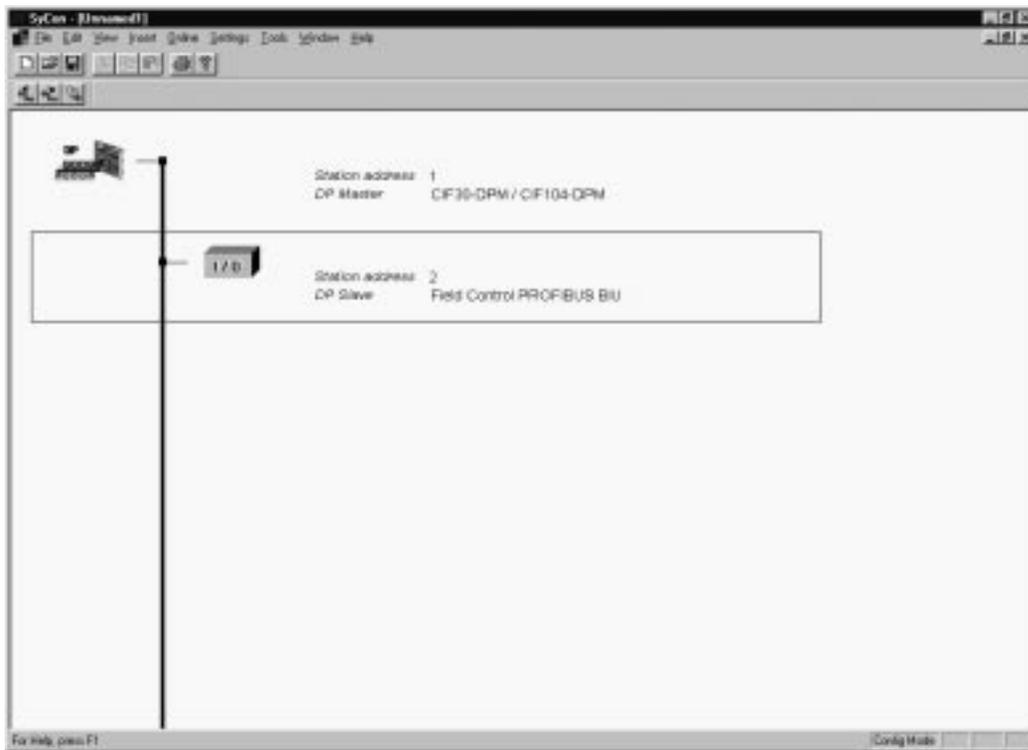
- Selezionare il pulsante Insert Slave della barra degli strumenti, oppure scegliere Insert e poi Slave sulla lista che compare. Posizionare il cursore sulla finestra rettangolare e cliccare con il pulsante sinistro del mouse. Ora dovrebbe essere visualizzato il seguente schermo, che contiene gli Slave disponibili per tutti i file GSD installati nella cartella \Profibus.



- Selezionare l'unità slave appropriata e cliccare sul pulsante Add. L'indirizzo dell'unità sul Profibus viene acquisito automaticamente, assumendo il primo indirizzo disponibile. E' possibile cambiare questo indirizzo per farlo corrispondere all'indirizzo effettivo dello slave.



Dopo la selezione di OK, dovrebbe essere visualizzato il seguente schermo di rete. Ripetere questo passo per tutte le unità slave presenti nel sistema.



7. E' ora possibile editare la configurazione della o delle unità slave. Fare doppio click sull'unità che si intende editare. Comparirà una finestra di dialogo simile a quella presentata qui sotto. I dati contenuti in questa finestra di dialogo dovrebbero corrispondere al file GSD relativo all'unità slave selezionata.

Slave Configuration

General

Device: Field Control PROFIBUS BIU Station address: 2

Description:

Activate device in actual configuration

Enable watchdog control GSD file: GEF_0534.GSD

Max. length of in-/output data: 260 Byte Length of in-/output data: 0 Byte

Max. length of input data: 130 Byte Length of input data: 0 Byte

Max. length of output data: 130 Byte Length of output data: 0 Byte

Max. number of modules: 17 Number of modules: 0

Module	Inputs	Outputs	In/Out	Identifier
IC670PBI001 Profibus BIU Slot0			1 Word	0x70
IC670MDL643,5/12VDC,In,16pt	1 Word			0x50
IC670MDL640,24VDC,In,16pt	1 Word			0x50
IC670MDL644,24VDC,Fast In,16pt	1 Word			0x50
IC670MDL641,48VDC,In,16pt	1 Word			0x50
IC670MDL240,115VAC,In,16pt	1 Word			0x50
IC670MDL642,125VDC,In,16pt	1 Word			0x50
IC670MDL241,240VAC,In,16pt	1 Word			0x50

Idx	Module	Type	I Addr.	Type	O Addr.

Assigned master
Station address 1
1 / CIF30-DPM / CIF104-DI

Actual slave
Station address 2
2 / Field Control PROFIBUS

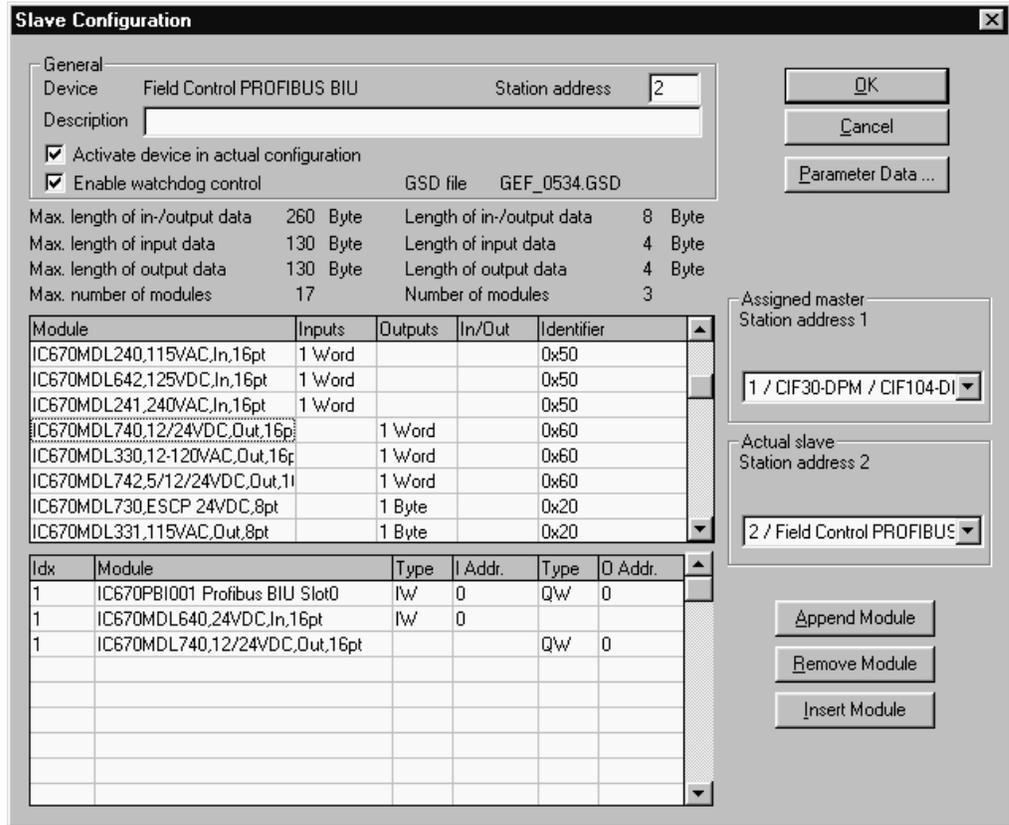
Buttons: OK, Cancel, Parameter Data ..., Append Module, Remove Module, Insert Module

In questo esempio, l'unità slave è un Field Control BIU modulare e per stabilire la quantità di dati I/O che saranno scambiati con il Master devono essere aggiunti i singoli moduli di I/O. Nel caso del Field Control, occorre prima selezionare la configurazione Profibus BIU Slot0. Poi si selezionano i singoli moduli di I/O in ordine di slot.

Nota

Per i moduli intelligenti esistono due entrate per ciascun modulo. Se si sta utilizzando la funzione di configurazione automatica del Field Control, si dovrà selezionare la configurazione Status/Control prima di selezionare la porzione I/O della configurazione.

Dovrebbe ora essere visualizzata una finestra di dialogo simile a questa:



8. Premere il pulsante OK per ritornare allo schermo di rete principale.
9. Per scaricare la configurazione generata sulla scheda master, selezionare Download sul menu Online. Comparirà la finestra di dialogo Device Assignment. Selezionare Board 0 e scegliere il modulo CIF30-DPM sulla lista visualizzata. Per scaricare la configurazione cliccare su OK.



10. Alla fine di questa operazione visualizzare la configurazione di ciascuna unità slave e prendere nota dei dati I Addr e Q Addr per ciascuno dei

moduli I/O. Nello schermo presentato qui sotto questi indirizzi sono 2.
 Questi dati saranno necessari per la configurazione sotto PC Control.

Slave Configuration

General

Device: Field Control PROFIBUS BIU Station address: 2

Description:

Activate device in actual configuration

Enable watchdog control GSD file: GEF_0534.GSD

Max. length of in-/output data: 260 Byte Length of in-/output data: 8 Byte

Max. length of input data: 130 Byte Length of input data: 4 Byte

Max. length of output data: 130 Byte Length of output data: 4 Byte

Max. number of modules: 17 Number of modules: 3

Module	Inputs	Outputs	In/Out	Identifier
IC670PBI001 Profibus BIU Slot0			1 Word	0x70
IC670MDL643,5/12VDC,In,16pt	1 Word			0x50
IC670MDL640,24VDC,In,16pt	1 Word			0x50
IC670MDL644,24VDC,Fast In,16pt	1 Word			0x50
IC670MDL641,48VDC,In,16pt	1 Word			0x50
IC670MDL240,115VAC,In,16pt	1 Word			0x50
IC670MDL642,125VDC,In,16pt	1 Word			0x50
IC670MDL241,240VAC,In,16pt	1 Word			0x50

Idx	Module	Type	I Addr.	Type	O Addr.
1	IC670PBI001 Profibus BIU Slot0	IW	0	QW	0
1	IC670MDL640,24VDC,In,16pt	IW	2		
1	IC670MDL740,12/24VDC,Out,16pt			QW	2

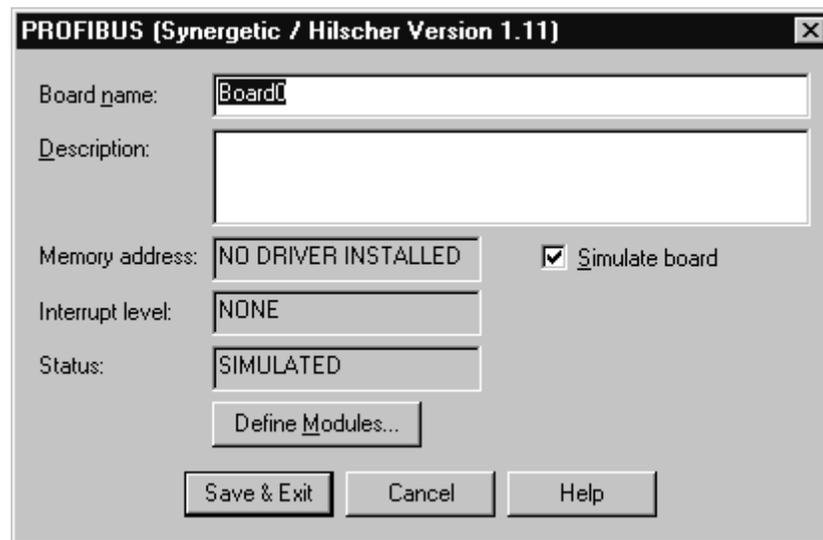
Assigned master
 Station address 1
 1 / CIF30-DPM / CIF104-DI

Actual slave
 Station address 2
 2 / Field Control PROFIBUS

Buttons: OK, Cancel, Parameter Data ...

Buttons: Append Module, Remove Module, Insert Module

11. Avviare l'applicazione PC Control, Program Editor..
12. Generare una nuova configurazione (scegliere New Config del menu File).
13. Nel primo slot vuoto, configurare un Profibus-DP (Hilscher CIF30-DPM).
14. Per configurare gli I/O sulla rete connessa a questa scheda, cliccare sul pulsante Define Board. Comparirà la seguente finestra di dialogo.



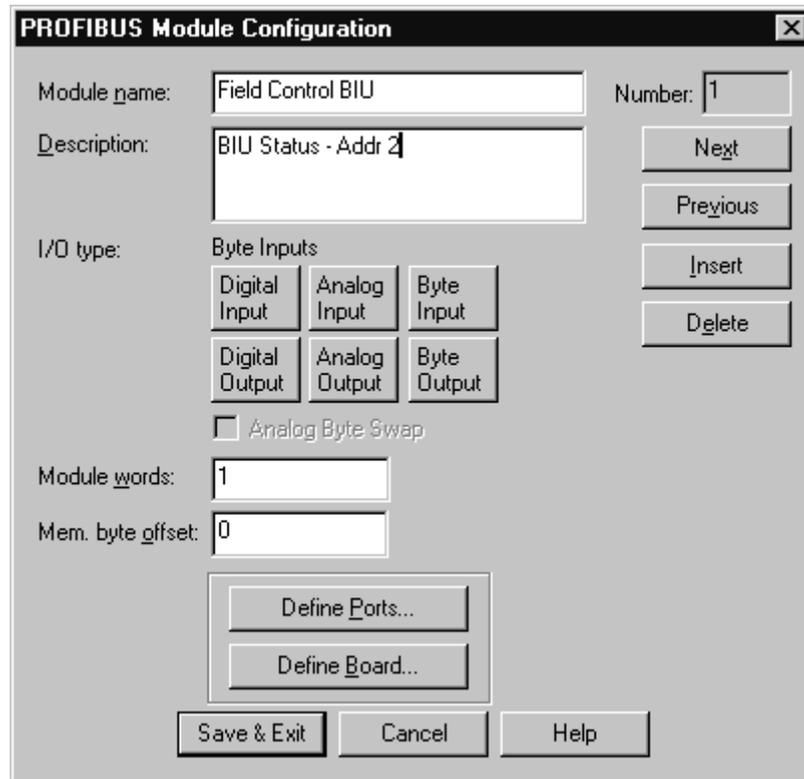
A questo punto è possibile dare alla scheda un nome significativo o lasciare il nome visualizzato per difetto.

15. Per definire la mappatura dei dati I/O della rete sulle variabili di PC Control, selezionare il pulsante Define Modules. Sarà visualizzata la finestra di dialogo PROFIBUS Module Configuration.

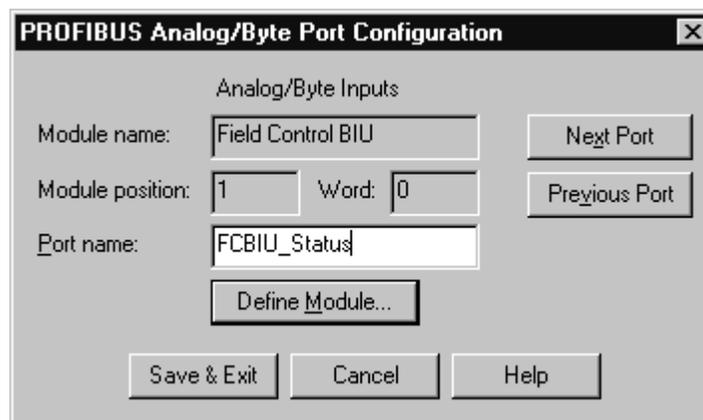
Deve essere creata una configurazione modulo per ciascun dato I/O ricevuto o trasmesso dal Master. Prima definire un nome e una descrizione per ciascun dato. Poi definire il tipo di dato, la lunghezza e l'offset nella memoria della scheda master. (Questo è l'offset annotato precedentemente.) Qui sotto sono proposti degli schermi di esempio per un Field Control BIU con un modulo di ingressi a 16 punti e un modulo di uscite pure a 16 punti.

Nota

Lo stato/controllo del Field Control BIU e i moduli discreti a 16 punti sono definiti come Byte Input o Byte Output con una parola per modulo..



16. Deve anche essere definito uno specifico nome di porta a ciascun dato definito. Dopo aver definito la configurazione del modulo, premere il pulsante Define Port. E' possibile attribuire alla porta il nome che si desidera o usare il nome che compare per difetto. Poiché questo nome sarà utilizzato per fare riferimento ai dati all'interno del programma, è utile scegliere un nome significativo. Lo schermo di definizione della porta per la parola di stato del BIU è presentato qui sotto.



Note

Per i moduli con più parole di dati, dovrà essere definita una porta per ciascuna di esse. Un esempio di questo è un modulo analogico a 8 canali. In tal caso dovranno essere definite 8

porte. E' possibile passare da una porta all'altra premendo il pulsante "Next Port".

Qui sotto vengono forniti alcuni esempi di schermi di configurazione modulo e configurazione porta per gli altri dati del BIU e dei moduli.

Configurazione modulo per la parola di controllo del BIU

The screenshot shows a dialog box titled "PROFIBUS Module Configuration". It contains the following fields and controls:

- Module name:** Field Control BIU
- Number:** 2
- Description:** BIU Control Word - Addr 2
- I/O type:** Byte Outputs (selected). Other options include Digital Input, Analog Input, Byte Input, Digital Output, and Analog Output.
- Analog Byte Swap
- Module words:** 1
- Mem. byte offset:** 0
- Buttons:** Next, Previous, Insert, Delete, Define Ports..., Define Board..., Save & Exit, Cancel, Help.

Configurazione porta per la parola di controllo

The dialog box is titled "PROFIBUS Analog/Byte Port Configuration". It contains the following fields and buttons:

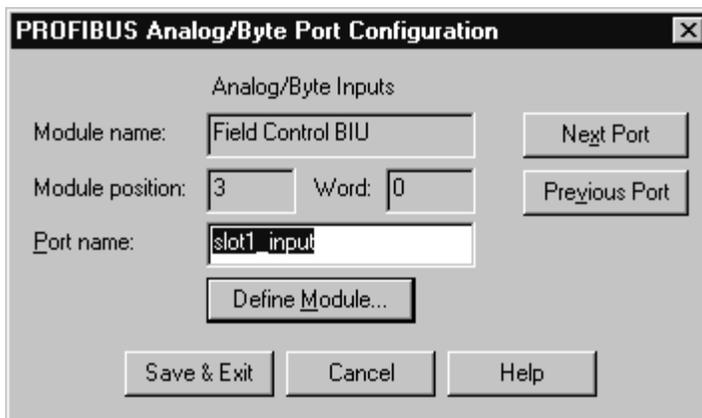
- Module name: Field Control BIU
- Module position: 2
- Word: 0
- Port name: FCBIU_control
- Buttons: Next Port, Previous Port, Define Module..., Save & Exit, Cancel, Help

Configurazione modulo per un modulo di ingressi a 16 punti

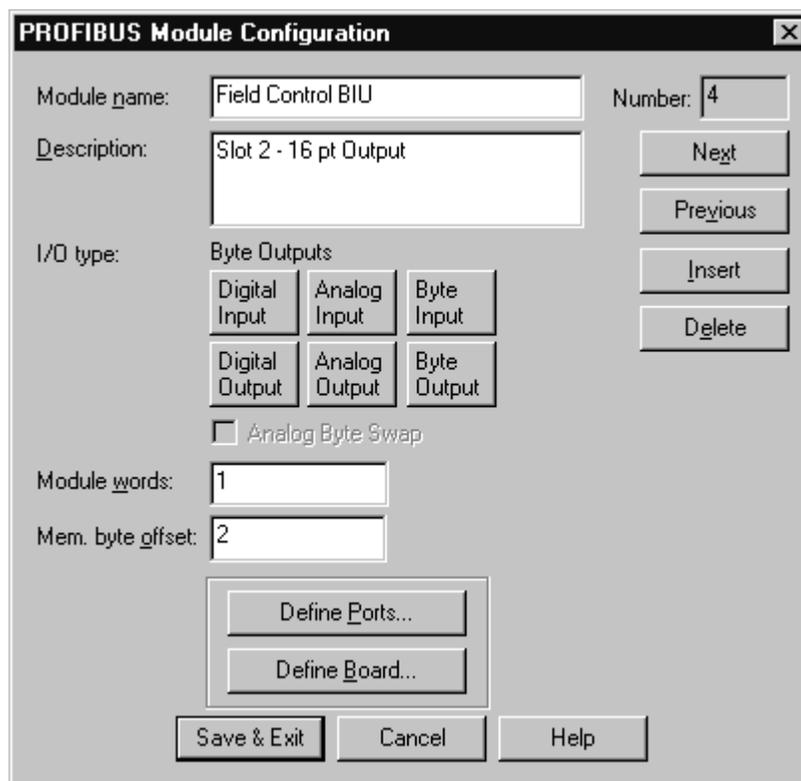
The dialog box is titled "PROFIBUS Module Configuration". It contains the following fields and buttons:

- Module name: Field Control BIU
- Number: 3
- Description: Slot1 - 16pt Input
- I/O type: Byte Inputs (Digital Input, Analog Input, Byte Input) and Byte Outputs (Digital Output, Analog Output, Byte Output)
- Analog Byte Swap
- Module words: 1
- Mem. byte offset: 2
- Buttons: Next, Previous, Insert, Delete, Define Ports..., Define Board..., Save & Exit, Cancel, Help

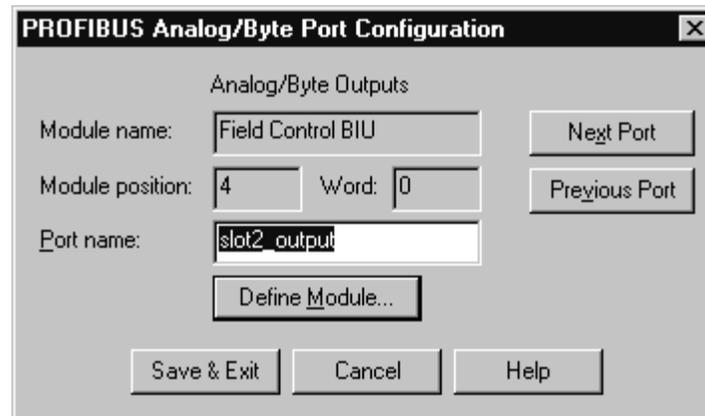
Configurazione porta per un modulo di ingressi a 16 punti



Configurazione modulo per un modulo di uscite a 16 punti

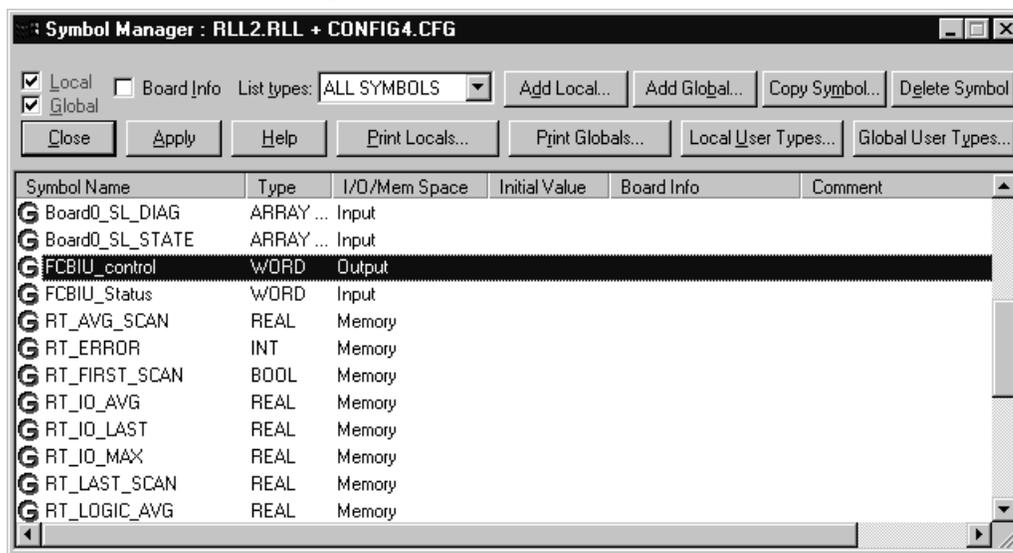


Configurazione porta per un modulo di uscite a 16 punti

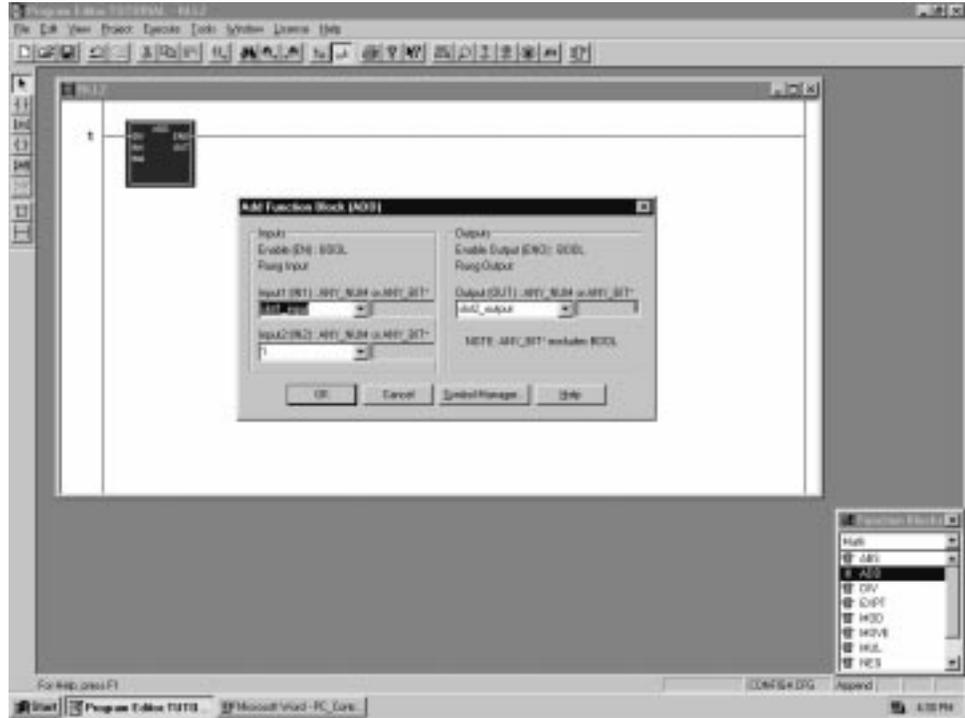


17. Il passo finale di questo processo consiste nel salvare e attivare la configurazione definita.

Dall'interno del Program Editor, è possibile accedere alle porte attraverso il symbol manager..



Qui sotto viene dato un esempio di utilizzo di un modulo di ingressi a 16 punti installato nello slot 1 in un blocco funzionale ADD.



Sezione 5: Scambio dinamico dei dati (DDE, Dynamic Data Exchange)

L'interfaccia DDE

Il software PC Control dispone di un'opzione che permette a software di terze parti di comunicare con il software del sistema di comando attraverso un'interfaccia DDE. Quando è abilitata l'interfaccia DDE, il software PC Control accetta comandi da programmi esterni, così come richieste di lettura/scrittura degli I/O del sistema di comando e delle variabili globali.

Comunicazioni DDE con il software Microsoft Excel

E' possibile usare le prestazioni DDE del software Microsoft Excel per lo scambio dei dati contenuti nelle variabili globali. E' possibile inviare ad Excel i dati del sistema di comando per compendiarli o analizzarli. E' anche possibile trasferire dati da Excel a PC Control per controllare l'esecuzione dei programmi del sistema di comando o per fornire prestazioni quali un sistema di gestione di ricette a basso livello.

Trasferimento di dati ad Excel

Per trasferire il valore di un simbolo globale a una cella di un foglio di lavoro Excel, inserire una formula simile alla seguente nella cella che deve ricevere il valore:

```
=ProgMgr|'_main _main'!VariableName
```

o

```
=ProgMgr|'_main _main'!'ArrayName[4]'
```

La stringa '_main _main' è un argomento DDE e permette di acquisire le variabili globali. Digitare uno e un solo spazio fra i due "_main", come si vede nella formula qui sopra. Al posto di "VariableName" scrivere il nome della variabile globale che si desidera trasferire. Il nome della variabile deve essere identico (maiuscole/minuscole e ortografia) a quello usato nell'applicazione PC Control. Il secondo formato si usa per accedere agli elementi degli array specificandone l'indice. Tuttavia, a causa delle parentesi quadre, il nome dell'elemento deve essere racchiuso tra virgolette singole.

Questa formula stabilirà un "Hot DDE link" con il sistema di comando, che aggiornerà il valore usato dal foglio di lavoro Excel ad ogni variazione della variabile nel sistema di comando. Il capitolo "Transferring Data to Excel" dell'aiuto in linea contiene un esempio dettagliato. Per altre informazioni sull'uso delle formule in un foglio di lavoro Excel, fare riferimento alla documentazione Excel per l'utente.

Trasferimento di dati al sistema di comando

Il trasferimento di un valore da Excel al sistema di comando richiede la codifica di una routine di transazione DDE in una macro di Excel simile alla seguente:

```
Sub transfer()  
  
    Dim x  
  
    x = Application.DDEInitiate("ProgMgr", "_main _main")  
  
    Set rangeToPoke = Sheets("Sheet1").Cells(7, 11)  
  
    Application.DDEPoke x, "VariableName", rangeToPoke  
  
    Application.DDETerminate x  
  
End Sub
```

Questa routine, quando viene eseguita, trasferisce il valore della cella K7 (riga 7, colonna 11) alla variabile globale il cui nome compare al posto di VariableName. Come si è detto precedentemente, tra “_main” e _main” deve essere digitato uno e un solo spazio. Sostituire Sheet1 con il nome del foglio di lavoro contenente il valore da trasferire.

Trasferimento di valori al sistema di comando su richiesta

Il trasferimento di valori da Excel al sistema di comando su richiesta richiede la codifica di una routine di transazione DDE in una macro Excel simile alla seguente:

```
Dim TimeSet As Double

Sub RunMeFirst()

    TimeSet = Now + TimeValue("00:00:05")

    Application.OnTime TimeSet, "Transfer"

End Sub

Sub Transfer()

    Dim x, y

    Dim z As Variant

    x = DDEInitiate("ProgMgr", "_main _main")

    z = DDERequest(x, "Trans1")

    y = Val(z(1))

    If y = 1 Then

        Set rangeToPoke = Sheets("Sheet1").Cells(2, 3)

        DDEPoke x, "Data1", rangeToPoke

        Set rangeToPoke = Sheets("Sheet1").Cells(3, 3)

        DDEPoke x, "Data2", rangeToPoke

        Set rangeToPoke = Sheets("Sheet1").Cells(4, 3)

        DDEPoke x, "Data3", rangeToPoke

        Set rangeToPoke = Sheets("Sheet1").Cells(1, 3)

        DDEPoke x, "Trans1", rangeToPoke

    End If

    DDETerminate x

    RunMeFirst

End Sub
```

Quando viene eseguita, questa macro controlla ogni 5 secondi una variabile macro di nome Trans1. Se Trans1 è 1, la macro trasferisce il valore della cella C2 (riga 2, colonna 3) alla variabile globale di nome Data1, allo stesso modo trasferisce C3 a Data2 e C4 a Data3. Infine, la macro trasferisce C1, il cui valore è preimpostato a 0, a Trans1, cancellando la richiesta di trasferimento. Come si è detto precedentemente, tra “_main” e _main” deve essere digitato uno e un solo spazio. Sostituire Sheet1 con il nome del foglio di lavoro contenente i valori da trasferire. L’intervallo di controllo di Trans1 (5 secondi) può essere modificato cambiando il valore di TimeValue nella funzione RunMeFirst.

Sezione 6: Importazione/esportazione di configurazioni

La prestazione importazione/esportazione dà le seguenti possibilità:

- Esportazione della configurazione per driver che non hanno capacità di taglia e incolla; usare un editore di testi per tagliare e incollare la configurazione; poi reimportare la configurazione nel sistema di comando.
- Editazione fuori linea della configurazione. Anziché il sistema di controllo, per editare le configurazioni è possibile usare un editore di testi o un'utilità compatibile con il formato CSV (Comma Separated Variable), come Microsoft Excel). Oppure è possibile usare proprio software di configurazione tramite dialoghi (con Visual Basic, Visual C++, ecc.) per creare file CSV, importando poi questi file nel sistema di controllo.
- Creazione di proprie applicazioni per automatizzare la generazione di configurazioni personalizzate.

L'importazione/esportazione in formato CSV è supportata per i seguenti tipi di dati di configurazione: simboli globali, driver I/O Yaskawa, driver GE 90/30, driver PCIM.

La configurazione è esportata in un file CSV, che può essere letto da un editore di testi o da un foglio di lavoro elettronico.

Per esportare una configurazione in un file CSV

1. Aprire la configurazione da esportare.
2. Selezionare *Export Config to CSV* nel menu *File*. Comparirà un dialogo di selezione file.



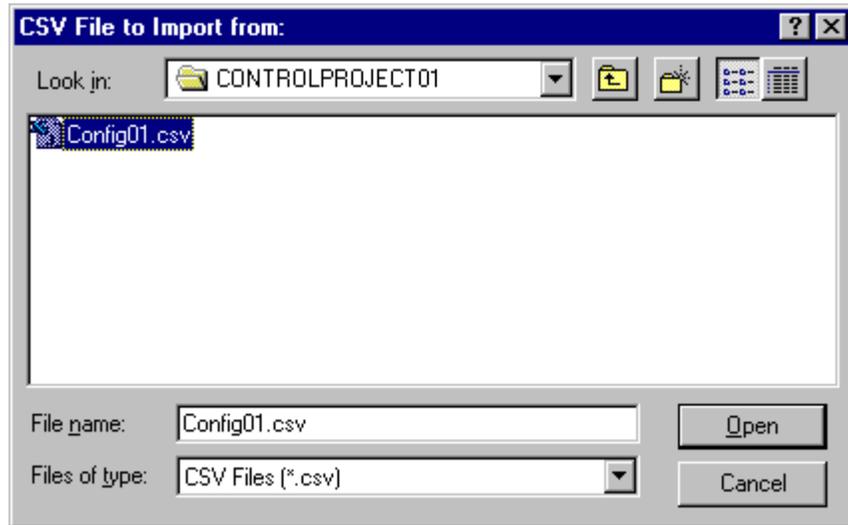
3. Selezionare o digitare un nome di file per l'esportazione e cliccare su *Open* per continuare. Comparirà la finestra di dialogo *Export Config to CSV*.



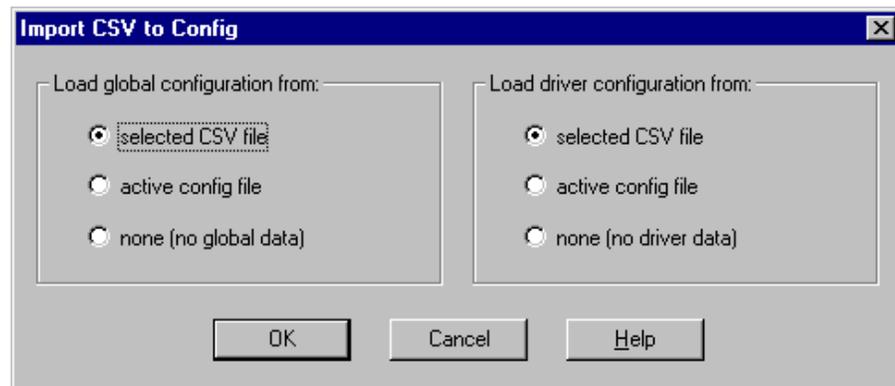
4. Selezionare il tipo desiderato di esportazione e cliccare su *OK* per continuare. La configurazione viene salvata nel file designato.

Per importare un file CSV in una configurazione

1. Selezionare *Import CSV to Config* nel menu *File*. Comparirà la richiesta di chiudere la configurazione attualmente aperta.
2. Comparirà un dialogo di selezione file.



3. Selezionare o digitare il nome del file di configurazione da importare e cliccare su *Open* per continuare. Comparirà la finestra di dialogo *Import CSV to Config*.



4. Selezionare le opzioni di importazione desiderate. E' possibile scegliere di importare i simboli globali e/o i dati del driver, di usare i simboli globali o i dati del driver della configurazione attiva, o di iniziare la configurazione senza simboli globali o dati del driver.
5. Cliccare su *OK* per proseguire. I dati di configurazione vengono importati e diventano la configurazione attiva.

Il Program Editor si usa per sviluppare, mettere a punto, compilare ed eseguire i programmi applicativi. Ogni volta che viene avviato il Program Editor compare la richiesta della password, che determina il livello di accesso. La password standard per l'accesso illimitato è "4567".

Questo capitolo fornisce le seguenti informazioni:

- Come configurare i simboli
- Come strutturare e creare programmi applicativi RLL
- Come strutturare e creare Diagrammi funzionali sequenziali (FSC)
- Come creare programmi applicativi in testo strutturato (ST)
- Come creare programmi applicativi nel formato lista di istruzioni (IL)
- Programmazione dei movimenti
- Editazione in linea

Sezione 1: Configurazione dei simboli

Che cosa sono i simboli

Gli strumenti di programmazione di PC Control permettono di definire e usare i simboli, che rappresentano specifiche posizioni della memoria interna nelle quali risiedono le informazioni. Il contenuto di queste informazioni è definito dal tipo di dato: numero reale, intero, stringa di caratteri, ecc. Per definire un simbolo, specificandone il nome, il tipo di dato, e la visibilità, che può essere locale o globale, si usa il Symbol Manager. Le definizioni dei simboli globali sono registrate nel file di configurazione attivo. Le definizioni dei simboli locali sono registrate nel file del programma applicativo (SFC, RLL, Testo strutturato o Lista di istruzioni).

I punti di I/O e i simboli di sistema e di runtime sono predefiniti e possono essere visti e usati all'interno dei programmi, ma non possono essere editati. Essi si presentano come simboli globali.

Visibilità dei simboli

La visibilità di un simbolo può essere locale ad un programma o globale.

Simboli locali

Un simbolo locale può essere usato solo all'interno del programma nel quale è definito o nei passi macro chiamati dal programma. Perché possa essere definito un simbolo locale da usare al suo interno, un programma deve essere aperto. Non è possibile accedere ai simboli locali dall'interno dell'interfaccia operatore e per operazioni DDE. Le definizioni dei simboli locali sono registrate nel file del programma applicativo.

Simboli globali

Un simbolo globale può essere usato da tutti i programmi di un progetto. I simboli globali possono essere usati all'interno dell'interfaccia operatore e per operazioni DDE. Le definizioni dei simboli globali sono registrate nel file di configurazione attivo.

Punti di I/O

I punti di I/O sono posizioni esterne. Poiché è possibile usarli in un programma esattamente come i simboli, i punti di I/O compaiono nel Symbol Manager e sono elencati come simboli globali.

I punti di I/O funzionano a tutti gli effetti come i simboli globali: è possibile usarli in qualsiasi programma, all'interno dell'interfaccia operatore e per operazioni DDE. I simboli relativi agli I/O possono essere editati solo dall'utilità di configurazione, non è possibile editare un punto di I/O con il Symbol Manager.

Blocchi funzionali

La maggior parte dei blocchi funzionali deve essere istanziata. Per poter essere usata nei programmi ST (testo strutturato) o IL (lista di istruzioni), un'istanza del tipo di blocco funzionale deve essere dichiarata esplicitamente nel Symbol Manager.

Oggetti di sistema (PID, PRGBC e TMR)

Questi oggetti devono essere istanziati. Per poter essere usata in un programma ST, un'istanza del tipo di oggetto di sistema deve essere dichiarata esplicitamente nel Symbol Manager.

Identificatori

Gli identificatori sono i nomi dei simboli e hanno le seguenti caratteristiche:

- Sono costituiti da lettere maiuscole e minuscole (A—Z e a—z), numeri (0—9), e dal carattere di sottolineatura (_).
- Devono iniziare con una lettera o con un solo carattere di sottolineatura.
- Una lettera maiuscola è diversa dalla corrispondente lettera minuscola.
- Per determinare l'unicità di un identificatore, sono considerati tutti i caratteri che lo costituiscono.
- Non possono contenere più caratteri di sottolineatura consecutivi.
- Non possono contenere spazi.
- Possono essere lunghi fino a 100 caratteri.
- Non possono essere una parola riservata del sistema. Queste sono elencate nel paragrafo "Parole riservate".
- Devono essere unici nel rispettivo campo di visibilità.

I seguenti identificatori sono validi:

```
_Sym1
Sym_Two_A
SYM      (SYM e sym sono diversi e quindi)
Sym
A        (A e AA sono diversi e quindi unici)
AA
```

I seguenti identificatori non sono validi:

```
1A      (inizia con un numero)
Sym__Two (due caratteri di sottolineatura consecutivi)
Sym Two (contiene uno spazio)
Sym&One (contiene un carattere non valido)
END     (un identificatore riservato)
```

Costanti letterali

Le costanti letterali si usano per definire o rappresentare dei valori. Ad esempio, una costante letterale può fornire il valore in ingresso ad una funzione o ad un blocco funzionale, può essere usata per assegnare un valore ad una variabile o ad una costante, e può comparire nelle istruzioni di un programma. Esistono quattro tipi di costanti letterali: numeriche, stringhe di caratteri, durata, ora/data.

Costanti letterali numeriche

I valori numerici letterali possono essere interi o reali. I numeri interi possono essere espressi in formato decimale, in base 2, in base 8 o in base 16. Esempi di valori numerici letterali:

0	456	+34	-7_000	interi
0.0	0.11			reali
2#1010_1010	(170 decimale)			in base 2
8#252	(170 decimale)			in base 8
16#AA	(170 decimale)			in base 16
FALSE	0	TRUE	1	Booleani

Le costanti letterali numeriche hanno le seguenti caratteristiche:

- Una costante letterale numerica può contenere un singolo carattere di sottolineatura (_), che non ne influenza il valore.
- Le costanti letterali reali contengono un punto decimale.
- Le costanti letterali in base 10 possono iniziare con il segno (+ o -).
- Le parole riservate FALSE e TRUE corrispondono rispettivamente ai valori booleani 0 e 1.

Stringhe di caratteri letterali

Una stringa letterale è una stringa di 0 o più caratteri delimitata da virgolette singole ('). In una stringa letterale il carattere \$ (simbolo del dollaro) ha un significato speciale. Se è seguito da due cifre esadecimali, questo viene interpretato come la rappresentazione esadecimale di un codice carattere di otto bit. Il simbolo \$ si usa anche per rappresentare, con una stringa di due caratteri, il simbolo del dollaro (\$), la virgoletta semplice (') e specifici caratteri non stampabili.

Esempi di stringhe di caratteri letterali:

"	una stringa vuota
'XYZ'	una stringa di tre caratteri
' '	uno spazio
'\$41 \$42 \$43'	la stringa di cinque caratteri 'A B C'
'\$'	simbolo del dollaro
'\$' '	virgoletta singola
'\$L' o '\$l'	avanzamento riga
'\$N' o '\$n'	nuova riga
'\$P' o '\$p'	avanzamento pagina
'\$R' o '\$r'	ritorno carrello
'\$T' o '\$t'	tabulazione

Costanti letterali di durata

Queste costanti letterali sono formate da uno dei prefissi T#, TIME#, t#, o time# seguito da una o più unità di tempo. Esempi:

T#1D1H1M1S	1 giorno, 1 ora, 1 minuto, 1 secondo
Time#1d_1h_1m_1s	come sopra
time#25h1ms	25 ore, 1 millisecondo
t#1m_2.5s	1 minuto, 2.5 secondi

Le costanti letterali di durata hanno le seguenti caratteristiche:

- I caratteri che rappresentano le unità di tempo possono essere maiuscoli o minuscoli: D, d=giorni; H, h=ore; M, m=minuti; S, s=secondi; MS, ms=millisecondi.
- Per separare le unità di tempo è possibile usare un carattere di sottolineatura (_).
- E' permesso il "superamento" dell'unità di tempo più significativa.
- L'unità di tempo meno significativa può essere un numero reale (senza esponente).

Costanti letterali che rappresentano ora e data

Queste costanti letterali possono essere scritte con uno dei seguenti formati:

DATE#YYYY-MM-DD	solo la data
D#YYYY:MM:DD	come sopra
TIME_OF_DAY#HH:MM:SS.MS	solo l'ora
TOD#HH:MM:SS.MS	come sopra
DATE_AND_TIME#YYYY-MM-DD-HH:MM:SS.MS	data e ora
DT#YYYY-MM-DD-HH:MM:SS.MS	Come sopra

Esempi:

DATE#1998-02-13
D#1998-02-13
TIME_OF_DAY#12:00:00
TOD#12:00:00.01
DATE_AND_TIME#1998-02-13-12:00:00.01

- Le parole chiave date e time possono essere abbreviate: DATE o D; TIME_OF_DAY o TOD; DATE_AND_TIME o DT.

Tipi di dati

A ciascun simbolo (variabile o costante) deve essere assegnato il tipo di dato. Le caratteristiche dei tipi di dati elementari sono descritte qui sotto. I tipi di dati generici sono descritti nel paragrafo “Tipi di dati generici”. I tipi di dati definiti dall’utente sono descritti nel paragrafo “Tipi di dati definiti dall’utente”,

Nelle descrizioni che seguono:

- “Tipi generici” indica i tipi generici ai quali questo tipo di dato può essere sostituito.
- “Dimensione” è la quantità di memoria occupata da una istanza del tipo di dato (ad esempio, il tipo booleano di 1 bit è in realtà memorizzato in un byte).
- “Campo dei valori” indica i valori che possono essere assunti da un’istanza del tipo di dato.
- “Valore per difetto” è il valore iniziale assunto da un’istanza del tipo di dato, se non altrimenti specificato.

BOOL (Booleano)

Un BOOL può avere uno dei due stati 0 e 1, corrispondenti rispettivamente a FALSE e TRUE.

Tipi generici	ANY, ANY_BIT.
Dimensione	1 bit.
Campo dei valori	0 (FALSE), 1 (TRUE).
Valore per difetto	0

BYTE

Un BYTE è una stringa di 8 bit.

Tipi generici	ANY, ANY_BIT.																		
Dimensione	1 byte.																		
Campo dei valori	Non applicabile.																		
Formato	MSB LSB																		
	<table style="margin: auto;"> <tr> <td style="text-align: center;">7</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">0</td> </tr> <tr> <td></td> <td style="border: 1px solid black; width: 15px; height: 15px;"></td> <td style="border: 1px solid black; width: 15px; height: 15px;"></td> <td style="border: 1px solid black; width: 15px; height: 15px;"></td> <td style="border: 1px solid black; width: 15px; height: 15px;"></td> <td style="border: 1px solid black; width: 15px; height: 15px;"></td> <td style="border: 1px solid black; width: 15px; height: 15px;"></td> <td style="border: 1px solid black; width: 15px; height: 15px;"></td> <td style="border: 1px solid black; width: 15px; height: 15px;"></td> </tr> </table>	7								0									
7								0											
Valore per difetto	00000000																		

DATE

Questo tipo di dato si usa per rappresentare una data (solamente) nel formato YYYY-MM-DD.

In una espressione di tipi di dati DATE, tutti i valori devono essere dello stesso tipo e il risultato deve essere una data.

Tipi generici ANY, ANY_DATE.
 Dimensione 4 byte
 Campo dei valori -
 Valore per difetto D#0001-01-01

DINT (Doppio intero)

Un dato di tipo DINT è un intero con segno composto da una o più cifre (0 .. 9).

Un dato di tipo DINT non può contenere il punto decimale.

Tipi generici ANY, ANY_NUM, ANY_INT.
 Dimensione 4 byte
 Campo dei valori da -2147483648 a +2147483647.
 Valore per difetto 0

DWORD (Doppia parola)

Un dato di tipo DWORD è una stringa si 32 bit..

Tipi generici ANY, ANY_BIT.
 Dimensione 4 byte
 Campo dei valori Non applicabile

Format



Valore per difetto 0

INT (Integer)

Un dato di tipo INT è un intero con segno composto da una o più cifre (0 .. 9).

Un dato di tipo INT non può contenere il punto decimale.

Tipi generici ANY, ANY_NUM, ANY_INT.
 Dimensione 4 byte (un miglioramento rispetto alle specifiche IEC 1131-3)
 Campo dei valori da -2147483648 a +2147483647.
 Valore per difetto 0

REAL

Un dato di tipo REAL è un numero di 64 bit composto da una o più cifre(0 – 9), ha il segno e contiene un punto decimale.

Tipi generici ANY, ANY_NUM, ANY_REAL.

Dimensione 8 byte.

Campo dei valori da -3.402823 E38 a -1.401298 E-45 (numeri negativi),
da +1.401298 E-45 a +3.402823 E38 (numeri positivi).

Valore per difetto 0.0

STRING

Stringa di caratteri ASCII di lunghezza variabile.

Tipi generici ANY.

Dimensione 64 byte.

Valore per difetto " (stringa vuota)

Formato Stringa di caratteri ASCII racchiusa tra virgolette singole. Esempio:
' Questa è una stringa valida. '

TIME

Questo tipo di dati si usa per rappresentare un periodo di tempo nel formato T#[nD][nH][nM][nS][nMS], dove n è un numero di giorni, ore, minuti, secondi o millisecondi.

In una espressione di dati di tipo TIME, tutti i valori devono essere dello stesso tipo e il risultato deve essere un tempo.

Tipi generici ANY.

Dimensione -

Campo dei valori -

Valore per difetto T#0S

Formato

Tipi di dati generici

La seguente tabella fornisce la gerarchia dei tipi di dati e dei tipi generici. I tipi generici sono caratterizzati dal prefisso ANY_ e si usano nella descrizione delle funzioni e dei blocchi funzionali quando è possibile (anziché dettagliare una lunga lista di tipi di dati).

Ad esempio, se l'ingresso di una funzione accetta un dato di tipo ANY_NUM, all'ingresso di quella funzione può essere assegnato un simbolo il cui tipo è REAL, uno dei tipi interi (INT, DINT, ecc.) o una stringa di bit (WORD, DWORD, ecc.). Se l'ingresso di una funzione accetta un dato di tipo ANY_BIT, ad esso può essere assegnato solo un simbolo il cui tipo è WORD, DWORD, ecc.

Gerarchia dei tipi di dati					
ANY_ OR_ DERIVED	ANY	ANY_NUM	ANY_REAL	REAL	
			ANY_INT_OR_BIT	ANY_INT	DINT, INT, UINT
		ANY_BIT		DWORD, WORD, BYTE, BOOL	
	STRING				
	ANY_DATE		DATE, TIME_OF_DAY		
	TIME				
	Derivati				

Tipi di dati definiti dall'utente

In caso di necessità, l'utente può creare i propri tipi di dati strutturati. Una struttura vari membri di diversi tipi base o definiti dall'utente. Si consideri una struttura definita dall'utente denominata UserStructure01 i cui membri siano Usint, di tipo intero, USBool, di tipo booleano e USString, di tipo stringa. Ai singoli membri di questa struttura si accede come segue:

```
UserStructure01.USInt:=101;
UserStructure01.USBool:= TRUE;
UserStructure01.USString:="ABC";
```

I tipi definiti dall'utente sono validi in qualunque punto che accetti un tipo di dati ANY o USER_DEFINED. Riferirsi a "Editazione dei tipi di dati definiti dall'utente" per ulteriori informazioni.

Array

Per accedere a un elemento di un array, specificare il nome simbolico dell'array seguito dal numero dell'elemento racchiuso tra parentesi quadre. Ad esempio, per accedere al quinto elemento di un array chiamato Myarray, si scrive Myarray[5]. Questo se per il limite inferiore dell'array è stato specificato 1. E' anche possibile specificare l'indice di un array con un simbolo di tipo INT racchiuso tra parentesi quadre.

Puntatori

Diffida

I puntatori devono essere usati esclusivamente da esperti. Un uso errato dei puntatori può dare luogo a operazioni imprevedibili e a grandi difficoltà di messa a punto del programma.

Il formato ST (testo strutturato) ha due operatori per i puntatori: l'operatore indirizzo & e l'operatore di indirizzamento *. Questi operatori si usano nelle operazioni di indirizzamento indiretto. Le informazioni sull'uso dei puntatori, riservato ad utenti esperti, si trovano nell'appendice F.

Symbol Manager

Il Symbol Manager si usa per vedere, creare, editare, copiare e cancellare simboli, variabili e costanti, e istanze di blocchi funzionali. I punti di I/O e i simboli di sistema possono solo essere visti. Il Symbol Manager visualizza una lista di simboli locali, definiti nel file attivo, e una lista di simboli globali, definiti nella configurazione attiva.

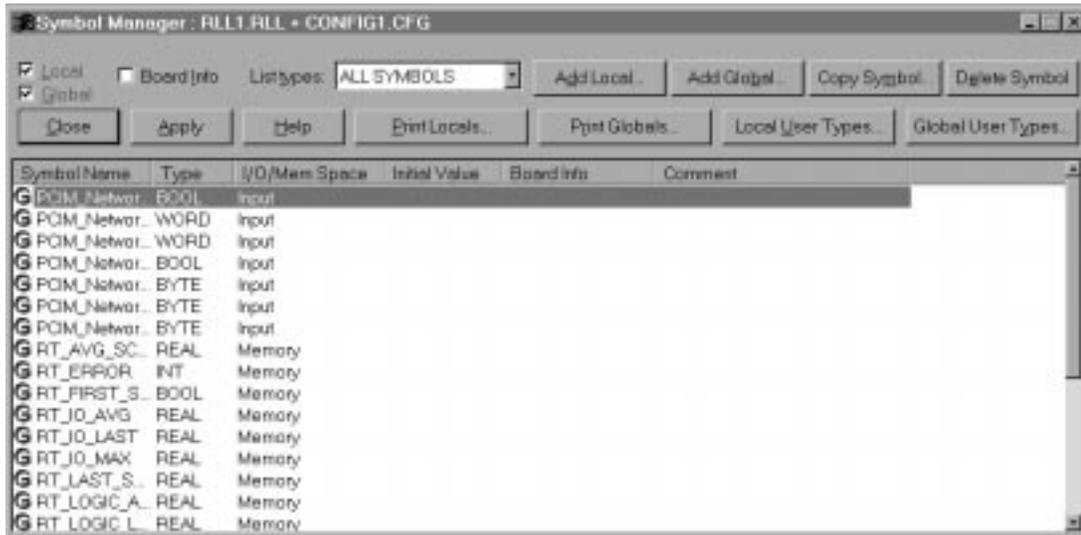
Nel Symbol Manager è possibile aggiungere un simbolo a una frase di un programma in testo strutturato o in lista di istruzioni, oppure un contatto o una bobina a un programma RLL mediante trascinamento. Il trascinamento può anche essere usato per passare informazioni sui simboli ad altri software con funzioni OLE, quali Microsoft WordPad, Word, Excel ed altri. I simboli possono anche essere selezionati su liste visualizzate in varie finestre di dialogo.

Apertura del Symbol Manager

Per aprire il Symbol Manager, selezionare Symbol Manager nel menu *Tools* del Program Editor o dell'Operator Interface Editor.

- Il Symbol Manager può anche essere aperto da varie finestre di dialogo dei due editor.
- Non è possibile avere il Symbol Manager aperto sia nel Program Editor che nell'Operator Interface Editor.

La sottostante figura presenta un esempio del Symbol Manager. La descrizione degli I/O e dei simboli in memoria è visualizzata sotto forma di tabella. Prima di ciascun simbolo compaiono una G per i simboli globali o una L per i simboli locali. Se il nome di un simbolo è preceduto da un asterisco, si tratta di un puntatore. Se il nome del simbolo corrisponde a un punto di I/O, nella colonna appropriata compaiono le informazioni relative alla scheda.



Controlli del Symbol Manager

Controllo	Descrizione
Local	Se sono selezionate, la lista visualizzata include il corrispondente tipo di simboli.
Global	Se sono selezionate, la lista visualizzata include il corrispondente tipo di simboli.
Board Info	Se è selezionata, sono visualizzate le informazioni relative alla scheda (scheda di interfaccia, rack, ecc.).
List Types	Filtra la lista dei simboli in base al tipo di dato e al tipo di blocco funzionale.
Add Local	Crea e aggiunge un simbolo locale.
Add Global	Crea e aggiunge un simbolo globale.
Copy Symbol	Copia il simbolo attualmente selezionato.
Delete Symbol	Cancella il simbolo attualmente selezionato.
Close	Chiude il Symbol Manager.
Apply	Rende le modifiche eseguite visibili all'esterno del Symbol Manager..
Help	Visualizza l'aiuto del Symbol Manager..
Print Locals	Stampa una lista di simboli locali.
Print Globals	Stampa una lista di simboli globali.
Local User Types	Crea un tipo definito dall'utente, locale.
Global User Types	Crea un tipo definito dall'utente, globale.

Creazione di un simbolo

Per creare un nuovo simbolo, cliccare su *Add Local* o *Add Global*, oppure usare il menu contestuale della lista di simboli. Comparirà la finestra di dialogo *Symbol Details*. Introdurre in questa finestra le informazioni appropriate e cliccare su *OK*. Il nuovo simbolo sarà aggiunto alla lista.

Se non è aperto un programma, è possibile creare soltanto simboli globali. Se è aperto un programma è possibile creare simboli globali o locali. Un simbolo locale può essere usato solo nel programma aperto (attivo) al momento della creazione del simbolo.

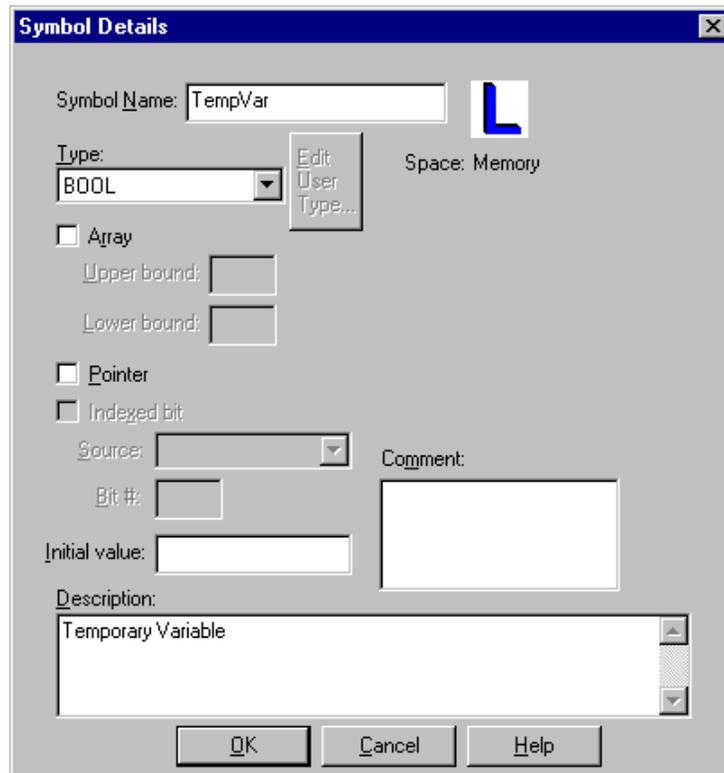
Nota

Terminata l'editazione dei simboli, non si dimentichi di cliccare su *Apply* per rendere le modifiche visibili all'esterno del *Symbol Manager*.

Editazione di un simbolo

1. Selezionare il simbolo che si desidera editare e premere *Invio*, oppure fare doppio click sul simbolo.
2. Comparire la finestra di dialogo *Symbol Details*.
3. Eseguire le modifiche e cliccare su *OK* per convalidarle.

Finestra di dialogo Symbol Details



Campo	Descrizione
Symbol Name	Il nome del simbolo può essere qualsiasi identificatore valido.
Type	Questo è il tipo di dato, elementare o definito dall'utente, o il tipo di blocco funzionale del simbolo. Per ulteriori informazioni fare riferimento a "Tipi di dati".
Edit User Type	Se il simbolo è di un tipo definito dall'utente, questo pulsante è abilitato ed è possibile editare il tipo. Per ulteriori informazioni fare riferimento a "Editazione dei tipi di dati definiti dall'utente".
Array	Se è selezionata, questa casella di controllo definisce il simbolo come un array. Fornire i limiti inferiore e superiore dell'array. Fare riferimento ad "Array" per ulteriori informazioni.
Pointer	Se è selezionata, questa casella di controllo definisce il simbolo come un puntatore. Fare riferimento a "Puntatori" per ulteriori informazioni.
Indexed Bit	E' possibile fare riferimento a uno specifico bit di un simbolo di tipo BYTE, WORD o DWORD. Fare riferimento a "Denominazione di un bit in un simbolo".
Initial Value	Questo campo si usa per assegnare al simbolo un valore iniziale. Un simbolo locale assume il proprio valore iniziale ogni volta che viene eseguito il programma. Un simbolo globale assume il proprio valore iniziale ogni volta che viene attivata la configurazione.
Comment	Una breve descrizione che compare nel Symbol Manager.
Description	La descrizione non compare nel Symbol Manager, essa compare solo in questa finestra di dialogo.

Copia di un simbolo

Con il comando *Copy Symbol* è possibile creare rapidamente un nuovo simbolo con le stesse proprietà di un simbolo esistente.

1. Selezionare il simbolo da copiare.
2. Cliccare sul pulsante *Copy Symbol* o usare il menu contestuale. Si aprirà la finestra di dialogo *Symbol Details* con tutti i parametri riempiti come nel simbolo selezionato.
3. Deve ora essere introdotto il nome del nuovo simbolo. Digitare questo nome, eseguire le altre modifiche eventualmente necessarie e cliccare su *OK*.

Nota

Terminata l'editazione dei simboli, non dimenticare di cliccare su *Apply* per rendere le modifiche visibili all'esterno del *Symbol Manager*.

Cancellazione di un simbolo

Selezionare il simbolo che si desidera cancellare e cliccare sul pulsante *Delete Symbol*, oppure usare il menu contestuale.

Denominazione di un bit in un simbolo

La prestazione di indicizzazione dei bit permette di fare riferimento a una specifica bit appartenente ad un simbolo di tipi *BYTE*, *WORD* o *DWORD*. E' possibile indicizzare un simbolo globale solo con un bit globale e un simbolo locale con un bit locale.

1. Aprire il *Symbol Manager* e cliccare su *Add Global* o *Add Local*. Comparirà la finestra di dialogo *Symbol Details*.
2. Inserire il nome del bit nel campo *Symbol Name*.
3. Nel campo *Type* selezionare il tipo di dato *BOOL*.
4. La casella di controllo *Indexed Bit* è abilitata solo se è già stato definito un simbolo di tipo *BYTE*, *WORD* o *DWORD*. Selezionare la casella *Indexed Bit*. I campi *Source* e *Bit #* diventano attivi.
5. La lista *Source* presenta tutti i simboli di tipo *BYTE*, *WORD* e *DWORD* già definiti (globali o locali). In questa lista, selezionare il simbolo che si desidera indicizzare.
6. Introdurre il numero del bit nel campo *Bit #*. I valori validi sono 0—7 per un byte, 0—15 per una parola e 0—31 per una doppia parola.
7. Introdurre la descrizione facoltativa nel campo *Description*.
8. Cliccare su *OK*. Il bit indicizzato comparirà nella lista dei simboli.

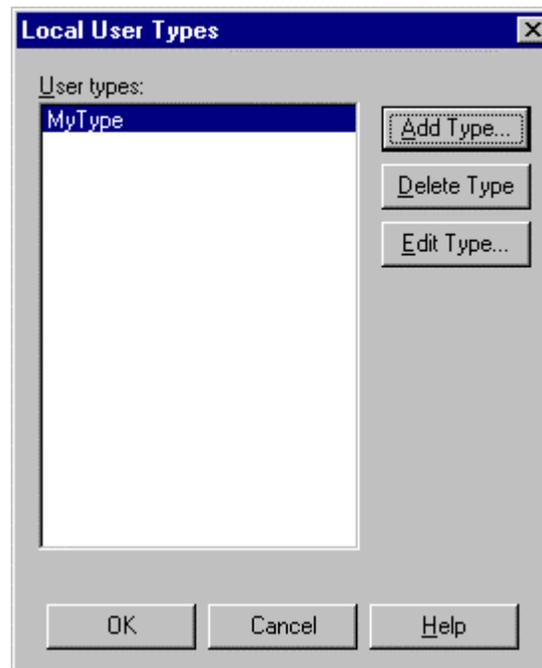
Editazione dei tipi di dati definiti dall'utente

L'utente può definire un tipo di dati. Un tipo di dati definito dall'utente è costituito da un gruppo di dati non necessariamente dello stesso tipo (interi, reali, stringhe, ecc.), che vengono trattati nel loro insieme. Ad esempio, può essere definito un tipo di dati chiamato Report e costituito da tre membri chiamati Year, Task e ItemNumbers. Year e ItemNumbers sono interi e Task è una stringa. Da quel momento qualsiasi programma potrà usare un simbolo di tipo Report il cui contenuto sarà costituito da tre membri che rappresenteranno l'anno, il nome di un task e un numero.

I tipi di dati definiti dall'utente possono essere globali o locali. Un tipo di dati locale può essere usato solo nel programma aperto (attivo) al momento della sua definizione.

Per editare i tipi di dati definiti dall'utente:

1. Aprire il Symbol Manager.
2. Cliccare su *Local User Types* o *Global User Types*. Comparirà la corrispondente finestra di dialogo *User Types*. Questa finestra contiene i tipi di dati definiti dall'utente (globali o locali) e i pulsanti che permettono di aggiungere, modificare e cancellare i tipi utente.



3. Procedere come segue:

- Per creare un nuovo tipo, cliccare su *Add Type*. Comparirà la finestra di dialogo *Edit User Type*.
- Per editare un tipo, selezionare il nome del tipo e cliccare su *Edit Type*. Comparirà la finestra di dialogo *Edit User Type*..
- Per cancellare un tipo, selezionare il nome del tipo e cliccare su *Delete Type*.



Campo	Descrizione
Type Name	Nome del tipo utente. Nel caso dell'aggiunta di un nuovo tipo, inserirne il nome..
Ordered member list	Elenco dei membri del tipo.
Add Member	Aggiunge al tipo un nuovo membro. Visualizza una finestra di dialogo <i>Edit Type Member</i> nella quale è possibile assegnare al nuovo membro un nome e un tipo di dato.
Delete Member	Cancella il membro selezionato.
Edit Member	Edita il membro selezionato. Visualizza una finestra di dialogo <i>Edit Type Member</i> nella quale è possibile assegnare al membro un nome e un tipo di dato
Move Up	Spostano i membri selezionati verso l'alto o verso il basso
Move Down	nella lista ordinata.

Uso dei simboli

E' possibile inserire manualmente un simbolo nel punto in cui serve digitandone il nome. Tuttavia è più facile aggiungerli usando le liste di simboli visualizzate (ad esempio nell'editor RLL) o trascinandoli dal Symbol Manager (documenti in testo strutturato o lista di istruzioni).

Trascinamento

La funzione di trascinamento del Symbol Manager permette di:

- trascinare un simbolo booleano su un rung RLL. Viene chiesto se devono essere aggiunti un contatto o una bobina. A seconda della scelta eseguita compaiono le finestre di dialogo standard *Edit Contact* o *Edit Coil*;
- trascinare un simbolo booleano su un contatto o su una bobina RLL esistenti. Viene chiesto di confermare la sostituzione del contatto o della bobina esistenti con quello o quella trascinati dal Symbol Manager;
- trascinare uno o più simboli dal Symbol Manager direttamente nella posizione voluta di un documento in testo strutturato o lista di istruzioni.

Enumerazioni

Quando viene trascinato dal Symbol Manager a un documento in testo strutturato (ad esempio) un simbolo costituito da più membri (strutture, blocchi funzionali e oggetti di sistema), compare la lista dei membri del simbolo selezionato. E' quindi possibile trascinare il membro che si desidera usare prelevandolo da questa lista. Questo funziona anche quando si devono aggiungere simboli (costituiti da più elementi) alla finestra di controllo, descritta a pagina 4-102.

Simboli di sistema

Simboli di sistema predefiniti

Il software di sistema crea automaticamente i seguenti simboli, che possono essere usati nei programmi applicativi.

Simbolo	Descrizione
TODAY	Contiene la data corrente del sistema. Il simbolo di sistema TODAY è un simbolo di tipo DATE contenente la data corrente del sistema e può essere usato per determinare la data di un evento. Con il simbolo TODAY è possibile usare i seguenti operatori: EQ, LT, GT, LE, GE e NE. Usare la frase di assegnazione o il comando MOVE per definire un valore per TODAY. Usare il comando ADD per sommare un periodo di tempo a TODAY.
NOW	Contiene l'ora corrente del sistema. Il simbolo di sistema NOW è un simbolo di tipo TOD contenente l'ora corrente del sistema e può essere usato per determinare l'ora di un evento. Con il simbolo NOW è possibile usare i seguenti operatori: EQ, LT, GT, LE, GE e NE. Usare la frase di assegnazione o il comando MOVE per definire un valore per NOW. Usare il comando ADD per sommare un periodo di tempo a NOW.
NULL	Questo simbolo si usa per assegnare il valore nullo a un puntatore o per confrontare un puntatore (uguale/non uguale) con un valore nullo.
Variabili TMR	Queste variabili contengono informazioni di stato per il tipo di dati TMR.
Variabili contatori	Queste variabili contengono informazioni di stato per i contatori RLL (CTD, CTU e CTUD). Questi simboli sono locali al programma applicativo.
Variabili temporizzatori	Queste variabili contengono informazioni di stato per i temporizzatori RLL (TOF, TON e TP). Questi simboli sono locali al programma applicativo.
"nomepasso".X	Contiene lo stato (attivo/non attivo) di un passo SFC. Questi simboli sono locali al programma applicativo.
"nomepasso".T	Contiene il tempo di esecuzione trascorso di un passo SFC. Questi simboli sono locali al programma applicativo.
Motion Control	Queste variabili contengono informazioni di stato per asse, gruppo di variabili asse, controllo programma e mandrino.
Blocco controllo file	Queste variabili contengono informazioni di stato per le operazioni su file.
Controllo programma	Queste variabili contengono informazioni di stato per il tipo di dati PRGCB.

Simboli di Run-Time

I seguenti simboli sono creati automaticamente dal software di sistema. Essi sono accessibili dal Symbol Manager o dalla finestra di controllo e possono essere usati nei programmi applicativi dell'utente.

Nome del simbolo	Descrizione
RT_ERROR	(INT) Errori matematici: 0 = nessun errore 1 = divisione per zero 2 = radice quadrata di un numero negativo RT_ERROR deve essere azzerato dall'utente.
RT_FIRST_SCAN	(BOOL) è TRUE durante la prima scansione del primo programma eseguito nel motore di runtime di PC Control. Dopo la chiusura di tutti i programmi, RT_FIRST_SCAN viene nuovamente messo a TRUE per la prima scansione del primo programma eseguito.
RT_SCAN_OVERRUN	(BOOL) è TRUE quando la scansione degli I/O + la scansione della logica eccede l'intervallo di scansione.
RT_MAX_SCAN	(REAL) durata in millisecondi della massima scansione del motore di runtime.
RT_LAST_SCAN	(REAL) durata in millisecondi dell'ultima scansione del motore di runtime.
RT_AVG_SCAN	(REAL) durata in millisecondi della scansione media del motore di runtime (scansione di runtime = logica + I/O + overhead). Media mobile delle ultime 100 scansioni.
RT_LOGIC_MAX	(REAL) durata in millisecondi della massima scansione della logica.
RT_LOGIC_LAST	(REAL) durata in millisecondi dell'ultima scansione della logica.
RT_LOGIC_AVG	(REAL) durata in millisecondi della scansione media della logica. Media mobile delle ultime 100 scansioni.
RT_IO_MAX	(REAL) durata in millisecondi della scansione massima degli I/O.
RT_IO_LAST	(REAL) durata in millisecondi dell'ultima scansione degli I/O.
RT_IO_AVG	(REAL) durata in millisecondi della scansione media degli I/O. Media mobile delle ultime 100 scansioni.
RT_MEM_PCT	(REAL) percentuale residua della RAM di sistema (heap) allocata al software del sistema di comando.
RT_LOW_BATTERY	(BOOL) segnale di batteria scarica dall'UPS.
RT_POWER_FAIL	(BOOL) segnale di caduta dell'alimentazione dall'UPS.
RT_SCAN_RATE	(REAL) intervallo di scansione (in millisecondi) specificato nella configurazione attiva.

Parole riservate

Gli identificatori listati nella tabella che segue sono simboli di sistema riservati. Questi identificatori non possono essere assegnati a simboli creati dall'utente. Notare che tutti i simboli di sistema usano lettere maiuscole. Questa lista può cambiare nelle prossime edizioni del software. Per evitare conflitti futuri, è opportuno che i simboli creati dall'utente contengano lettere maiuscole e minuscole.

Parole riservate

ABORT_ALL	ABS	AC
ACCEL	ACOS	ACTION
ADD	ADD_NOFLY	AND_SLOWLY
AND	AND_BITS	ANDN
ANDT	ANDTN	ANY
ANY_BIT	ANY_DATE	ANY_INT
ANY_NUM	ANY_REAL	APPENDFILE
ARRAY	ARRAY_TO_STRING	AS
ASIN	AT	ATAN
AXIS	AXISGRP	AXSJOG
BCD_TO_INT	BEGIN	BEGIN_IL
BEGIN_RS274	BOOL	BREAK
BY	BYTE	CASE
CAL	CALC	CALCN
CD	CLK	CLOSEFILE
CONCAT	CONFIGURATION	CONSTANT
COPYFILE	COS	CTD
CTU	CTUD	CU
CV	D	DATE
DATE_AND_TIME	DELETE	DELETEFILE
DINT	DIV	DO
DS	DSPMSG	DT
DWORD	ELSE	ELSEIF
EN	END	END_ACTION
END_CASE	END_CONFIGURATION	END_FOR
END_FOR_NOWAIT	END_FUNCTION	END_FUNCTION_BLOCK
END_IF	ENDIF	END_IL
END_RS274	ENO	END_PROGRAM
END_REPEAT	END_RESOURCE	END_STEP
END_STRUCT	END_TRANSITION	END_TYPE
END_VAR	END_WHILE	END_WHILE_NOWAIT
EQ	ESTOP	ET
EXIT	EXP	EXPT

Parole riservate

F	FALSE	FB
F_EDGE	FILE	FIND
F_EDGE	FNAME	FOR
FROM	F_TRIG	FTYPE
FUNCTION	FUNCTION_BLOCK	G
GE	GLOBAL	GOTO
GT	H	I
IF	IN	IN1
IN2	INCLUDE	INIT
INITIAL_STEP	INSERT	INT
INTERVAL	INT_TO_BCD	INT_TO_REAL
INT_TO_STRING	IP	J
JMP	JMPC	JMPCN
JOGCONT	JOGDIR	JOGDIST
JOGHOME	JOGINCR	JOGMINUS
JOGPLUS	JOGSPD	JOGTYPE
K	L	LD
LDN	LDT	LE
LEFT	LEN	LIMIT
LINT	LL	LN
LOG	LREAL	LT
LWORD	M	MACROSTEP
MAX	MC	MID
MIN	MOD	MOVE
MOVEAXS	MS	MSGWND
MUL	MULP	MUX
N	NAME	NE
NEWFILE	NIL	NOT
NOW	NT	NULL
		OF
ON	OPENFILE	OR
OR_BITS	ORN	ORT
ORTN	OUT	P
PID	POSTN	POW
PRGCB	PRIORITY	PROGRAM
PT	PV	PW
Q	QU	QD
R	R1	READFILE
READ_ONLY	READ_WRITE	REAL
REAL_TO_STRING	R_EDGE	REPEAT
REPLACE	RESET_ESTOP	RESOURCE
RET	RETC	RETCN

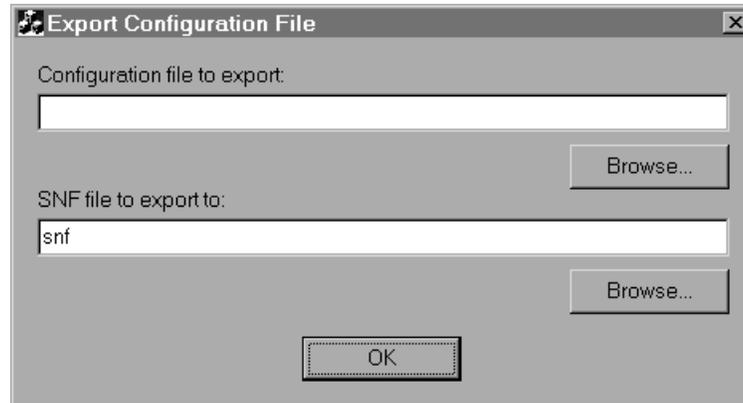
Parole riservate

RETAIN	RETURN	REWINDFILE
RIGHT	ROL	ROR
RS	RTC	R_TRIG
RUNG	S1	SCAN
SD	SEL	SEMA
SET_LOADSIZE	SHL	SHR
SIN	SINT	SL
SR	SQRT	ST
STT	STN	STTN
STEP	STOPJOG	STRING
STRING_TO_ARRAY	STRUCT	SUB
T	TAN	TASK
THEN	TIME	TIME_OF_DAY
TMR	TO	TOD
TODAY	TOF	TON
TP	TRANS	TRANSITION
TRUE	TRUNC	TYPE
UDINT	UL	ULINT
UNTIL	USINT	VAR
VAR_ACCESS	VAR_EXTERNAL	VAR_GLOBAL
VAR_INPUT	VAR_IN_OUT	VAR_OUTPUT
VEL	WHILE	WRITEFILE
WITH	WORD	XOR
XOR_BITS	XORN	XTON
XTOF	XTP	ZZZZ

Esportazione di simboli per CIMPLICITY HMI

I simboli possono essere esportati in formato Shared Name File (.snf), che è una forma di CSV specifica dei PLC GE Fanuc.

Per esportare il file dei simboli, scegliere **Export Symbols for CIMPLICITY HMI** nel menu **Tools** del **Program Editor**. Comparirà la finestra di dialogo **Export Configuration File**.



Usare i pulsanti **Browse** per selezionare il file di configurazione (.cfg) origine e il percorso del file .snf di destinazione.

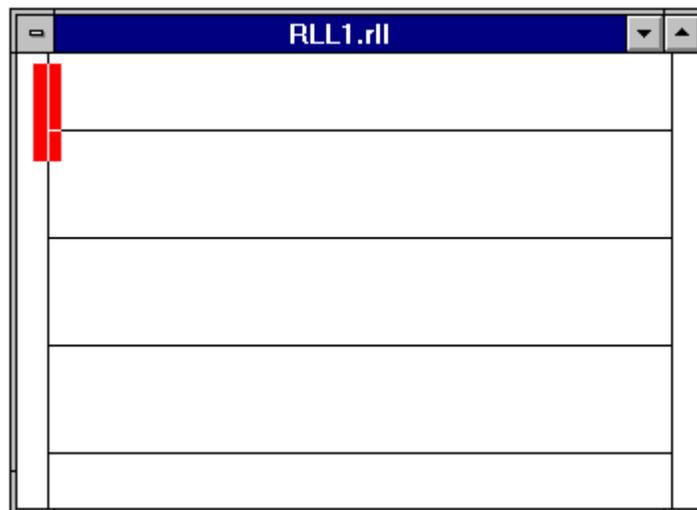
Per la descrizione dettagliata del formato dei file .snf, fare riferimento al *CIMPLICITY HMI for Windows NT and Windows 95 Base System User's Manual*, GFK-1180.

Sezione 2: Programmazione RLL

Descrizione generale dei diagrammi RLL (Relay Ladder Logic)

I diagrammi RLL sono usati comunemente sui controllori programmabili per costruire programmi a logica continua. Un diagramma RLL assomiglia allo schema elettrico di un circuito logico a relè equivalente.

Un diagramma RLL contiene due linee verticali rappresentanti le barre di alimentazione. La barra di sinistra è una sorgente di corrente elettrica eccitata ogni volta che il programma RLL è in esecuzione. La barra di destra assorbe il flusso di corrente proveniente dalla barra di sinistra. Le due barre sono collegate da linee orizzontali (che ricordano i pioli di una scala (ladder), in inglese rung) sulle quali sono posizionate le istruzioni che formano la logica del programma.



Le istruzioni base dei programmi RLL, contatti e bobine, rappresentano sia componenti hardware (fine corsa, bobine di solenoidi, spie, ecc.) sia singoli bit della memoria interna.

I contatti di un diagramma RLL corrispondono a simboli booleani (variabili) in ingresso. Un contatto può essere:

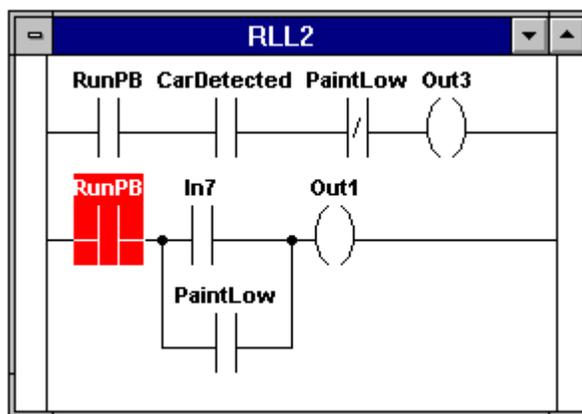
- normalmente aperto (alto = attivo): la corrente passa ed eccita l'elemento RLL alla sua destra quando il simbolo associato al contatto è alto (1 o ON)
- normalmente chiuso (basso = attivo): la corrente passa ed eccita l'elemento RLL alla sua destra quando il simbolo associato al contatto è basso (0 o OFF)

Normalmente, sopra all'elemento grafico che rappresenta questi ingressi compare il nome del simbolo booleano che essi rappresentano. Esistono vari tipi

di contatti di ingresso, inclusi i sensori di transizione, che passano la corrente all'elemento successivo solo per una scansione del programma RLL.

Le bobine di un diagramma RLL rappresentano variabili booleane in uscita. Se la logica a sinistra di una bobina è eccitata (ha valore true), il simbolo booleano rappresentato dalla bobina riceve un 1 booleano (alto); altrimenti riceve uno 0 booleano (basso). Esistono vari tipi di bobine, che includono le bobine invertenti e le bobine impulsive (uscite per una sola scansione). Alcune bobine sono di tenuta (latching) e si limitano a mettere a true (o a false) la variabile in uscita, se la logica alla loro sinistra è true.

Gli elementi della logica RLL che risiedono sullo stesso rung sono tra loro in AND logico. Per facilitare la programmazione di operazioni complesse, che usano più di un simbolo booleano, è possibile ricorrere ai blocchi funzionali.

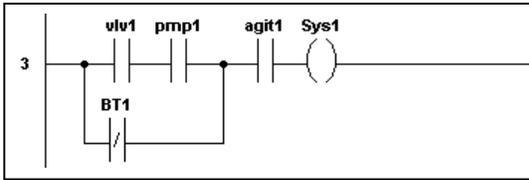


Come vengono risolti i programmi applicativi RLL

Questa sezione spiega come vengono risolti i programmi RLL che usano semplici relè e quelli che usano i blocchi funzionali.

Come viene risolta la semplice logica a relè

Dopo aver scritto le uscite fisiche, il sistema legge gli ingressi fisici e poi risolve la logica RLL. Il flusso di corrente e la soluzione della logica del programma procedono sempre dall'alto verso il basso e da sinistra verso destra. Nel seguente esempio, il flusso di corrente inizia dalla barra di sinistra. Se il contatto vlv1 è on, il flusso continua fino al contatto pmp1. vlv1, pmp1 e agit1 sono in serie e rappresentano l'AND logico dei tre contatti. Il contatto BT1 è in parallelo con i contatti vlv1 e pmp1. Questo rappresenta l'OR logico di BT1 con vlv1 e pmp1. Se il contatto BT1 è on, il flusso di corrente arriva a agit1 anche se vlv1 e pmp1 non sono on. Se il flusso di corrente raggiunge la bobina Sys1, attiva il circuito, completando il flusso di corrente fino alla barra di destra.

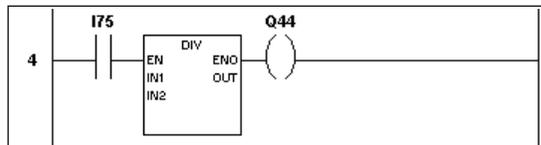


Come viene risolta la logica RLL quando si usano i blocchi funzionali

I blocchi funzionali forniscono un meccanismo per la soluzione di operazioni più complesse, non facilmente gestite mediante contatti e bobine, quali:

- operazioni matematiche
- funzioni logiche
- contatori e temporizzatori, ecc.

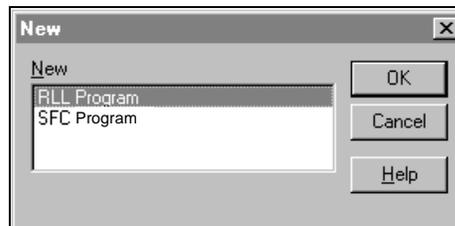
Un blocco funzionale riceve la corrente dal rung attraverso il suo ingresso e la trasferisce all'elemento successivo del rung attraverso la sua uscita. Un blocco funzionale può anche leggere e scrivere dati mediante i suoi ingressi e uscite interni. Nell'esempio che segue, il blocco funzionale divisione (DIV) è abilitato dal suo ingresso di rung EN, che riceve corrente quando è abilitato il contatto I75. Il blocco funzionale riceve il dividendo e il divisore tramite due ingressi interni (IN1, IN2), scrive il quoziente su un'uscita interna (OUT) e trasferisce il flusso di corrente all'elemento successivo del programma, Q44, attraverso la sua uscita di rung (ENO).



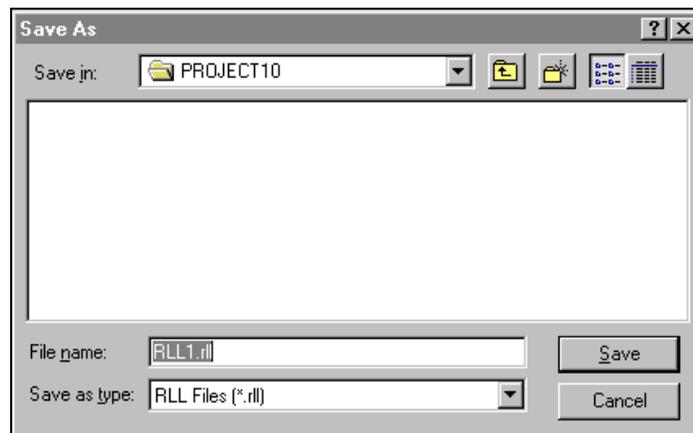
Creazione di un programma RLL

Per aprire l'editor RLL e iniziare la creazione di un programma, si deve entrare nell'ambiente di sviluppo programmi.

1. Avviare il Program Editor.
2. Cliccare su *File* e selezionare *New Editor*. Comparirà la finestra di dialogo *New*.



3. Selezionare *RLL Program* e cliccare su *OK*. Comparirà la finestra di dialogo *Save As*.

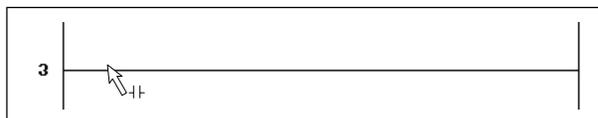


4. Scegliere il nome per difetto o un nuovo nome per il programma e cliccare su *Save*. L'estensione *.rll* è appesa quando il file viene salvato nel progetto. L'editor RLL visualizza un nuovo file RLL con le due barre di alimentazione e un rung.

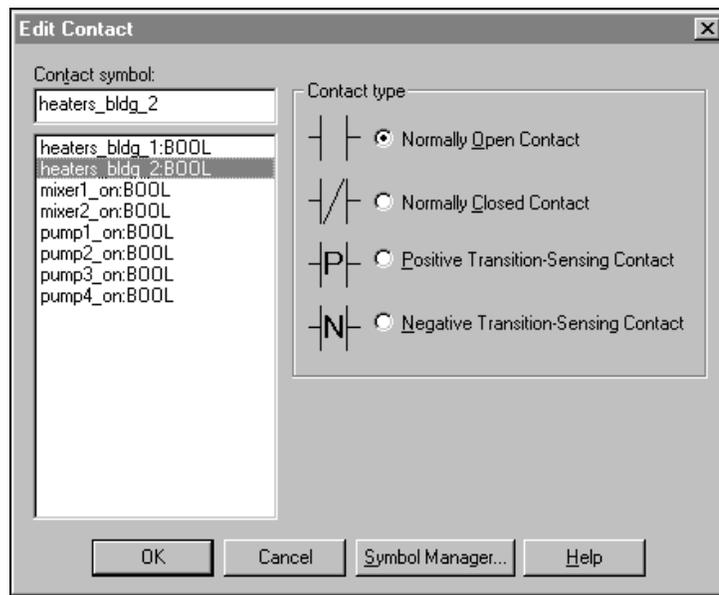
Aggiunta di elementi al programma

Aggiunta di un contatto

1. Cliccare sul pulsante contatto della barra degli strumenti RLL. Il cursore cambia forma per rappresentare il contatto.
2. Portare il cursore nel punto del rung nel quale si desidera inserire il nuovo contatto.



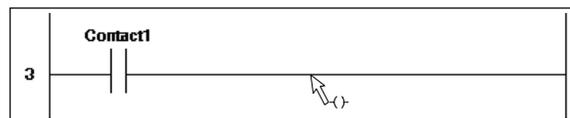
3. Fare click con il pulsante sinistro del mouse. Comparirà la finestra di dialogo **Edit Contact**.



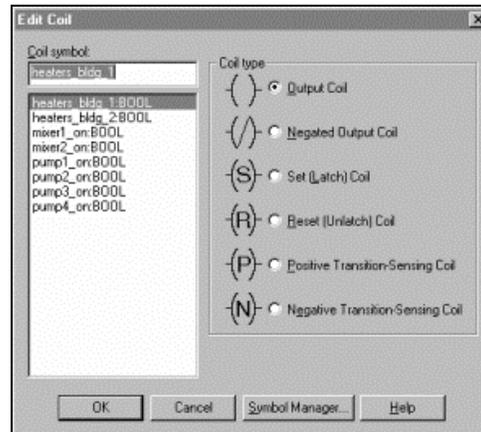
4. Se i nomi dei simboli sono già stati definiti, cliccare sul simbolo che rappresenta il contatto (nella figura qui sopra, heaters_bldg_2).
5. Se i simboli non sono ancora stati definiti, o se si desidera definire un nuovo simbolo, è necessario accedere al Symbol Manager e compilare la scheda del contatto.
6. Selezionare il tipo di contatto (normalmente aperto, Normalmente chiuso, ecc.) e cliccare su **OK**. Il nuovo contatto viene inserito nel rung, nella posizione specificata.

Aggiunta di una bobina

1. cliccare sul pulsante bobina della barra degli strumenti RLL. Il cursore cambia forma per rappresentare la bobina.
2. Portare il cursore nella posizione del rung nella quale si desidera inserire la bobina.



3. Fare click con il pulsante sinistro del mouse. Comparirà la finestra di dialogo **Edit Coil**.

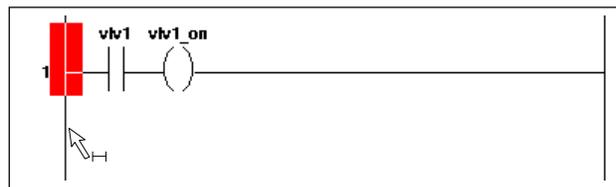


4. Se i nomi dei simboli sono già stati definiti, cliccare sul simbolo che rappresenta la bobina.
5. Se i simboli non sono ancora stati definiti, o se si desidera definire un nuovo simbolo, è necessario accedere al Symbol Manager e compilare la scheda della bobina.
6. Selezionare il tipo di bobina (uscita, uscita negata, ecc.) e cliccare su *OK*. La nuova bobina viene inserita nel rung, nella posizione specificata..

E' possibile inserire una bobina in qualunque punto del rung, anche a sinistra di un ingresso o in un ramo OR. Una bobina memorizza il risultato della logica valutata fino alla sua posizione nel rung.

Inserimento di un nuovo rung

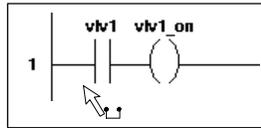
1. Cliccare sul pulsante Rung della barra degli strumenti RLL. La forma del cursore cambia per rappresentare il rung.
2. Portare il cursore sulla posizione della barra di sinistra nella quale deve essere inserito il rung.



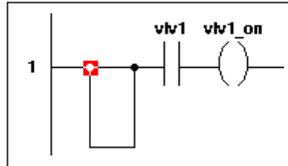
3. Cliccare con il pulsante sinistro del mouse. L'editor inserisce il rung nella posizione specificata.

Aggiunta di un ramo

1. Cliccare sul pulsante ramo della barra degli strumenti RLL. La forma del cursore cambia per rappresentare il ramo.
2. Portare il cursore nella posizione del rung nella quale deve essere inserito il ramo.



3. Cliccare con il pulsante sinistro del mouse. L'editor inserisce il ramo nella posizione specificata.



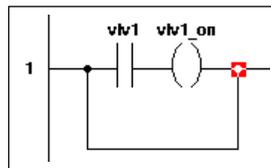
4. Dopo aver inserito il ramo, è possibile regolare i punti di contatto in base alle necessità.

Per spostare i punti di contatto di un ramo:

1. Cliccare sullo strumento di selezione.



2. Portare il cursore sul punto di contatto che si desidera spostare e premere il pulsante sinistro del mouse.
3. Tenendo premuto il pulsante del mouse, muovere il cursore, e il punto di contatto, lungo il rung fino a portare il punto di contatto nella posizione desiderata, e rilasciare il pulsante del mouse. L'editor collega il ramo al rung nella nuova posizione.

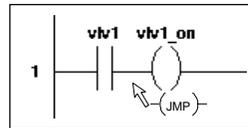


E' possibile inserire un ramo vuoto (una derivazione). Una derivazione può essere utile per disabilitare temporaneamente una sezione della logica senza cancellarla dal programma. Usta insieme ad un contatto che taglia il flusso di corrente per la logica in questione, la derivazione mantiene il flusso di corrente per il resto del rung. Questa possibilità è soprattutto utile per la messa a punto del programma.

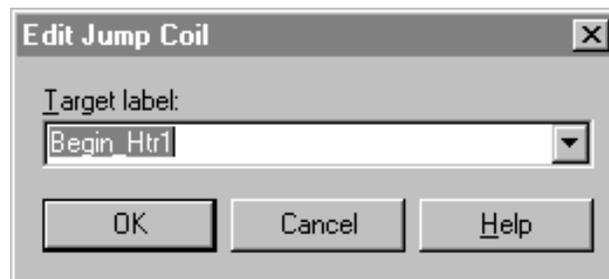
E' anche possibile spostare un punto di contatto da un rung ad un altro, senza cancellare la logica contenuta nel ramo.

Aggiunta di una bobina di salto

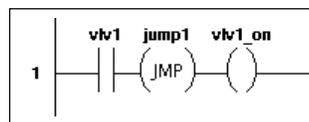
1. Cliccare sul pulsante *Jump* della barra degli strumenti. Il cursore cambia forma per indicare la bobina di salto.
2. Portare il cursore nella posizione del rung nella quale si desidera inserire il salto.



3. Cliccare con il pulsante sinistro del mouse. Comparirà la finestra di dialogo **Edit Jump Coil**.



4. Introdurre l'etichetta di destinazione e cliccare su *OK*. Un'etichetta di destinazione non può contenere spazi. L'editor inserisce il salto nella posizione specificata.



Aggiunta di una bobina di transizione SFC

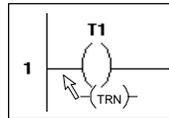
La bobina di transizione SFC è un elemento del linguaggio RLL che può essere usato solo in specifiche condizioni in un programma SFC (all'interno di una azione SFC).

Nota

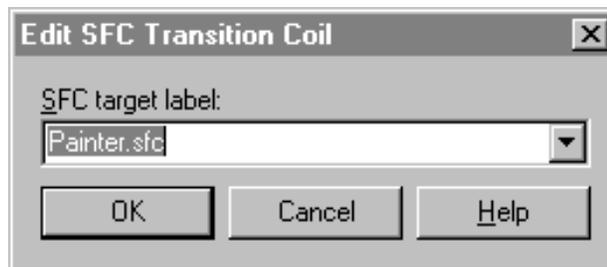
L'aggiunta al programma di una bobina di transizione SFC è possibile solo durante l'editazione di una azione in un programma SFC.

Per aggiungere una bobina di transizione SFC:

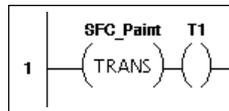
1. Cliccare sul corrispondente pulsante della barra degli strumenti RLL. Il cursore cambia forma per indicare la bobina di transizione.
2. Portare il cursore nella posizione del rung nella quale si desidera inserire la bobina di transizione SFC.



3. Cliccare con il pulsante sinistro del mouse. Comparirà la finestra di dialogo **Edit SFC Transition Coil**.



4. Digitare il nome della destinazione SFC e cliccare su **OK**. Un nome SFC non può contenere spazi. L'edito inserisce la bobina di transizione SFC nella posizione specificata.

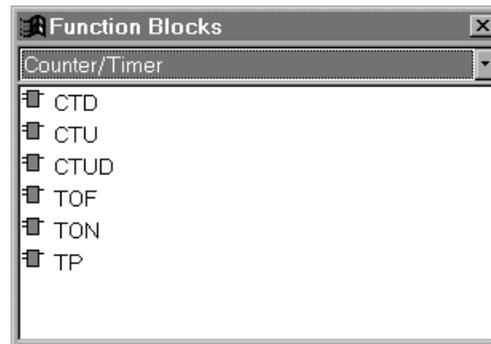


Aggiunta di blocchi funzionali

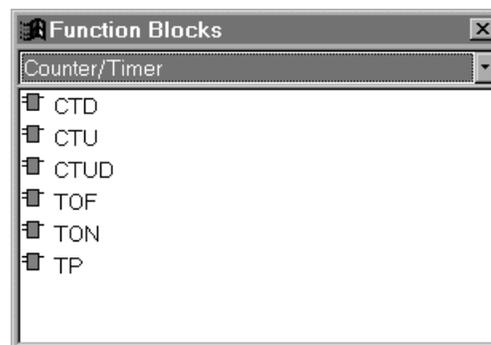
Esistono vari algoritmi predefiniti detti blocchi funzionali e utilizzabili in un programma RLL. I blocchi funzionali sono abilitati da un ingresso dal rung, eseguono funzioni trigonometriche, matematiche o logiche, operazioni di scorrimento di bit, operazioni su file, ecc., e infine inviano il risultato ad un'uscita che alimenta un altro elemento del rung.

Per aggiungere un blocco funzionale:

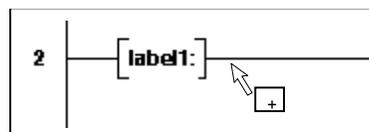
1. Se la paletta dei blocchi funzionali non è visualizzata, selezionare *View/Function Block Palette* dalla barra del menu. L'editor visualizzerà la paletta dei blocchi funzionali.



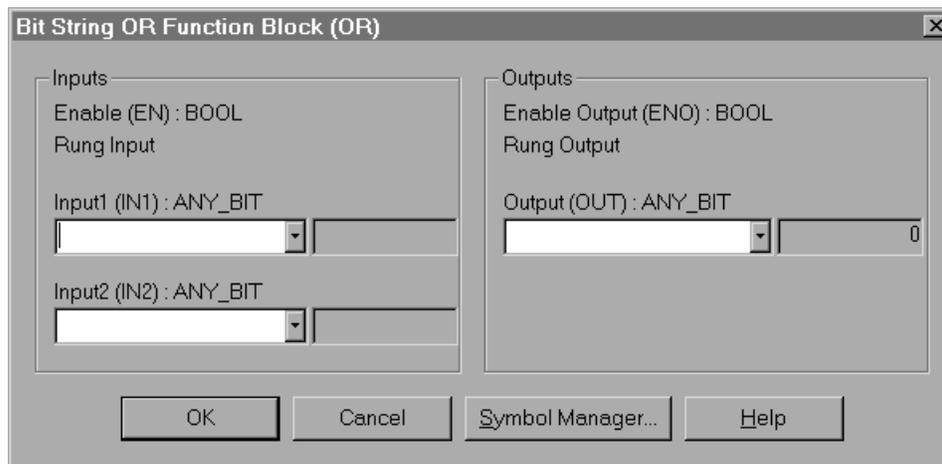
2. Selezionare il tipo di blocco funzionale che interessa nell'elenco a discesa. La paletta dei blocchi funzionali cambia per visualizzare i blocchi funzionali del tipo selezionato.



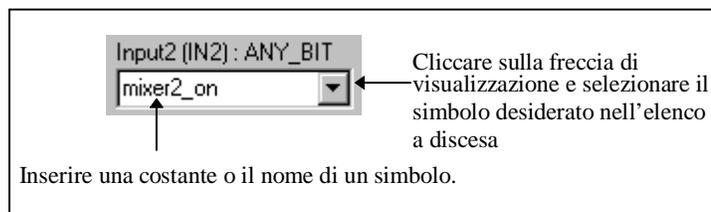
3. Cliccare sullo specifico blocco funzionale che si desidera inserire, quale, ad esempio, un blocco OR bit per bit.
4. Trascinare il blocco funzionale nella posizione del rung nella quale si desidera inserirlo.



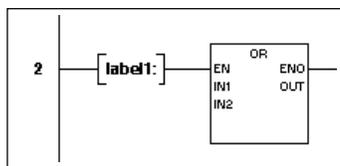
- Quando viene rilasciato il pulsante sinistro del mouse, compare la finestra di dialogo per il blocco funzionale prescelto.



- Inserire le informazioni appropriate.
- Per introdurre una costante (un intero, un numero reale, una stringa di caratteri, ecc.) digitarne il valore direttamente nel campo. Per introdurre un simbolo digitarne il nome nel campo o cliccare sulla freccia di visualizzazione e selezionare il simbolo nell'elenco dei simboli validi che compare.



- Dopo aver specificato i dati necessari, cliccare su *OK*. L'editor inserirà il blocco nella posizione specificata.



Spostamento ed editazione degli elementi del programma

Selezione degli elementi del programma

Per selezionare gli elementi con la tastiera:

Usare i tasti freccia per selezionare l'elemento successivo nella direzione voluta. Se la selezione non è possibile, viene dato un segnale acustico.

Per selezionare gli elementi con il mouse:

1. Cliccare sullo strumento di selezione nella barra degli strumenti RLL o nella barra degli strumenti SFC.



2. Portare il cursore sull'elemento desiderato e premere il pulsante sinistro del mouse. L'elemento sarà evidenziato con il colore selezionato a questo scopo.

Note speciali sulla selezione di un singolo elemento

Per selezionare una divergenza selettiva o una divergenza simultanea in un programma SFC, selezionare la barra superiore o la barra inferiore della divergenza.

Per selezionare un rung in un programma SFC, selezionare la barra di alimentazione destra o sinistra del rung.

Per selezionare contemporaneamente più elementi:

1. Cliccare sullo strumento di selezione nella barra degli strumenti RLL o nella barra degli strumenti SFC.



2. Portare il cursore a sinistra e più in alto dell'elemento di estrema sinistra del gruppo desiderato (il cursore non deve essere su un elemento).
3. Premere e tenere premuto il pulsante sinistro del mouse, trascinando il cursore verso destra e in basso. Durante il trascinamento del cursore è visualizzata una cornice di selezione.
4. Quando tutti gli elementi desiderati sono all'interno della cornice di selezione, rilasciare il pulsante sinistro del mouse. Tutti gli elementi interamente all'interno della cornice di selezione saranno selezionati.

Note speciali sulla selezione multipla

Per selezionare un connettore di ramo in un programma RLL, ambedue i connettori e tutti gli elementi del ramo devono essere all'interno della cornice di selezione.

Per selezionare un rung in un programma RLL, la barra di sinistra, la barra di destra e tutti gli elementi del rung devono essere all'interno della cornice di selezione.

Per selezionare un ciclo in un programma SFC, la freccia superiore del ciclo, tutte le transizioni del ciclo e tutti gli elementi contenuti nel ciclo devono essere all'interno della cornice di selezione.

Per selezionare un salto in un programma SFC, il diamante di salto, tutte le transizioni e tutte le etichette di destinazione devono essere all'interno della cornice di selezione.

Per selezionare una divergenza selettiva o una divergenza simultanea in un programma SFC, la barra superiore, la barra inferiore e tutti gli elementi della divergenza devono essere all'interno della cornice di selezione.

Spostamento di un ramo

1. Cliccare sullo strumento di selezione nella barra degli strumenti RLL o nella barra degli strumenti SFC..



2. Portare il cursore sul connettore di ramo desiderato. Premere e tenere premuto il pulsante sinistro del mouse.
3. Trascinare il connettore di ramo nella posizione desiderata (il cursore indicherà il connettore di ramo).
4. Rilasciare il pulsante sinistro del mouse per posare il connettore di ramo nella posizione desiderata.

Per cancellare l'operazione di trascinamento premere il tasto <ESCAPE>.

Se la posizione di destinazione del connettore di ramo è sullo stesso rung e non è all'interno di un ramo incluso o all'esterno di un ramo circostante, il connettore di ramo trascinato andrà nella nuova posizione e il connettore opposto resterà dove si trova. Invece, se la posizione di destinazione del connettore di ramo è su un altro rung, è all'interno di un ramo incluso o all'esterno di un ramo circostante, ambedue le estremità del ramo andranno nella nuova posizione (sarà spostato l'intero ramo).

Spostamento di elementi del programma

1. Selezionare l'elemento o gli elementi da spostare e portare il cursore su uno degli elementi selezionati.
2. Premere e tenere premuto il pulsante sinistro del mouse. Trascinare gli elementi selezionati nella posizione desiderata. All'inizio del trascinamento gli elementi selezionati scompaiono e il cursore cambia forma per rappresentare l'elemento trascinato. Se sono trascinati più elementi, il cursore assume la forma che indica il trascinamento di un gruppo. Rilasciare il pulsante sinistro del mouse per posare l'elemento, o gli elementi, nella posizione desiderata.

Per cancellare l'operazione di trascinamento premere il tasto <ESCAPE>.

Editazione di elementi del programma

1. Cliccare sullo strumento di selezione.
2. Fare doppio click sull'elemento (Passo, transizione, etichetta, ecc.). Comparirà la finestra di dialogo corrispondente all'elemento prescelto (**Edit Step**, **Select RLL Transition Logic**, **Bypass Jump Transition Logic**, etc.).
3. Eseguire le necessarie modifiche in questa finestra di dialogo.

Cancellazione di un ramo

Per cancellare un oggetto, si seleziona quell'oggetto e poi si clicca sullo strumento Taglia. Tuttavia, per cancellare un ramo procedere come segue.

Per cancellare un ramo privo di elementi:

1. Cliccare sullo strumento di selezione.
2. Posizionare il cursore al centro del ramo e cliccare.
3. Cliccare sullo strumento Taglia per cancellare il ramo.

Per cancellare un ramo contenente uno o più elementi:

1. Cliccare sullo strumento di selezione.
2. Selezionare un'area che includa l'intero ramo con i suoi punti di connessione.
3. Cliccare sullo strumento Taglia per cancellare il ramo.

Annullamento/ripetizione delle operazioni

Per annullare un'operazione cliccare sul pulsante *Undo* della barra degli strumenti.

Documentazione dei programmi RLL

Un commento può essere una qualsiasi descrizione significativa visualizzata su un rung. E' possibile scegliere se visualizzare o nascondere i commenti.

Aggiunta e modifica di commenti di rung

1. Cliccare sullo strumento commenti della barra del menu RLL.



Se a un rung non è associato un commento, la stringa visualizzata è (***Rung Comment***).

2. Portare il cursore sul rung del quale si vuole editare il commento e fare doppio click. Comparirà la finestra di dialogo **Program Comments**.
3. Introdurre il commento e cliccare su OK. Il commento compare sul rung specificato.

Aggiunta e modifica di descrizioni di simboli

1. Aprire un programma RLL o SFC.
2. Cliccare sullo strumento Symbol Manager.



Comparirà la finestra di dialogo del Symbol Manager per i simboli locali o globali.

3. Aggiungere o modificare le informazioni come si desidera.

Sezione 3: Programmazione SFC

Descrizione generale dei diagrammi funzionali sequenziali

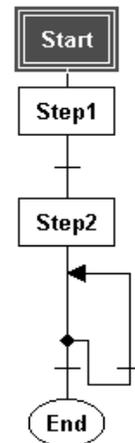
Un diagramma funzionale sequenziale rappresenta un programma come una serie di passi in sequenza. La logica di controllo è associata a questi passi, chiamati azioni. La logica contenuta in una azione viene eseguita quando il passo diventa attivo.

I passi sono collegati con linee orientate e il controllo va da un passo al successivo mediante transizioni. Una transizione può essere booleana oppure può essere un singolo rung RLL.

E' possibile gestire più percorsi di controllo tramite le divergenze. Una divergenza selettiva permette di scegliere il percorso attivo tra due o più percorsi alternativi; una divergenza simultanea permette di attivare più percorsi in parallelo.

Esistono altre possibilità di determinazione del flusso del programma. E' possibile includere cicli per la ripetizione di una serie di passi o trasferire il controllo a una certa posizione usando la struttura salto/etichetta.

E' possibile creare diagrammi funzionali sequenziali a più percorsi ed è possibile attivare più di un SFC per volta.



Che cosa sono i passi

Un passo rappresenta una condizione nella quale il comportamento del sistema segue un insieme di regole definite dalle azioni e dalle funzioni associate a quel passo. Un passo può essere attivo o inattivo. Lo stato del sistema in un certo momento è definito dall'insieme dei passi attivi e dai valori delle sue variabili interne e di uscita.

In un diagramma funzionale sequenziale, un passo è rappresentato graficamente con un rettangolo contenente il nome che identifica il passo. Il flusso del programma entra nel passo attraverso la linea verticale che raggiunge dall'alto il rettangolo che lo rappresenta e ne esce attraverso la linea verticale che lo lascia dal basso.

Quando viene creato un nuovo SFC, il sistema genera automaticamente:

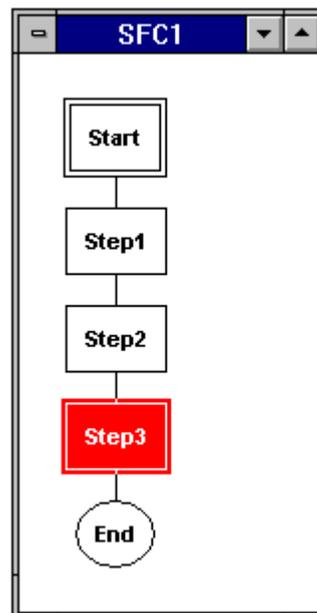
- il primo passo, etichettato Start,
- l'ultimo passo, etichettato End.

Non è possibile editare questi passi. Essi rappresentano semplicemente l'inizio e la fine del diagramma.

Tipicamente, i passi di un SFC sono separati da transizioni, che sono elementi del programma. Il software PC CONTROL, migliorando le specifiche IEC-1131-3, permette di programmare un passo immediatamente prima o immediatamente dopo un altro passo; durante l'esecuzione (runtime) il sistema inserisce la transizione fittizia richiesta.

A un passo SFC possono essere associate una o più azioni. Un'azione può contenere uno o più rung di logica a contatti e la sua esecuzione è controllata da speciali qualificatori di azione..

Dall'interno di un passo (passo macro) è possibile richiamare per l'esecuzione un intero SFC (il figlio). Completata l'esecuzione dello SFC figlio, il controllo del programma ritorna al passo macro che ha eseguito il richiamo.



Proprietà dei passi

Per creare un passo, ne devono essere configurate le proprietà nella finestra di dialogo Edit Step. Questa tabella sintetizza le proprietà configurabili:

Campo/pulsante:	Descrizione:
Motion/Process Commands	Contiene il codice programmato per il passo.
Structured Text	Selezione il linguaggio di programmazione Testo Strutturato.
RS-274D	Selezione il linguaggio di programmazione RS-274D. (Descritto nell'appendice E.)
Link File	Collega il passo a un file contenente il codice del programma.
Edit Linked File	Apri il file di codice collegato al passo e lo visualizza in un editor.
Symbol Manager	Accede al Symbol Manager.
Display (Cliccare su uno.)	
Step Name	Visualizza il nome del passo al suo interno.
Motion/Process Commands	Visualizza il codice (Testo strutturato o Motion Control) contenuto nel passo.
Step Description	Visualizza la descrizione del passo al suo interno.
Icon	Visualizza un'icona all'interno del passo.
Width	Stabilisce la larghezza in pixel della descrizione del passo.
Icon...	Accede alla paletta delle icone.
Remove	Cancella un'icona dal passo, se gliene era stata assegnata una.

Uso dei simboli di sistema del passo

Ciascun passo SFC ha due simboli di sistema (.X e .T). E' possibile usare questi simboli in qualsiasi espressione, contatto o bobina, al posto di un simbolo dello stesso tipo. A questi simboli ci si riferisce scrivendo il nome del passo seguito da un punto e dal suffisso corrispondente.

Simbolo	Definizione	Esempio
X	Passo attivo (booleano) Il simbolo X (passo attivo) è TRUE quando il passo è attivo ed è FALSE quando il passo è inattivo.	STEP1.X riferimento al simbolo passo attivo del passo STEP1.
T	Tempo di passo. Il simbolo T (tempo di passo) contiene il tempo finora trascorso dall'attivazione del passo in millisecondi. Quando un passo è inattivo, T contiene il tempo totale di esecuzione del passo. T viene messo a zero quando il passo diventa attivo.	STEP1.T riferimento al simbolo tempo di passo del passo STEP1.

Che cosa sono le azioni

Le azioni contengono la logica a contatti e sono rappresentate graficamente con un rettangolo che ne contiene il nome. Questo rettangolo è connesso al passo con una linea orizzontale. E' possibile associare più azioni a uno stesso passo.

Le azioni associate a un passo sono invocate quando il passo diventa attivo. E' possibile specificare il momento dell'esecuzione di un'azione rispetto al momento in cui il passo diventa attivo, ricorrendo a un qualificatore di azione.

- Il qualificatore di azione determina quando viene eseguita la logica a contatti rispetto all'attivazione del passo, che può essere in testo strutturato o in linguaggio Motion Control. E' possibile usare un qualificatore di azione con o senza un qualificatore di movimento. I qualificatori di azione sono elencati a pagina 4-47.
- Il qualificatore di movimento determina quando viene eseguita la logica a contatti rispetto all'esecuzione del codice Motion Control contenuto nel passo. I qualificatori di movimento sono elencati a pagina 4-46.

Il momento dell'esecuzione di un'azione può anche essere determinato usando un'etichetta del programma. In questo caso, l'azione non viene eseguita finché nel codice del passo non viene incontrata l'etichetta specificata.

Il qualificatore di azione è rappresentato da un rettangolo contenente un'abbreviazione, attaccato al lato destro dell'azione.

Esecuzione delle azioni

Nell'esempio che segue, l'azione PaintColorAction è costituita da vari rung RLL eseguiti quando diventa attivo il passo Step2. In questo esempio, l'esecuzione della logica a contatti non inizia finché l'esecuzione del codice del passo non incontra l'etichetta label_a. Il codice P è il qualificatore dell'azione e significa che la logica a contatti è eseguita solo una volta, cioè è impulsiva.



E' possibile associare zero o più azioni ad un passo ed è possibile associare un'azione a più di un passo facendo riferimento al nome dell'azione.

Una bobina di transizione SFC usata in un'azione associata ad un passo o ad un passo macro fornisce i seguenti controlli del flusso del programma:

- Associata ad un passo—Quando la bobina di transizione SFC riceve il flusso di corrente, il testo strutturato del passo è cancellato e l'esecuzione del programma salta all'etichetta SFC specificata nella bobina di transizione SFC.
- Associata a un passo macro— Quando la bobina di transizione SFC riceve il flusso di corrente, l'SFC figlio chiamato dal passo macro è cancellato e l'esecuzione del programma nell'SFC padre (l'SFC con il passo macro) salta all'etichetta SFC specificata nella bobina di transizione SFC.

Parametri delle azioni

Per creare un'azione ne devono essere configurati i parametri nella finestra di dialogo Edit Action. Questa tabella sintetizza i parametri:

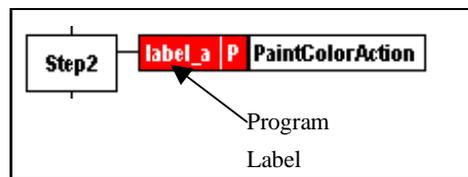
Campo/pulsante	Descrizione
Program Label	(facoltativo) L'esecuzione del codice RLL non inizia finché nel codice del passo non viene incontrata l'etichetta specificata.
Action Qualifier	Specifica un qualificatore di azione.
Motion Qualifier	Specifica un qualificatore di movimento. Selezionare None per nessun qualificatore. Per ulteriori informazioni riferirsi a "Qualificatori di movimento".
Time Duration	(Solo per qualificatore di azione) Specifica la durata per i qualificatori Limited e Delay. Se il qualificatore di azione non usa una durata, il valore di questo parametro è ignorato.
Specify Duration	Apri la finestra di dialogo Define Time Duration.
Action Name	Specifica il nome dell'azione.

Program Label (etichetta di programma)

Se è specificato questo parametro facoltativo, il codice RLL dell'azione non viene eseguito finché nell'esecuzione del codice del passo non viene incontrata l'etichetta specificata. L'azione e l'etichetta di programma devono essere nello stesso SFC. Non è ammesso il riferimento incrociato fra programmi. Se il passo al quale è associata l'azione non contiene l'etichetta, il parametro Program Label è ignorato.

Nel caso di un passo macro, non è possibile fare riferimento a un'etichetta dell'SFC figlio dall'SFC padre e viceversa.

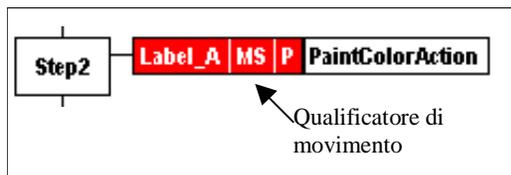
Per un passo con testo strutturato, l'etichetta è un nome seguito da due ":" (due punti), ad esempio, LabelA::. Per un passo con codice Motion Control, l'etichetta è una N seguita da un numero di blocco, ad esempio, N45. Se l'etichetta è specificata, essa compare nell'azione come segue:



Motion Qualifier (qualificatore di movimento)

Nota: Non tutti i qualificatori di movimento sono supportati da tutte le schede di controllo dei movimenti. Per sapere quali qualificatori sono supportati, fare riferimento alla documentazione della scheda.

I qualificatori di movimento specificano vincoli di movimento che devono essere soddisfatti prima che inizi l'esecuzione del codice RLL. Se è specificata un'etichetta di programma, il vincolo si applica ai blocchi di movimento successivi all'etichetta. Questi qualificatori compaiono nell'azione come segue:

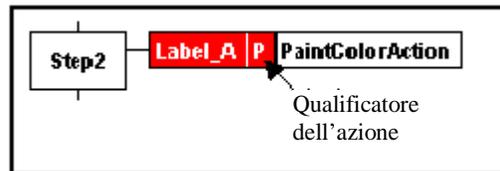


Scegliere tra i seguenti qualificatori. Lasciare in bianco se non deve essere specificato un qualificatore di movimento.

Se si vuole che l'esecuzione del codice RLL non inizi prima che:	Usare questo qualificatore:
inizi il movimento	Motion Started (MS)
sia completato il profilo di accelerazione	Acceleration Complete (AC)
il movimento raggiunga la velocità specificata	At Speed (AS)
inizi il profilo di decelerazione	Deceleration Started (DS)
sia finito il movimento	Motion Complete (MC)
sia finito il movimento e tutti gli assi associati al movimento siano entro la tolleranza di posizione rispetto al punto finale programmato	In Position (IP)
sia completata l'esecuzione del comando	End of Block (EB)

Action Qualifier (qualificatore dell'azione)

Questo parametro specifica un vincolo al quale è soggetta l'esecuzione del codice RLL. Il qualificatore compare nell'azione come segue:



Se dopo l'attivazione del passo si vuole che:	Usare il qualificatore
l'esecuzione dell'RLL inizi e continui finché il passo è attivo.	Non Stored (N)
l'esecuzione dell'RLL inizi e continui finché non viene resettata dal qualificatore Reset.	Stored (S)
l'RLL venga eseguito una sola volta.	Pulsed (P)
l'esecuzione dell'RLL inizi dopo un ritardo* e continui finché il passo è attivo.	Time Delayed (D)
l'esecuzione dell'RLL inizi e continui finché non scade il limite di tempo o il passo non diventa inattivo.	Time Limited (L)
l'esecuzione dell'RLL inizi dopo un ritardo* e continui finché non viene resettata dal qualificatore Reset. Se un altro qualificatore di azione resettava l'RLL durante il ritardo, il reset non ha effetto perché l'RLL non è ancora stata memorizzata. Se il passo diventa inattivo durante il ritardo, l'RLL non viene memorizzata né eseguita.	Delayed and Stored (DS)
l'esecuzione dell'RLL inizi dopo un ritardo* e continui finché non viene resettata dal qualificatore Reset. Se un'azione viene resettata durante il ritardo, l'RLL non è eseguita.	Stored and Time Delayed (SD)
l'esecuzione dell'RLL inizi e continui per il tempo specificato*. Per poter essere eseguita di nuovo, l'RLL deve essere resettata da un qualificatore Reset. Se il passo diventa inattivo, l'esecuzione dell'RLL continua finché non scade il limite di tempo. Non è richiesto un qualificatore Reset, ma se ne può usare uno per fermare l'esecuzione dell'RLL.	Stored and Time Limited (SL)
l'esecuzione dell'RLL inizi e continui per il tempo specificato*. Se il passo diventa inattivo, l'esecuzione dell'RLL continua finché non scade il limite di tempo. Non è richiesto un qualificatore Reset, ma se ne può usare uno per fermare l'esecuzione dell'RLL.	Pulse Width (PW)

L'RLL avviata dal qualificatore Stored viene fermata dal qualificatore Reset (R). Il qualificatore Reset può essere usato in un'altra azione associata allo stesso passo o in un'azione associata ad un altro passo. L'RLL continua a girare fra i passi. Se l'azione è associata ad un altro passo, essa deve avere lo stesso nome dell'azione da resettare.

Time Duration (Durata)

Introdurre la durata direttamente oppure cliccare su **Specify Duration** per visualizzare la corrispondente finestra di dialogo. Nello specificare una durata, fare riferimento alle regole e agli esempi forniti qui sotto.

- Se si introduce la durata direttamente, seguire le specifiche IEC 1131-3:
- Digitare uno dei seguenti prefissi T#, TIME#, t#, time#, seguito dalla durata in giorni, ore, minuti, secondi, millisecondi.

Durata	Formato	Durata	Formato
14.7 giorni	T#14.7d	4 secondi	Time#4s
2 minuti 5 secondi	T#2m5s	1 giorno 29 minuti	t#1d29m
74 minuti*	Time#74m	1 ora 5 secondi 44 millisecondi	T#1h5s44ms

*La specifica IEC 1131-3 permette il “superamento” dell’unità più significativa di una durata.

Se si preferisce usare la finestra di dialogo, introdurre in ciascun campo il valore appropriato.

Nota

Se per un’azione si specifica una durata e si sceglie un qualificatore di azione non dipendente dal tempo, la durata è ignorata.

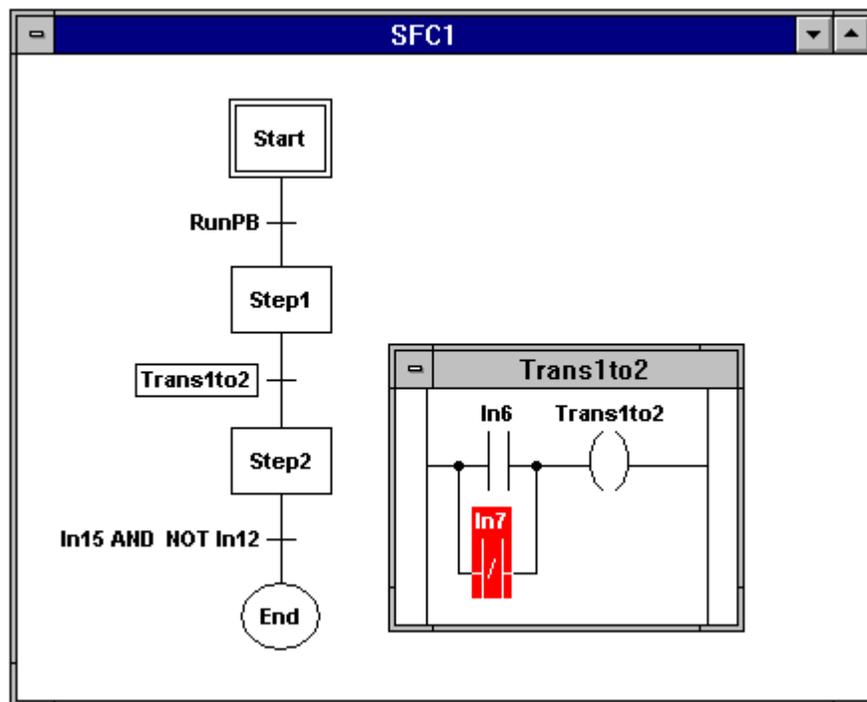
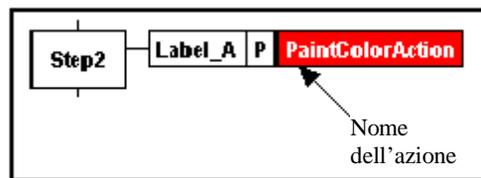
Se per un’azione sono specificate sia l’etichetta di programma che la durata, il conteggio della durata non inizia finché nel codice del passo non viene incontrata l’etichetta specificata.

Action Name (nome dell'azione)

Questo campo si usa per identificare l'azione con un nome. Specificare questo nome se ci si riferisce a questa azione dall'interno di un'altra azione, ad esempio, quando un'azione memorizzata si resetta in un'altra azione.

Un'azione può avere lo stesso nome di una transizione RLL. Tuttavia, la logica RLL delle azioni e quella delle transizioni hanno visibilità diversa. Di conseguenza, usare lo stesso nome per un'azione e per una transizione RLL non significa che nei due casi verrà eseguita la stessa logica.

Il nome appare nell'azione come si vede qui sotto:



Gestore delle azioni

Il gestore delle azioni (selezionare Action Manager nel menu Tools) si usa per manipolare le azioni associate ai passi SFC. Il gestore delle azioni visualizza l'elenco di tutte le azioni contenute nel file SFC attivo e permette di rinominarle o cancellarle.

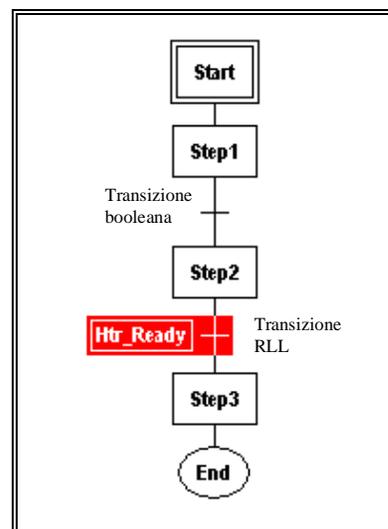
Che cosa sono le transizioni

Una transizione rappresenta la condizione in base alla quale il flusso del programma passa da uno o più passi precedenti la transizione ad uno o più passi seguenti la transizione, lungo il corrispondente collegamento orientato. A ciascuna transizione è associata una condizione di transizione, che è il risultato di una singola espressione booleana. Quando il sistema valuta il codice di una transizione, il risultato deve essere TRUE o FALSE.

Una condizione di transizione può essere una:

- **Transizione booleana** – Espressione booleana espressa in linguaggio ST (testo strutturato). Si rappresenta con una linea orizzontale non etichettata.
- **Transizione RLL** – Oggetto RLL denominato contenente un singolo rung con una bobina di uscita avente lo stesso nome dell'oggetto transizione. Si rappresenta con una linea etichettata (Transizioni RLL) contenente il nome della bobina di uscita RLL..

Il passo che segue una transizione non può essere eseguito finché la transizione che lo precede non è stata valutata, fornendo il risultato TRUE.



In questa figura il flusso del programma ha superato la transizione booleana e il passo Step2, che la segue, ed è attualmente alla transizione RLL. Finché la transizione RLL non risulta TRUE, il passo Step3 non può essere eseguito.

Il flusso del programma entra ed esce dalla transizione attraverso la linea verticale che incrocia la linea orizzontale rappresentante la transizione.

Lo standard IEC 1131-3 specifica che in un diagramma SFC tra due passi deve sempre trovarsi una transizione e che tra due transizioni deve sempre trovarsi un passo. L'editor SFC permette di specificare consecutivamente due passi o due transizioni. Tuttavia, durante il runtime vengono inseriti automaticamente le transizioni o i passi fittizi necessari

Parametri delle transizioni

Transizione	Parametri
RLL	Nome della transizione Logica RLL
Boolean	Espressione booleana contenente operatori e simboli

Il gestore delle transizioni RLL

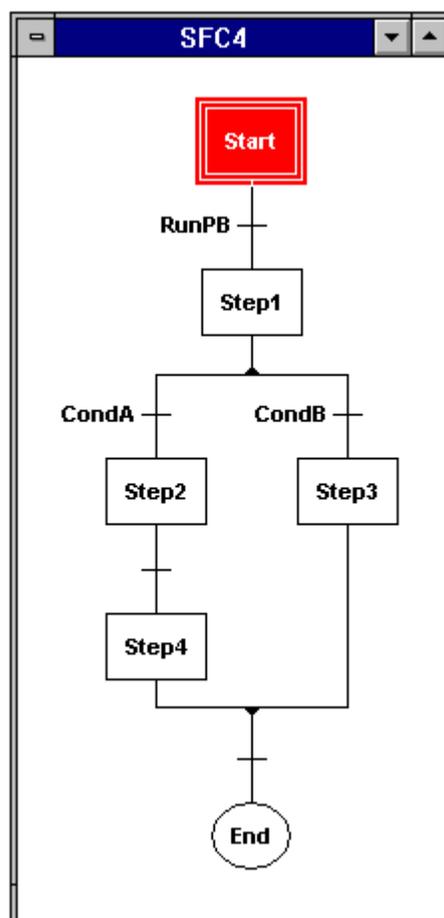
Il gestore delle transizioni RLL (RLL Transition Manager) si usa per manipolare le azioni RLL incluse in un programma SFC. Questo gestore visualizza un elenco di tutte le transizioni RLL contenute nel file attivo, permettendo di rinominarle e di cancellarle.

Che cosa sono le divergenze

Le divergenze permettono di controllare più percorsi in un programma SFC. Le divergenze possono essere selettive o simultanee.

Divergenza selettiva

Una divergenza selettiva permette di scegliere tra due o più percorsi di controllo. Ciascun percorso inizia con una condizione di transizione che determina quale percorso sarà attivato. Ad un certo punto del diagramma SFC, tutti i percorsi di una divergenza selettiva devono convergere in un unico percorso.

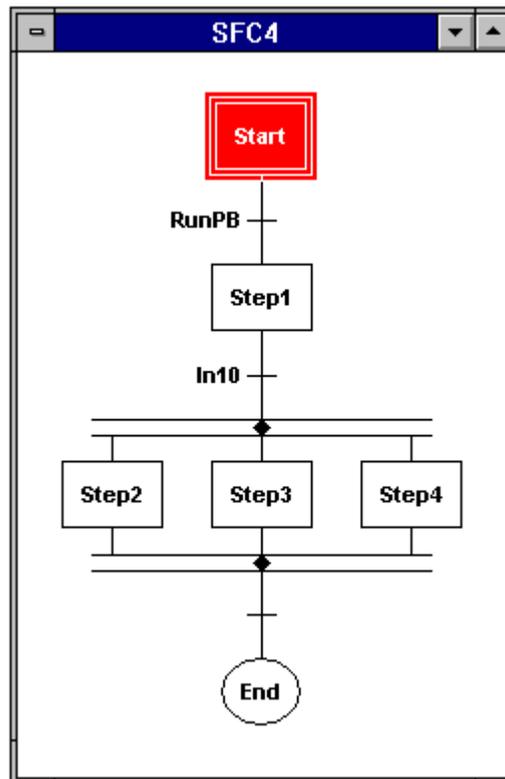


Divergenza simultanea

Una divergenza simultanea permette di eseguire più percorsi di controllo simultaneamente in parallelo. Tutti i percorsi di una divergenza simultanea sono attivati non appena il flusso di corrente entra nella divergenza. Ad un certo punto del diagramma SFC, tutti i percorsi della divergenza devono convergere in un unico percorso, però la convergenza deve attendere che tutti i percorsi siano:

- stati eseguiti
- arrivati al punto di convergenza simultanea.

Una divergenza simultanea si rappresenta come un singolo percorso di controllo che perviene a una doppia linea orizzontale da sotto alla quale escono due o più percorsi. La convergenza simultanea è rappresentata da due o più percorsi che pervengono ad una doppia linea orizzontale da sotto alla quale esce un unico percorso di controllo.



Per evitare di compromettere la correttezza della convergenza, non usare etichette per saltare:

- dall'interno all'esterno di una divergenza simultanea
- dall'esterno all'interno di una divergenza simultanea
- da un percorso ad un altro all'interno della stessa divergenza simultanea

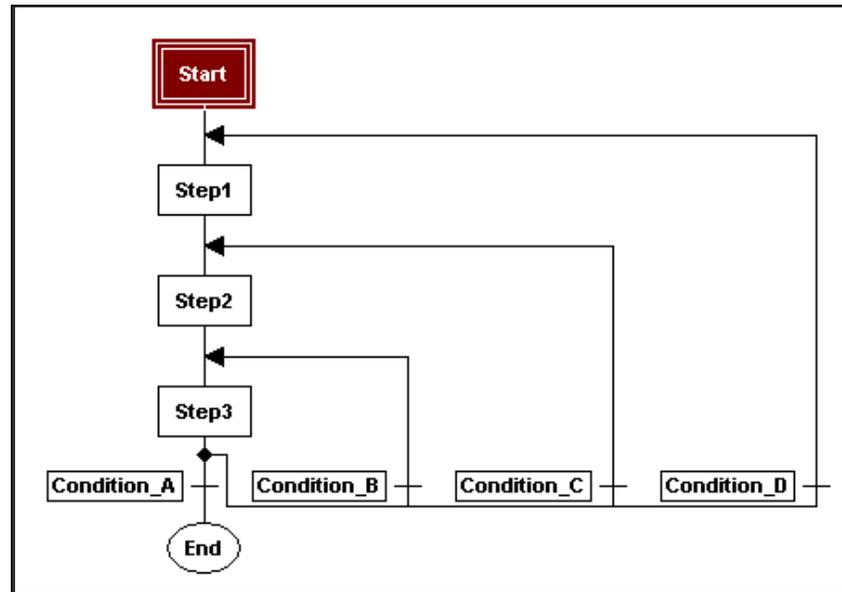
Mezzi di controllo del flusso del programma

Per controllare il flusso di un SFC è possibile ricorrere agli anelli di controllo e alle strutture salto/etichetta.

Anelli di controllo

In un SFC, il flusso del programma procede, usualmente, dall'alto verso il basso. Gli anelli di controllo permettono di ritornare a una posizione precedente per ripetere una serie di passi. Un anello di controllo è costituito da due transizioni: una per continuare verso il basso e una su un collegamento orientato che forma un anello ritornando verso l'alto. Il punto di rientro all'inizio dell'anello è indicato da una freccia.

Nella figura che segue, il flusso del programma continua fino a End se Condition_A è TRUE. Se Condition_B è True e Condition_A è FALSE, il controllo ritorna sopra a Step3. Se Condition_C è TRUE e Condition_A e Condition_B sono FALSE, il controllo ritorna sopra a Step2.



In presenza di diramazioni multiple, la valutazione della logica ha luogo da sinistra verso destra. Se tutte le transizioni sono FALSE, il flusso del programma si arresta finché una di esse non diventa TRUE.

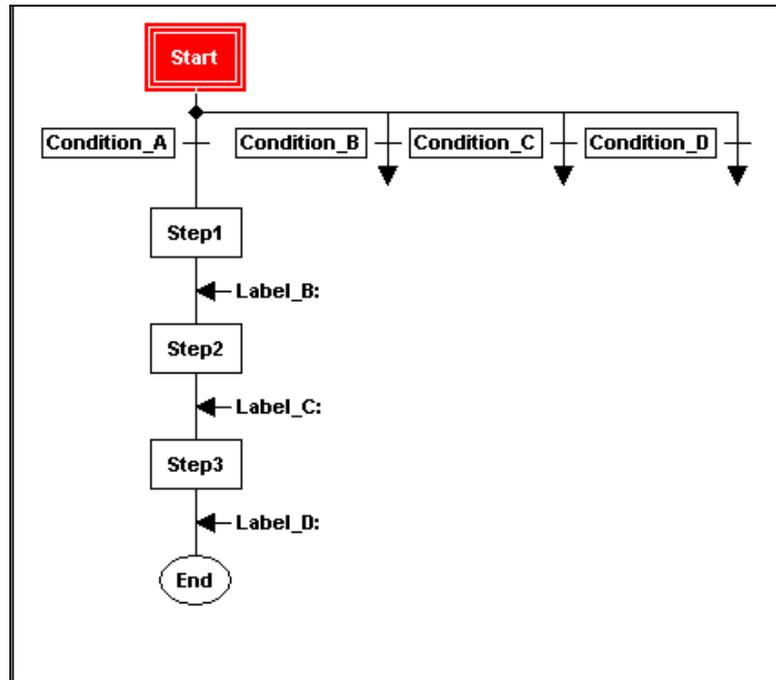
Le transizioni per l'anello di controllo possono essere definite in uno dei seguenti due modi:

- Con una transizione booleana consistente in un'espressione booleana scritta in testo strutturato.
- Con una transizione RLL consistente in un singolo rung RLL con una bobina di uscita avente lo stesso nome della transizione.

Salto e etichetta

Il flusso può essere trasferito a qualsiasi punto di un SFC ricorrendo al costrutto salto (jump) e etichetta (label). L'elemento salto è rappresentato da due transizioni: una per continuare normalmente verso il basso e una per trasferire il flusso a una etichetta. L'elemento etichetta è rappresentato da un arco orientato verso sinistra che identifica il punto in cui il flusso rientra nel percorso di controllo e da un identificatore seguito dal carattere ":" (due punti).

Nella figura che segue, il flusso del programma continua con Step1 se Condition_A è TRUE. Se Condition_B è True e Condition_A è FALSE, il flusso salta a Label_B, bypassando Step1. Se Condition_C è TRUE e Condition_A e Condition_B sono FALSE, il flusso del programma salta a Label_C, bypassando Step1 e Step2.



In presenza di diramazioni multiple, la valutazione della logica ha luogo da sinistra verso destra. Se tutte le transizioni sono FALSE, il flusso del programma si arresta finché una di esse non diventa TRUE.

Le transizioni per il salto possono essere definite in uno dei seguenti due modi:

- Con una transizione booleana consistente in un'espressione booleana scritta in testo strutturato.
- Con una transizione RLL consistente in un singolo rung RLL con una bobina di uscita avente lo stesso nome della transizione.

Parametri di salto e etichetta

Campo	Descrizione
Jump Target Label	Specifica l'etichetta alla quale viene trasferito il flusso del programma.
Label Name	Specifica il punto dell'SFC in cui riprende il flusso del programma..

Un'etichetta deve iniziare con un carattere alfabetico e può contenere qualsiasi carattere alfanumerico e il carattere di sottolineatura..

Come vengono risolti gli SFC

In un SFC il flusso del programma scorre dall'alto verso il basso. Viene eseguito il codice di ciascun passo. Quando il codice è stato completato, il flusso del programma passa all'elemento successivo dell'SFC.

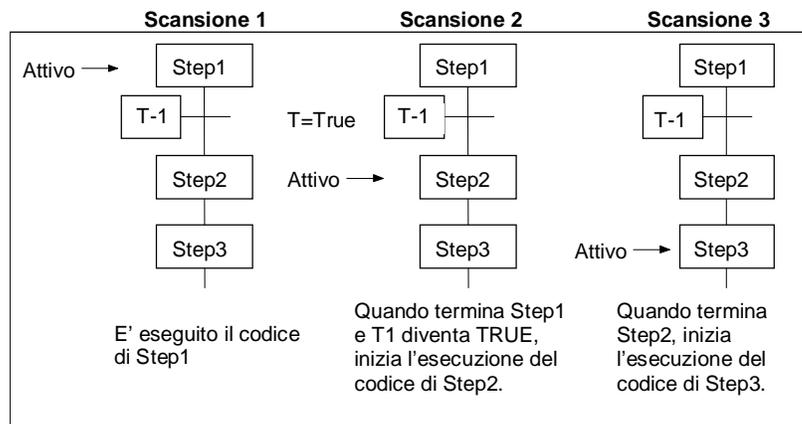
Se l'elemento successivo è:	Allora:
un passo	viene eseguito il codice di quel passo
una transizione	il flusso del programma continua quando la transizione diventa TRUE

Il completamento di un passo dell'SFC richiede un minimo di due scansioni degli I/O. Nella prima scansione, vengono valutate le azioni associate al passo. Se non esistono strutture cicliche (cioè FOR, WHILE, REPEAT), la logica scritta in testo strutturato viene risolta fino alla fine. Dopo che è stata risolta la logica (nella seconda scansione) viene testata la transizione.

Se un programma in testo strutturato è ciclico, il programma viene eseguito fino alla fine e si ripete all'inizio della successiva scansione degli I/O. Il test della transizione non è eseguito finché il testo strutturato non è stato completamente risolto. Se si desidera il ciclo venga completato in una scansione degli I/O, ricorrere ai costrutti END_FOR_NOWAIT e END_WHILE_NOWAIT.

Anche se l'editor dell'SFC permette di programmare passi consecutivi (non separati da una transizione), il completamento di un passo richiede comunque al **minimo due scansioni degli I/O**, perché la compilazione del programma crea tra quei passi una transizione fittizia. Una transizione fittizia è sempre TRUE. La presenza di queste transizioni è richiesta dallo standard IEC 1131-3. L'editor riduce lo sforzo di programmazione includendo automaticamente queste transizioni nel programma compilato.

Nell'esempio che segue, un passo segue direttamente il passo che lo precede. Il flusso del programma scorre dall'alto verso il basso e l'esecuzione di un elemento del programma non inizia finché non è stato completato l'elemento precedente. Dopo che la prima transizione è diventata TRUE, diventa attivo il passo Step2. Completata la logica di Step2 diventa attivo Step3.



Il software permette di eseguire più programmi di diversi tipi. Ad esempio, possono essere eseguiti contemporaneamente due programmi RLL e tre programmi SFC. L'esecuzione di questi programmi può essere sincronizzata mediante simboli globali, che sono riconosciuti da tutti i tipi di programmi.

Come sono valutate le transizioni

Il sistema valuta una transizione dopo che tutto il testo strutturato, o la logica RS-274D di un passo sono stati completati. Questa è un'altra estensione delle specifiche IEC 1131-3, che richiedono soltanto che tutti i passi che la precedono siano attivi, perché una transizione possa essere valutata.

Una transizione può seguire un'altra transizione, senza alcun passo fra le due. Quando una condizione di transizione è soddisfatta, la transizione viene disabilitata e viene attivato l'elemento SFC successivo, passo o transizione che sia. Nel runtime, fra due transizioni consecutive viene inserito un passo fittizio (per soddisfare lo standard IEC). Un passo fittizio non contiene logica, ma sono necessarie due scansioni degli I/O per attivarlo e disattivarlo, esattamente come per un passo normale.

La valutazione di una transizione avviene come segue:

Questa transizione:	Diventa TRUE quando:
Transizione RLL	Il flusso di corrente sul rung raggiunge la bobina di uscita e la attiva.
Transizione booleana	Il valore dell'espressione booleana diventa TRUE.

Il flusso del programma può continuare con il passo successivo solo dopo che la valutazione di una transizione dà come risultato TRUE.

Se la transizione è:	Allora:
True	Nella scansione successiva degli I/O, la logica RLL associata all'azione viene risolta senza corrente per disattivare le uscite. Poi, nella scansione successiva degli I/O, il flusso di corrente attiva il passo successivo.
False	Il passo rimane attivo e viene risolta la logica RLL. La logica RLL viene risolta per tutto il tempo in cui il passo è attivo; invece la logica in testo strutturato viene risolta una sola volta, nella prima scansione degli I/O.

Se una transizione è FALSE e rimane FALSE, il sistema non riesegue la logica di movimento o in testo strutturato dei passi che precedono la transizione. Invece la logica RLL viene rieseguita. Il flusso del programma rimane fermo alla transizione finché questa non diventa TRUE.

Uso delle divergenze simultanee

Quando si crea una divergenza simultanea:

- Non fare riferimento allo stesso simbolo come uscita di percorsi diversi della divergenza.
- Non richiamare lo stesso SFC figlio da passi macro di diversi percorsi della divergenza.

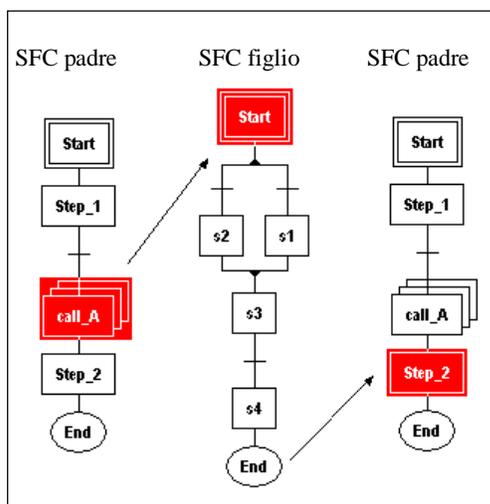
Per garantire la correttezza della divergenza non usare etichette per saltare:

- all'esterno di una divergenza simultanea
- all'interno di una divergenza simultanea
- ad un altro percorso della stessa divergenza simultanea.

Uso dei passi macro

In un SFC, un passo macro richiama per l'esecuzione un altro SFC. Il flusso del programma viene trasferito all'SFC richiamato (SFC figlio). Quando l'SFC figlio completa la propria esecuzione, il flusso del programma ritorna all'SFC padre e riprende dopo il passo macro.

Nella sottostante figura, call_A è il macro passo nell'SFC padre che richiama per l'esecuzione l'SFC figlio. Quando è completata l'esecuzione dell'SFC figlio, il flusso del programma riprende dal passo Step_2 dell'SFC padre.



La rappresentazione di un passo macro è simile a quella di un passo, un gruppo di rettangoli sovrapposti con un identificatore. Il flusso del programma entra nel passo macro attraverso la linea verticale che lo raggiunge dall'alto e ne esce attraverso la linea verticale che lo lascia dal basso.

E' possibile usare un altro elemento di programmazione, chiamato azione, come aiuto per coordinare l'esecuzione del codice di un passo macro con quella dell'altro codice del programma..

Se nell'SFC padre è definito un simbolo locale, l'SFC figlio può vederlo. Tuttavia, quando si apre il Symbol Manager per l'SFC figlio, quel simbolo non compare.

Estensioni dello standard IEC 1131-3

PC Control include varie estensioni opzionali dei diagrammi ladder e degli schemi funzionali sequenziali definiti dallo standard IEC 1131-3.

Estensioni del linguaggio RLL

- Le bobine di uscita possono essere inserite in qualsiasi punto del rung..
 - Esse memorizzano il risultato booleano parziale valutato fino a quel punto.
 - Esse passano la corrente all'elemento successivo alla loro destra.
- I rami "OR" che non contengono nessun elemento logico sono accettabili..
 - Essi possono essere usati per bypassare un blocco di logica senza cancellarlo.
 - Il compilatore dà un avvertimento per segnalare la possibile omissione di elementi..

Estensioni del linguaggio SFC

- I passi possono contenere comandi RS-274D o in testo strutturato. I comandi contenuti in un passo sono eseguiti in sequenza. L'esecuzione di un passo non è completa finché non sono stati completati tutti i comandi in esso contenuti. Una condizione di transizione non è valutata finché non è stato eseguito il contenuto del passo che la precede.
- Un passo può essere rappresentato con un rettangolo contenente il nome del passo, con un rettangolo contenente la lista dei comandi del passo, o con un'icona costituita da una bitmap e da un massimo di due righe di testo..
- Una paletta di icone contiene una raccolta di passi predefiniti costituiti da un'icona e da una lista di comandi. Questi passi possono essere usati come routine di libreria o fisse.
- E' possibile marcare i file per includerli in un passo. E' possibile includere più di un file. E' possibile mescolare i comandi del programma e le frasi di inclusione file.
- Un passo può seguire un altro passo senza interporre una transizione. In questo caso, non appena è stato eseguito il contenuto di un passo, quel passo viene disattivato e viene attivato il passo successivo. Nel runtime, tra due passi consecutivi viene inserita una transizione fittizia (per soddisfare lo standard IEC). Una transizione fittizia è sempre TRUE.
- Una transizione può seguire un'altra transizione senza interporre un passo. Non appena la condizione di transizione è soddisfatta la transizione viene disabilitata e viene attivato l'elemento successivo, passo o transizione che sia. Nel runtime tra due transizioni consecutive viene inserito un passo

fittizio (per soddisfare lo standard IEC). Un passo fittizio non contiene logica eseguibile, ma sono necessarie due scansioni degli I/O per attivarlo e disattivarlo, come per i passi normali.

- Agli elementi logici delle azioni è stata aggiunta una bobina di transizione (TRANS) che mette fine ai comandi contenuti nel passo, ferma eventuali movimenti in esecuzione, disattiva il passo e causa un'immediata transizione ad un'etichetta con lo stesso identificatore della bobina di transizione.
- E' possibile usare passi macro per includere in un passo un intero SFC. Quando viene attivato il passo macro, il diagramma SFC in esso incluso inizia ad essere eseguito dal passo Start. Quando il diagramma SFC incluso esegue il passo End, il passo macro è completato.
- A un passo macro possono essere associate azioni che saranno eseguite finché il passo macro sarà attivo.

Creazione di diagrammi funzionali sequenziali

Creazione di un programma SFC

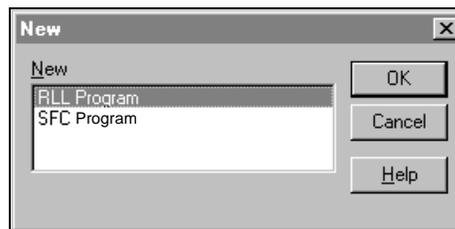
Dopo aver avviato il **Program Editor**, è possibile creare un nuovo programma SFC o modificarne uno esistente.

Per creare un nuovo programma SFC:

1. Cliccare sul pulsante *Nuovo file* della barra degli strumenti *Program Editor*.



Comparirà il menu nuovo programma.

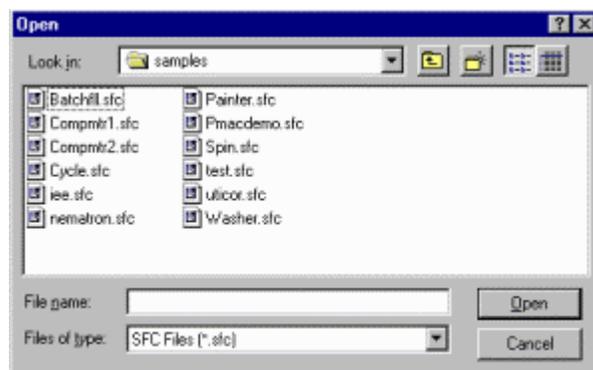


2. Selezionare *SFC Program* e cliccare su *OK*.
Comparirà un nuovo file SFC, con il passo Start e il passo End.

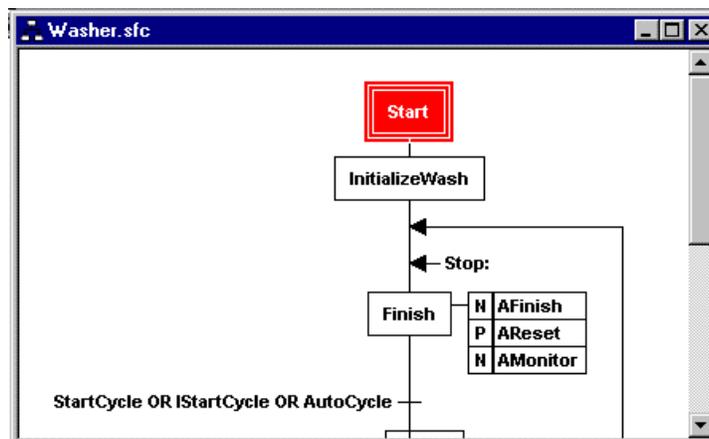


Per modificare un programma SFC esistente:

1. Cliccare sul pulsante Apertura file della barra degli strumenti del *Program Editor*. Comparirà la lista dei programmi esistenti.



2. Cliccare sul programma da aprire. Prima può essere utile selezionare il tipo di file SFC nel campo **Files of Type**. Comparirà il programma SFC selezionato.



3. Iniziare ad editare gli elementi del programma.

Uso della barra del menu e degli strumenti SFC

La barra degli strumenti SFC contiene tutti gli strumenti necessari per creare un programma SFC.



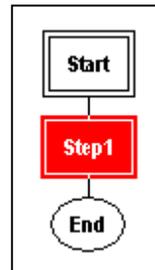
Icona	Scelta	Funzione
	—	Permette di selezionare gli elementi del programma.
	<i>Insert/Step</i>	Aggiunge un passo al programma.
	<i>Insert/Macro Step</i>	Aggiunge un passo macro al programma.
	<i>Insert/Action</i>	Aggiunge un'azione al programma.
	<i>Insert/Transition</i>	Aggiunge una transizione al programma.
	<i>Insert/Label</i>	Aggiunge un'etichetta al programma.
	<i>Insert/Jump</i>	Aggiunge un salto al programma.
	<i>Insert/Loop</i>	Aggiunge un anello al programma.
	<i>Insert/Select Diverge</i>	Aggiunge una divergenza selettiva al programma.
	<i>Insert/Simultaneous Diverge</i>	Aggiunge una divergenza simultanea al programma.
	<i>Insert/ New App Icon...</i>	Aggiunge al programma un passo predefinito registrato in libreria.
	<i>Insert/Comment</i>	Permette di aggiungere un commento al programma.

Operazioni sui passi

Un passo rappresenta una condizione nella quale il comportamento del sistema segue un insieme di regole definito nelle azioni e nelle funzioni associate a quel passo. Dopo aver aggiunto un passo al programma, è necessario configurarlo.

Aggiunta di un passo

1. Cliccare sullo strumento Passo della barra degli strumenti SFC. Il cursore cambia forma per indicare un passo.
2. Portare il cursore nella posizione del programma nella quale si desidera inserire il passo e cliccare. Il nuovo passo comparirà nel programma, nella posizione specificata.



3. Configurare il passo.

Configurazione di un passo

Per configurare un passo occorre:

- Specificare il linguaggio di programmazione che si desidera usare per il passo.
- Selezionare le informazioni che si desidera visualizzare nel passo (nome del passo, descrizione del passo, comandi di processo o icona)
- Introdurre il codice del programma per il passo o collegare un file contenente questo codice..

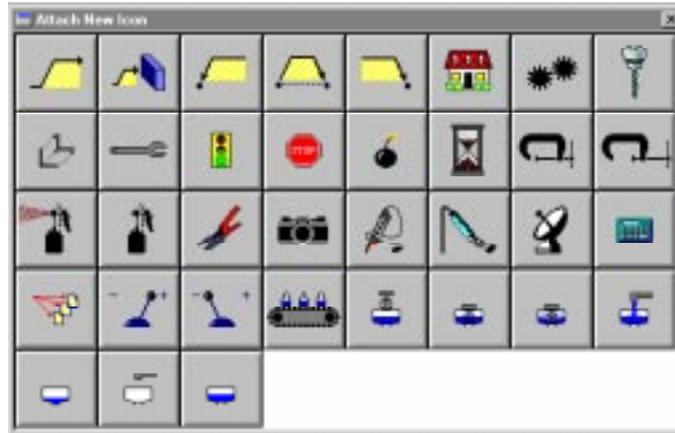
1. Cliccare sullo *strumento di selezione*  e fare doppio click sul passo. Comparirà la finestra di dialogo *Edit Step*.
2. Introdurre le informazioni di configurazione del passo, facendo riferimento alla sottostante tabella. Alla fine cliccare su **OK** per salvare le modifiche e chiudere la finestra di dialogo.

Per:	Fare questo:
Selezionare il linguaggio di programmazione RS-274D	Cliccare su RS-274D.

Per:	Fare questo:
Selezionare il linguaggio di programmazione Testo strutturato.	Cliccare su Structured Text.
Aggiungere codice del programma al passo.	<ol style="list-style-type: none"> 1. Cliccare su Process Commands. 2. Introdurre il codice o collegare un file.
Collegare il passo ad un file contenete il codice del programma.	Cliccare su Link File.
Aprire il file di codice collegato al passo e visualizzarne il contenuto in un editor.	Cliccare su Edit Linked File.
Accedere al Symbol Manager.	Cliccare su Symbol Manager Display.
Visualizzare il nome del passo nel rettangolo che lo rappresenta.	<ol style="list-style-type: none"> 1. Cliccare su Step Name. 2. Digitare il nome nella finestra di edit.
Visualizzare il codice (Motion Control o testo strutturato) del passo nel rettangolo che lo rappresenta.	<ol style="list-style-type: none"> 1. Cliccare su Motion/Process Commands. 2. Introdurre il codice o collegare un file. <p>Tutto il codice di programmazione per il passo compare nel passo. Se i comandi sono molti, la dimensione del passo aumenta significativamente.</p>
Visualizzare la descrizione del passo nel rettangolo che lo rappresenta.	<ol style="list-style-type: none"> 1. Cliccare su Step Description. 2. Cliccare su Edit Description. 3. Introdurre la descrizione nella finestra Edit Description.
Fissare la lunghezza in pixel della descrizione del passo.	Introdurre il numero di pixel per la descrizione del passo nel campo Width.
Visualizzare un'icona nel passo.	Cliccare su Icon.
Accedere alla paletta Application Icon.	Cliccare su Icon. . . .
Cancellare l'icona da un passo, se gli era stata assegnata.	Cliccare su Remove.

Visualizzazione di un passo sotto forma di icona

1. Editare il passo.
2. Nella finestra di dialogo Edit Step, cliccare su Icon. Comparirà la paletta delle icone.



3. Cliccare sull'icona desiderata e introdurre un titolo quando il sistema lo chiede.
4. Cliccare su OK per ritornare alla finestra di dialogo Edit Step. Terminata l'editazione del passo, l'icona e il titolo compariranno nel rettangolo che rappresenta il passo.

Aggiunta di un passo predefinito (Application Icon Step)

Un Application Icon Step è un passo contenente un modello di codice predefinito per una specifica funzione e un'icona appropriata alla funzione. Il software PC Control fornisce una libreria di passi ampliabile dall'utente. Un SFC può usare questi passi.

Per aggiungere un passo predefinito

1. Accedere alla libreria delle icone applicative cliccando sullo strumento Icon della barra degli strumenti SFC.
2. Trascinare l'icona applicativa desiderata nell'SFC.
3. Dopo aver posizionato l'icona, editarla allo stesso modo di un passo normale.

Dato che il codice è un modello, è necessario editarlo per adeguarlo alle esigenze reali dell'applicazione. E' anche possibile apportare altri cambiamenti alle opzioni del passo.

Dopo aver personalizzato un passo predefinito, è possibile aggiungere il passo personalizzato alla libreria. Poi, ogni volta che sarà necessario usare un copia di quel passo in un SFC, sarà possibile recuperarla dalla libreria.

Operazioni sulle transizioni

Una transizione è un elemento grafico del linguaggio di programmazione SFC che rappresenta una singola condizione booleana che deve essere soddisfatta prima che l'esecuzione del programma continui.

Aggiunta di una transizione RLL

1. Cliccare su *Edit* nella barra del menu e se accanto a *Boolean Transition* compare il segno di selezione cliccare per deselegionarla.
2. Cliccare sullo strumento transizione della barra degli strumenti SFC. Il cursore cambia forma per indicare la transizione.
3. Portare il cursore nel punto del programma nel quale si desidera inserire la nuova transizione e cliccare. La nuova transizione compare nel programma nella posizione specificata.

Editazione di una transizione RLL

1. Cliccare sullo strumento di selezione  e fare doppio click sulla transizione da editare. Comparirà la finestra di dialogo *Select RLL Transition Logic*.
2. Introdurre il nome della transizione e cliccare su *OK*.
 - Una transizione RLL può avere lo stesso nome di un'azione. Tuttavia, la logica RLL delle transizioni e la logica RLL delle azioni hanno visibilità diversa. Questo significa che se un'azione e una transizione hanno lo stesso nome non verrà comunque eseguita la stessa logica.
 - Comparirà una finestra contenente un rung RLL con una bobina avente lo stesso nome della transizione.
3. Usare l'editor RLL per aggiungere al rung gli elementi RLL necessari, usando le normali regole relative a contatti, bobine, salti, ecc.
4. Salvare il lavoro fatto, cliccando su *Control Menu* e selezionando *Close*.

Aggiunta di una transizione booleana

1. Cliccare su *Edit* nella barra del menu e, se accanto a *Boolean Transition* non compare il segno di selezione, cliccare per selezionarla.
2. Cliccare sullo strumento transizione della barra degli strumenti SFC. Il cursore cambia forma per indicare la transizione.
3. Portare il cursore nel punto del programma nel quale si desidera inserire la nuova transizione e cliccare. La nuova transizione compare nel programma, nella posizione specificata.

Editazione di una transizione booleana

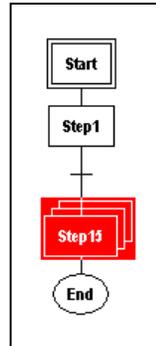
1. Per editare una transizione, cliccare sullo strumento di selezione  e fare doppio click sulla transizione. Comparirà la finestra di dialogo *Edit Transition Logic*.
2. Introdurre il codice booleano per la transizione. E' possibile digitare questo codice o costruirlo cliccando sui pulsanti degli operatori logici e selezionando gli operandi fra i simboli che sono stati definiti.
3. Cliccare su *OK* per salvare il lavoro fatto e chiudere la finestra di dialogo.
4. E' possibile accedere al Symbol Manager per configurare variabili locali e vedere l'elenco di tutte le variabili già configurate utilizzabili nell'espressione booleana.

Operazioni sui passi macro

Un passo macro permette di richiamare un altro SFC dall'SFC attualmente in esecuzione.

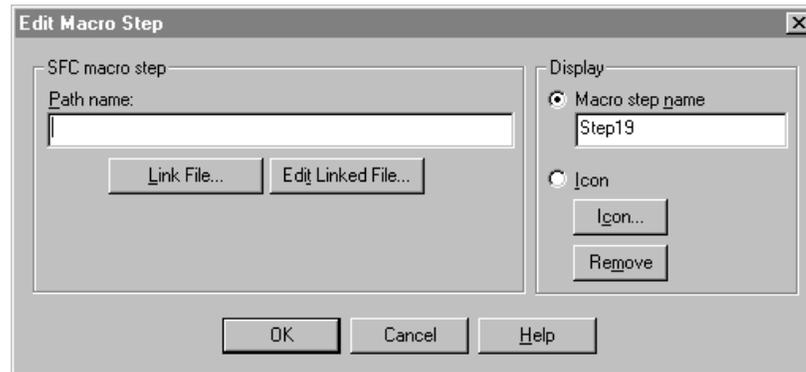
Aggiunta di un passo macro

1. Cliccare sullo strumento passo macro della barra del menu SFC. Il cursore cambia forma per indicare il passo macro.
2. Portare il cursore nel punto del programma nel quale si desidera inserire il nuovo passo macro e cliccare. Il nuovo passo macro compare nel programma, nella posizione specificata.



Configurazione di un passo macro

1. Cliccare sullo strumento di selezione  e fare doppio click sul passo macro. Comparirà la finestra di dialogo **Edit Macro Step**.



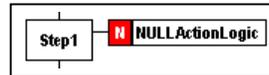
2. Introdurre le informazioni appropriate per la configurazione del passo macro.
3. Cliccare su *OK* per salvare le modifiche e chiudere la finestra di dialogo.

Operazioni sulle azioni

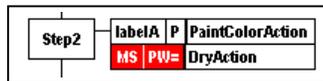
Un'azione consiste in una o più sezioni di codice RLL associate a un passo o a un passo macro. Il sistema esegue un'azione quando il passo al quale è associata diventa attivo.

Aggiunta di un'azione

1. Cliccare sullo strumento azione della barra del menu SFC. Il cursore cambia forma per indicare l'azione.
2. Portare il cursore nella posizione del programma (su un passo o su un passo macro) nella quale si desidera inserire la nuova azione e cliccare. La nuova azione comparirà nel programma, nella posizione specificata.

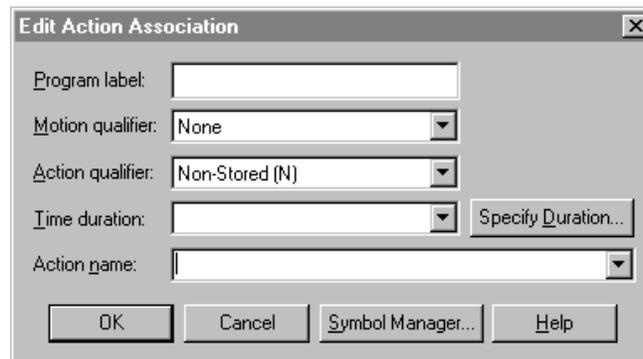


3. Posizionando il cursore su un'azione esistente è possibile aggiungere più di un'azione a un passo o a un passo macro.



Configurazione di un'azione

Per configurare le azioni si usa la finestra di dialogo Edit Action Association.



Campo/pulsante	Descrizione
Program Label	(Facoltativo) L'azione non è eseguita finché nel codice del programma non viene incontrata l'etichetta specificata. L'azione e l'etichetta devono essere nello stesso SFC. Non è ammesso il riferimento incrociato fra programmi. Per un passo macro, non è possibile fare riferimento alle etichette dell'SFC figlio dall'SFC padre e viceversa. Se il passo al quale l'azione è associata non ci sono etichette, questo parametro è ignorato.
Action Qualifier	Specifica un qualificatore di azione.
Time Duration	(Riferita al qualificatore di azione) Specifica la durata per i qualificatori limitata e ritardata. Se il qualificatore di azione specificato non usa la durata, questo parametro è ignorato.
Specify Duration	Apri la finestra di dialogo Define Time Duration.
Action Name	Specifica il nome dell'azione.

Editazione di un'azione

1. Cliccare sullo strumento di selezione  e fare doppio click sull'azione da editare. Comparirà la finestra di dialogo **Edit Action Association**. Introdurre le informazioni appropriate per la configurazione dell'azione.
2. Cliccare su *OK* per salvare le modifiche. Il sistema chiude la finestra di dialogo e visualizza un rung vuoto per l'editazione.
3. Introdurre il codice RLL.

Editazione dell'RLL di un'azione

4. Cliccare sullo strumento di selezione .
5. Fare doppio click sulla parte destra dell'azione, come si vede qui sotto..



Il sistema visualizza il codice RLL dell'azione. Introdurre il codice RLL.

Editazione dei parametri di configurazione di un'azione

1. Cliccare sullo strumento di selezione .
2. Fare doppio click sulla parte sinistra dell'azione, come si vede qui sotto. Il sistema visualizza la finestra di dialogo **Edit Action Association** per l'azione selezionata.



3. Introdurre le informazioni appropriate per la configurazione dell'azione.
4. Cliccare su *OK* per salvare le modifiche.

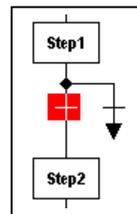
Aggiunta dei controlli di flusso del programma SFC

Il flusso del programma può essere controllato per mezzo di salti, anelli e divergenze.

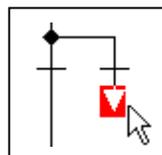
Aggiunta di un salto

E' possibile specificare una combinazione salto/etichetta che permette di trasferire l'esecuzione di un SFC alla posizione indicata dall'elemento etichetta.

1. Cliccare su *Edit* nella barra del menu e selezionare *Boolean Transition*.
2. Cliccare sullo strumento salto della barra del menu SFC. Il cursore cambia forma per indicare il salto.
3. Portare il cursore nella posizione del programma nella quale si desidera inserire il nuovo salto e cliccare. Nella posizione specificata compaiono il nuovo salto e due transizioni.



4. Per editare il salto, cliccare sullo strumento di selezione  e fare doppio click sulla punta della freccia del salto.



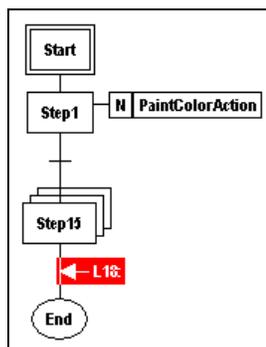
Compare la finestra di dialogo **Edit Jump Target**.



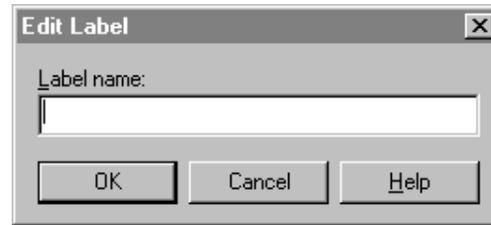
5. Introdurre il nome dell'etichetta alla quale il salto trasferisce il flusso del programma. Poi cliccare su *OK* per salvare le modifiche e chiudere la finestra di dialogo.
6. Editare le due transizioni.

Aggiunta di un'etichetta

1. Cliccare sullo strumento etichetta della barra del menu SFC. Il cursore cambia forma per indicare l'etichetta.
2. Portare il cursore nella posizione del programma nella quale si desidera inserire la nuova etichetta e cliccare. La nuova etichetta comparirà nel programma, nella posizione indicata.



3. Cliccare sullo strumento di selezione  e fare doppio click sull'etichetta da editare. Comparirà la finestra di dialogo **Edit Label**.



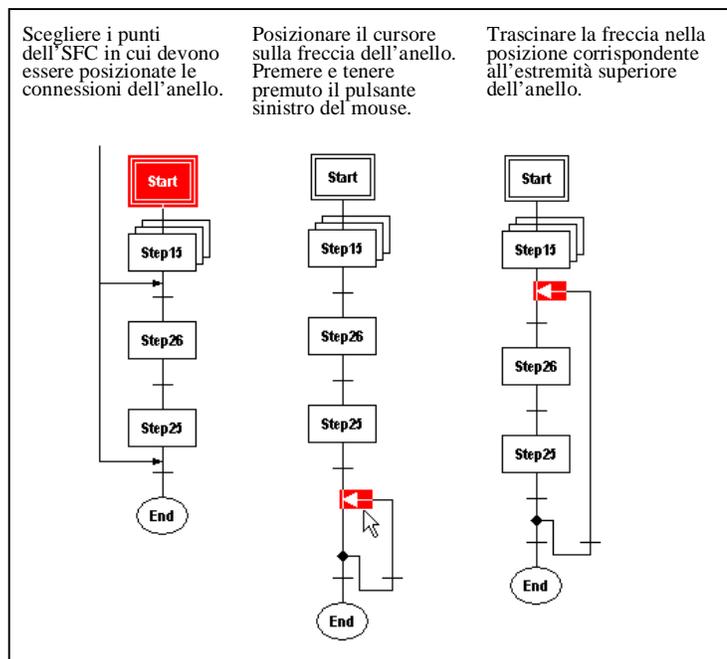
4. Introdurre un nome significativo e cliccare su OK per salvare le modifiche e chiudere la finestra di dialogo.

Aggiunta di un anello

Un anello permette di riportare il flusso del programma a un punto precedente per ripetere una serie di passi.

1. Cliccare sullo strumento anello della barra del menu SFC. Il cursore cambia forma per indicare l'anello.
2. Portare il cursore nella posizione del programma nella quale si desidera inserire l'estremità inferiore dell'anello e cliccare. L'anello comparirà nel programma, nella posizione indicata.
3. Cliccare sullo strumento di selezione  e portare il cursore sulla freccia dell'anello.

4. Premere e tenere premuto il pulsante sinistro del mouse. Trascinare la punta della freccia nella posizione corrispondente all'estremità superiore dell'anello.



5. Editare le transizioni dell'anello.

Spostamento di un anello

Trascinamento dell'estremità superiore dell'anello:

1. Cliccare sullo strumento di selezione della barra degli strumenti RLL o SFC.
2. Portare il cursore sulla freccia dell'anello. Premere e tenere premuto il pulsante sinistro del mouse. Trascinare la freccia dell'anello nella posizione desiderata (il cursore cambierà forma per indicare la freccia dell'anello).
3. Rilasciare il pulsante sinistro del mouse per posare la freccia dell'anello nella posizione desiderata. Per cancellare l'operazione di trascinamento premere il tasto <ESCAPE>.

Se la destinazione della freccia :	Allora:
E' sullo stesso ramo di una divergenza e non è più in basso della transizione di anello alla quale è legata questa freccia Non è all'interno di un anello incluso Non è all'esterno di un anello circostante	La freccia si posiziona nel punto in cui viene posta e la transizione dell'anello non si muove.
E' su una divergenza diversa E' più in basso della transizione di anello alla quale è legata questa freccia E' all'esterno di un anello circostante	L'intero anello con tutto il suo contenuto si sposta nella nuova posizione..
E' all'interno di un anello incluso	Il trascinamento fallisce.

Trascinamento dell'estremità inferiore dell'anello:

1. Portare il cursore sulla transizione dell'anello. Premere e tenere premuto il pulsante sinistro del mouse. Trascinare la transizione dell'anello nella posizione desiderata (il cursore cambierà forma per indicare la transizione di anello).
2. Rilasciare il pulsante sinistro del mouse per posare la transizione dell'anello nella posizione desiderata.

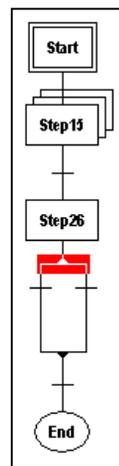
Per cancellare l'operazione di trascinamento premere il tasto <ESCAPE>

Se la destinazione della transizione	Allora:
E' sullo stesso ramo di divergenza e non è più in alto di una qualsiasi delle frecce di anello legate a questa transizione Non è all'interno di un anello incluso Non è all'esterno di un anello circostante	La transizione si posiziona nel punto in cui viene posata e la freccia dell'anello non si muove.
E' su una divergenza diversa E' più in alto di una qualsiasi delle frecce di anello legate a questa transizione E' all'esterno di un anello circostante	L'intero anello con tutto il suo contenuto si sposta nella nuova posizione..
E' all'interno di un anello incluso	Il trascinamento fallisce e l'anello ritorna nella sua posizione originale.

Aggiunta di una divergenza selettiva

Una divergenza selettiva permette di selezionare un percorso fra due o più percorsi possibili.

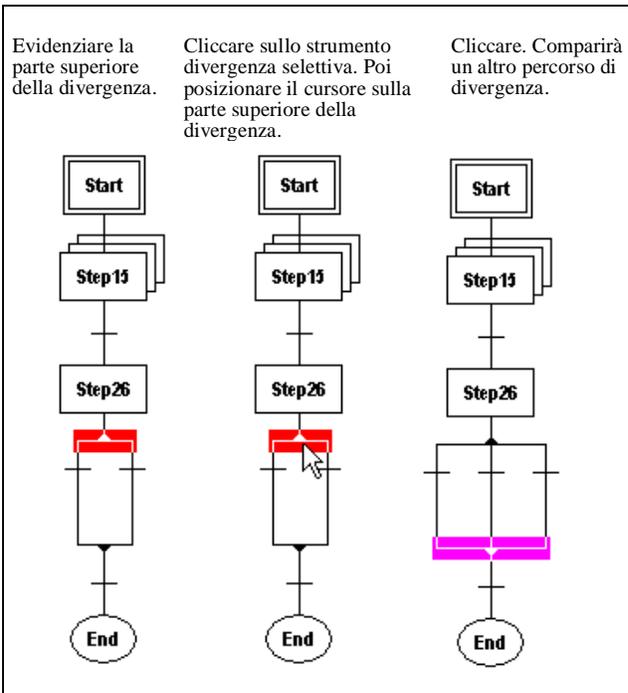
1. Cliccare su *Edit* nella barra del menu e selezionare *Boolean Transition* per il tipo delle transizioni da usare nella divergenza: RLL o booleane.
2. Cliccare sullo strumento divergenza selettiva nella barra del menu SFC. Il cursore cambia forma per indicare la divergenza selettiva.
3. Portare il cursore nella posizione del programma nella quale si desidera inserire la divergenza selettiva e cliccare. La divergenza selettiva compare nel programma, nella posizione specificata.



4. Editare le due transizioni.

Per aggiungere un altro percorso:

1. Prima di aggiungere un altro percorso, evidenziare la parte superiore della divergenza.
2. Posizionare il cursore sulla parte superiore della divergenza e cliccare. Comparirà un altro percorso.



Per cancellare un percorso:

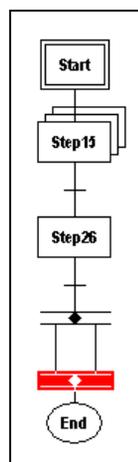
1. Evidenziare il percorso da cancellare.

2. Cliccare sullo strumento taglia .

Aggiunta di una divergenza simultanea

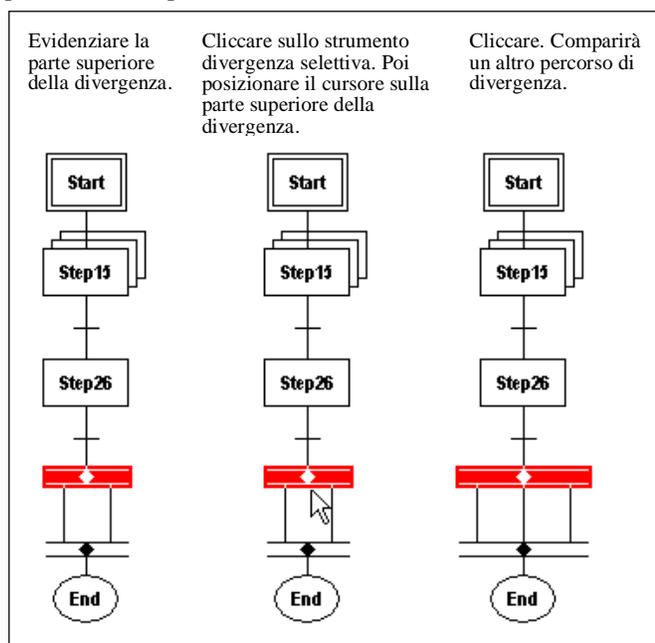
Una divergenza simultanea permette all'esecuzione di un programma SFC di seguire simultaneamente due o più percorsi. L'esecuzione del programma non procede oltre una divergenza simultanea finché tutti i percorsi della divergenza non sono stati completati.

1. Cliccare sullo strumento divergenza simultanea della barra del menu SFC. Il cursore cambia forma per indicare la divergenza simultanea.
2. Portare il cursore nella posizione del programma nella quale si desidera inserire la divergenza simultanea e cliccare. La divergenza simultanea compare nel programma, nella posizione specificata.



Per aggiungere un altro percorso:

1. Prima di aggiungere un altro percorso, evidenziare la parte superiore della divergenza..
2. Posizionare il cursore sulla parte superiore della divergenza e cliccare. Comparirà un altro percorso.



Per cancellare un percorso:

1. Evidenziare il percorso da cancellare.

2. Cliccare sullo strumento *taglia* .

Per cancellare un'intera divergenza simultanea:

1. Evidenziare la parte superiore o la parte inferiore della divergenza.

2. Cliccare sullo strumento *taglia*.

Linee guida per l'uso delle divergenze simultanee

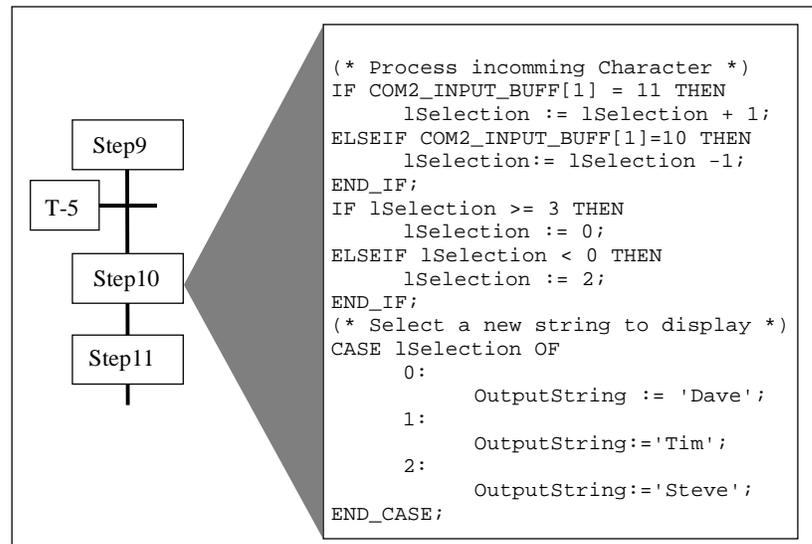
Nella creazione di una divergenza simultanea seguire queste linee guida.

Per ottenere una convergenza corretta, non usare etichette per:

- saltare dall'interno all'esterno di una divergenza simultanea;
- saltare dall'esterno all'interno di una divergenza simultanea;
- saltare da un percorso ad un altro della stessa divergenza simultanea.

Integrazione di testo strutturato in un SFC

Il linguaggio di programmazione Testo strutturato è un insieme di istruzioni testuali conforme allo standard IEC 1131-3 concepito per facilitare la creazione di operazioni matematiche e logiche. Per creare il codice di un passo SFC, si può scegliere di usare il linguaggio Testo strutturato, come è illustrato qui sotto. Quando viene eseguito l'SFC, il codice in testo strutturato incorporato in un passo viene processato all'attivazione del passo.



Nel progettare il programma, tenere presenti i seguenti punti:

- Il corpo di un programma in testo strutturato è costituito da espressioni, frasi e funzioni. Gli operatori e i tipi di dati utilizzabili in un programma in testo strutturato sono descritti nella sezione “Espressioni in testo strutturato”.
- Le dichiarazioni dei simboli da usare nel programma si eseguono nel Symbol Manager.
- E' possibile inizializzare i simboli attraverso una o più azioni, che possono essere associate al passo.
- Evitare di usare le parole riservate (o parole chiave) del PC Control.

Documentazione di un programma SFC

Aggiunta di commenti al programma

Un commento è una descrizione significativa che si desidera visualizzare vicino ad un elemento del programma. E' possibile scegliere se il sistema visualizza i commenti o li nasconde.

1. Cliccare sullo strumento commento della barra del menu SFC. Il cursore cambia forma per indicare il commento.
2. Portare il cursore nella posizione del programma nella quale si desidera inserire il commento e cliccare. Comparirà la finestra di dialogo *Program Comments*.
3. Introdurre il commento e cliccare su *OK*. Il commento compare nel programma, nella posizione specificata.

Modifica dei commenti al programma

1. Fare doppio click sul commento che si desidera modificare. Comparirà la finestra di dialogo **Program Comments**.
2. Modificare il commento come si desidera.

Visualizzazione dei commenti

Cliccare su *View* e selezionare *Program Comments* oppure cliccare sul pulsante commenti della barra degli strumenti dell'editor.

Per disattivare il modo visualizzazione commenti:

Cliccare su *View* e selezionare *Program Comments* oppure cliccare sul pulsante commenti della barra degli strumenti dell'editor.

Nota

Il modo visualizzazione commenti è attivo o inattivo per tutti i file.

Quando il modo visualizzazione commenti è attivo, nel menu view compare il segno di selezione accanto al comando Program Comments e il pulsante commenti della barra degli strumenti dell'editor è visualizzato come premuto.

Sezione 4: Programmazione con il linguaggio ST (Testo strutturato)

Introduzione

Il linguaggio di programmazione Testo strutturato è un linguaggio testuale di programmazione definito nello standard IEC 1131-3. E' conveniente per coloro che conoscono BASIC, PASCAL, C o altri linguaggi di programmazione ad alto livello.

Per creare programmi stand-alone in testo strutturato si usa l'editor per il testo strutturato. Questo editor dispone delle funzioni tipiche di un editor di testo quali taglia, copia, incolla, trova e sostituisci. Esso dispone anche di strumenti e comandi per l'inserimento automatico di costrutti come le frasi di selezione IF e CASE e le iterazioni FOR, REPEAT e WHILE.

E' anche possibile incorporare comandi in testo strutturato in un passo SFC. Una estensione brevettata del linguaggio SFC permette all'utente di scrivere in testo strutturato i comandi di un passo SFC. Quando l'utente crea il codice applicativo di un passo SFC, ha la possibilità di scegliere di usare il testo strutturato. Quando l'SFC viene eseguito, il codice in testo strutturato incorporato in un passo viene eseguito all'attivazione di quel passo. Salvo che per certe funzioni, l'editor di testo strutturato stand-alone e l'editor di testo strutturato di un passo SFC funzionano allo stesso modo.

Questa sezione descrive l'uso dell'editor di testo strutturato. Si assume che il lettore abbia familiarità con le normali operazioni di un editor di programmi e con il linguaggio Testo strutturato.

Nota

I programmi stand-alone in testo strutturato girano una volta ed escono.

Apertura di un documento in testo strutturato

Per aprire un documento esistente, selezionare *Open Editor* nel menu *File* del Program Editor e localizzare il documento nella finestra di dialogo *Open* che compare di conseguenza.

Per aprire un nuovo documento, selezionare *New Editor* nel menu *File* del Program Editor e scegliere *Structured Text Document* nella finestra di dialogo *New* che compare di conseguenza.

Editazione di testo strutturato in un passo SFC

Per editare del testo strutturato in un passo SFC esistente, fare doppio click sul passo o selezionare il passo e scegliere *Edit Element* nel menu *Edit* o *Edit ST* o *Macro Step* nel menu contestuale. Accertarsi che il contenuto di *Step Properties* sia *Structured Text*.

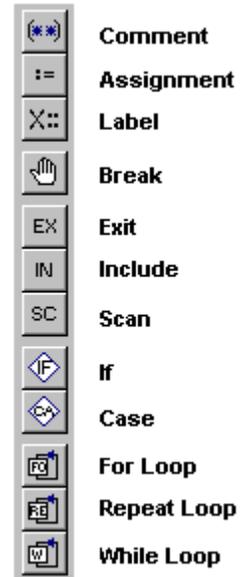
Introduzione delle istruzioni

Introduzione manuale

Le istruzioni possono essere introdotte digitando le frasi di comando o i richiami di funzione con i corrispondenti parametri. La sintassi delle frasi di comando è descritta nel paragrafo “Descrizione del linguaggio” e la sintassi dei richiami di funzione è descritta nel “Language Reference”.

Barra degli accessori

La barra degli accessori presenta i comandi in forma grafica. E' un metodo alternativo di introduzione delle istruzioni. La barra degli accessori compare solo se è abilitata commutando *Accessory Bar* nel menu *View*. La figura a lato presenta la barra degli accessori, che può essere posizionata dall'utente in base alla propria convenienza.



Menu Insert ST Statements (inserisci frasi ST)

Quando è attivo l'editor di testo strutturato, il menu *Edit* del Program Editor include una voce *Insert ST Statements* che elenca le frasi del linguaggio. Selezionare la frase necessaria, che comparirà automaticamente nella posizione del cursore, con la sintassi appropriata. Sostituire i parametri e le espressioni come necessario. Le parti facoltative non utilizzate della frase devono essere rimosse.

Menu Insert ST Functions (inserisci funzioni ST)

Quando è attivo l'editor di testo strutturato, il menu *Edit* del Program Editor include una voce *Insert ST Function Calls* che elenca le funzioni standard utilizzabili nel linguaggio Testo strutturato. Selezionare la funzione necessaria, che comparirà automaticamente nella posizione del cursore, con la sintassi appropriata. Sostituire i parametri con quelli definiti nel Symbol Manager.

Fare riferimento a “Descrizione del linguaggio” per ulteriori informazioni sull'uso di funzioni e di blocchi funzionali nel linguaggio Testo strutturato.

Editazione del testo strutturato

L'editor di testo strutturato dispone delle normali funzioni di un editor di testo, quali taglia, copia, incolla, trova e sostituisci. Questi comandi si trovano nel menu *Edit* e nel menu contestuale.

Descrizione del linguaggio

Espressioni

Un'espressione è definita come una combinazione di operatori (matematici, logici, relazionali) e di operandi (costanti, simboli, valori letterali, altre espressioni) che possono essere valutati, fornendo un risultato in uno dei tipi di dati supportati, come intero, numero reale, e così via.

Operatori

La tabella che segue elenca gli operatori che possono essere usati in un'espressione. L'ordine di precedenza determina la sequenza con cui sono valutati all'interno dell'espressione. L'operatore con la precedenza più elevata è applicato per primo, seguito dall'operatore con la precedenza più elevata tra i restanti. Gli operatori con eguale precedenza sono valutati da sinistra a destra.

Operatore	Simbolo	Precedenza
Indice di struttura	.	1 (Più elevata)
Indice di array	[]	1
Assegnazione indirizzo a puntatore	&	2
Indirizzamento con puntatore	*	2
Parentesi	()	3
Valutazione di funzione	Identificatore (lista argomenti) es.: LN (A), ABS (X)	3
Elevazione a potenza	** , POW	4
Negazione	-	5
Complemento	NOT	5
Moltiplicazione	*	6
Divisione	/	6
Modulo	MOD	6
Addizione	+	7
Sottrazione	-	7
Confronto	<, >, <=, >=	8
Uguaglianza	=	9
Disuguaglianza	<>	9
Booleano/AND bit a bit	AND	10
Booleano/OR esclusivo bit a bit	XOR	11
Booleano/OR bit a bit	OR	12 (meno elevata)

Questi simboli hanno le seguenti funzioni.

- := assegna un'espressione a un simbolo.
- ; indica la fine di una frase (obbligatorio)
- [] sono usate per l'indirizzamento di un array; l'indice dell'array è un intero. Ad esempio, questa frase assegna il valore $j+10$ all'iesimo elemento dell'array "intarray": `intarray[i] := j + 10 ;`
- :: si usano per designare un'etichetta. Ad esempio, questa è un'etichetta: `spray_on::` Un'etichetta deve essere seguita da una frase, nella stessa riga.
- (* *) si usano per designare un commento. Ad esempio: `(*Questo è un commento*)`.

Operatori per i puntatori

Diffida

I puntatori devono essere usati solo da programmatori esperti. Un errato uso dei puntatori può dare luogo a un comportamento imprevedibile e a grandi difficoltà di messa a punto del programma

Il linguaggio ST ha due operatori per i puntatori: & (assegnazione indirizzo) e * (indirizzamento). Questi operatori si usano nelle operazioni di indirizzamento indiretto. Le informazioni per utenti avanzati sull'uso dei puntatori si trovano nell'appendice F.

Sintassi del Testo strutturato

Se si usano gli strumenti resi disponibili dall'editor (barra degli accessori ST e menu di inserimento) per l'introduzione delle frasi e delle funzioni del linguaggio S, la sintassi corretta viene automaticamente proposta dal sistema. Le parole chiave compaiono in lettere maiuscole, mentre per le espressioni e i parametri sostituibili dall'utente sono usate insieme lettere maiuscole e minuscole. Le parti opzionali sono racchiuse tra parentesi quadre. Per maggiori informazioni sull'uso del linguaggio ST, sulle funzioni e sui blocchi funzionali, fare riferimento al **Language Reference**.

Note:

- Tutte le frasi di un programma in testo strutturato devono terminare con un punto e virgola (;).
- I simboli usati in un programma in testo strutturato devono essere dichiarati nel Symbol Manager.

Frase di assegnazione

La frase di assegnazione sostituisce il valore di una variabile con il risultato di una espressione (dello stesso tipo di dato).

Formato

```
Variabile := Espressione;
```

Dove:

Variabile è un simbolo, un array, un particolare elemento di un array, ecc.

Espressione è un singolo valore, un'espressione o un'espressione complessa..

Esempi

Frase di assegnazione booleana:

```
VarBool1 := TRUE;
```

```
VarBool2 := (val <= 75);
```

Assegnazione ad un elemento di un array:

```
Array_1[13] := (RealA /RealB)* PI;
```

Assegnazione a una stringa. Le stringhe letterali devono essere racchiuse tra virgolette singole.

```
String_Val := 'Questa è una costante di tipo stringa';
```

Assegnazione del valore di una funzione:

```
Risultato := SQRT(2);
```

Assegnazione del valore di un blocco funzionale:

Se l'istanza CTU1 del blocco funzionale contatore è in un diagramma RLL, la seguente frase di assegnazione inclusa in un programma in testo strutturato acquisisce il valore corrente del contatore:

```
CurrentValue:=CTU1.CV;
```

Puntatori

- Se pVar1 è un puntatore, la frase che segue assegna a pVAR1 l'indirizzo del dato X.

```
pVar1 := & X;
```
- Se pVar1 è un puntatore, la frase che segue assegna a Y il valore contenuto in Var1, dato che pVar1 contiene l'indirizzo di Var1.

```
Y := * pVar1;
```
- Se pVar1 è un puntatore, la frase che segue assegna a Var1 il valore contenuto in Y.

```
* pVar1 := Y;
```

Frase BREAK

La frase **BREAK** ferma l'esecuzione del programma quando è abilitato il debug (il programma è stato avviato con il comando *Run with Debug*).

Formato

```
BREAK ;
```

Esempio

Se è abilitato il debug, l'esecuzione del programma si arresta alla frase **BREAK**. Le frasi in `ListaDiFrasit2` e le successive possono essere eseguite passo-passo fino alla fine del programma.

```
ListaDiFrasit1 ;
BREAK ;
ListaDiFrasit2 ;
```

Frase CASE

Il costrutto **CASE** offre la possibilità di selezionare la lista di frasi da eseguire da un insieme di liste di frasi possibili, in base ad una certa condizione. La condizione è determinata dal valore di una variabile di tipo integer.

Formato

```
CASE   EspressioneIntera OF
      Int:                               (*Singola*)
      ListaDiFrasit ;
      Int, Int, Int:                     (*Enumerazione*)
      ListaDiFrasit ;
      Int..Int:                           (*Intervallo*)
      ListaDiFrasit ;
[ ELSE                                     (*Parte facoltativa*)
  ListaDiFrasit ; ]
END_CASE ;
```

Dove:

EspressioneIntera Una variabile o un'espressione di tipo ANY_INT.

Int Un intero. E' possibile usare zero o più volte ciascuna delle forme singola, enumerazione o intervallo.

ListaDiFrasit Zero o più frasi in testo strutturato.

Esecuzione

I valori `Int` sono confrontati con `EspressioneIntera`. Viene eseguita la `ListaDiFrase` preceduta dal primo valore `Int` uguale a `EspressioneIntera`. Se `EspressioneIntera` è diversa da tutti i valori `Int` specificati, viene eseguita la `ListaDiFrase` che segue `ELSE`, se esiste, altrimenti non viene eseguita alcuna `ListaDiFrase`. La parte `ELSE` di un costrutto `CASE` è facoltativa.

Esempio

Questo frammento di codice assegna un valore ad una variabile di tipo stringa:

```
CASE ColorSelection OF
  0:
    ColorString := 'Red';
  1:
    ColorString:='Yellow';
  2,3,4:
    ColorString:='Green';
  5..9:
    ColorString:='Blue';
ELSE
    ColorString:='Violet';
END_CASE;
```

Commenti

I commenti permettono di incorporare nel programma annotazioni volte a documentarne il funzionamento. Il compilatore ignora qualsiasi cosa che compaia tra `(*...*)`. I commenti possono essere inseriti nella parte finale di una riga di codice o su una riga separata

Formato

```
(*Questo è un commento in testo libero*)
```

Esempio

```
Risultato:=SQRT(x);(*Usa la funzione radice quadrata*)
```

Frase EXIT

La frase EXIT si usa per uscire da un'iterazione (FOR, WHILE, REPEAT), terminandola prima della sua fine normale. L'esecuzione del programma riparte dalla frase che segue la fine dell'iterazione (END_FOR, END_WHILE, END_REPEAT).

Formato

```
CondizioneDiUscita EXIT;
```

Dove:

CondizioneDiUscita Un'espressione che determina la fine anticipata.

Esempio

Il seguente frammento di codice illustra il funzionamento della frase EXIT. Quando il valore della variabile `numero` supera 500, viene posta fine all'iterazione FOR e l'esecuzione del programma continua con la frase immediatamente successiva a `END_FOR`.

```
numero:=1
FOR contatore := 1 TO 100 DO
    numero := numero * contatore;
    IF numero > 500 THEN EXIT;
END_FOR;
```

Frase IF

Il costrutto IF offre la possibilità di condizionare l'esecuzione di una lista di frasi. La condizione è determinata dal risultato di un'espressione booleana. Il costrutto IF ha due parti facoltative: la prima permette di condizionare l'esecuzione di una lista di frasi alternativa sulla base di una seconda espressione booleana; la seconda permette di eseguire incondizionatamente una terza lista di frasi, se nessuna delle due condizioni è soddisfatta.

Se nessuna delle espressioni booleane è TRUE e il costrutto IF include una frase ELSE, viene eseguita la lista di frasi che segue ELSE. Se la frase ELSE non è presente, non viene eseguita nessuna frase.

Formato

```
IF EspressioneBooleana1 THEN
    ListaDiFrase1;
[ELSEIF EspressioneBooleana2 THEN (*Facoltativa*)
    ListaDiFrase2;]
[ELSE (*Facoltativa*)
    ListaDiFrase3;]
END_IF;
```

Dove:

EspressioneBooleana Qualsiasi espressione di tipo booleano.

ListaDiFrase Un insieme di frasi in testo strutturato.

Esecuzione

Se sono presenti ambedue le parti facoltative, ha luogo la seguente sequenza di valutazioni:

- Se **EspressioneBooleana1** è TRUE, viene eseguita la **ListaDiFrase1**. L'esecuzione del programma continua con la frase che segue immediatamente la parola chiave **END_IF**.
- Se **EspressioneBooleana1** è FALSE e **EspressioneBooleana2** è TRUE, viene eseguita la **ListaDiFrase2**. L'esecuzione del programma continua con la frase che segue immediatamente la parola chiave **END_IF**.
- Se ambedue le espressioni booleane sono FALSE, viene eseguita la **ListaDiFrase3**. L'esecuzione del programma continua con la frase che segue immediatamente la parola chiave **END_IF**.

Se non è presente una parte facoltativa, l'esecuzione del programma continua con la frase che segue immediatamente la parola chiave **END_IF**.

Esempio

Il seguente frammento di codice assegna un testo alla variabile di tipo string Message, in base ai valori in ingresso.

```
IF Input01 < 10.0 THEN
    Message := 'Allarme limite inferiore';
ELSEIF Input02 > 90.0 THEN
    Message := 'Allarme limite superiore';
ELSE
    Message := 'Limiti OK';
END_IF;
```

Frase INCLUDE

La frase INCLUDE incorpora nel programma le frasi contenute in un file esterno, durante la compilazione del programma stesso. Il file esterno può contenere frasi del linguaggio Testo strutturato o del linguaggio Lista di istruzioni.

Formato

```
INCLUDE PercorsoCompletoDelFile;
```

Dove:

PercorsoCompletoDelFile è una stringa che specifica il nome del file esterno.

Esempio

```
INCLUDE 'C:\ST_Files\ST_File1.TXT';
```

Frase FOR

L'iterazione FOR esegue ripetutamente la lista di frasi contenuta entro il costrutto FOR...END_FOR. E' utile quando il numero delle iterazioni può essere determinato in anticipo, come nel caso dell'inizializzazione di un array. Il numero delle iterazioni è determinato dal valore di una variabile di controllo, che viene incrementata (o decrementata) da un valore iniziale a un valore finale dalla frase FOR. Per inizializzare la variabile di controllo è possibile usare un'espressione, ma il suo valore deve essere 0 o positivo. Per difetto, ogni iterazione della frase FOR incrementa di 1 il valore della variabile di controllo. La parte facoltativa del costrutto BY permette di specificare un incremento o un decremento della variabile, specificando un intero positivo o negativo (diverso da 0) o con un'espressione il cui risultato sia un intero positivo o negativo.

La frase FOR esamina la variabile di controllo prima di ciascuna iterazione; le frasi contenute nel costrutto FOR...END_FOR sono eseguite solo se il valore di questa variabile non ha superato il valore finale specificato.

La parola chiave END_FOR provoca l'esecuzione di una scansione degli I/O da parte del sistema alla fine di ciascuna iterazione. In alternativa è possibile usare la parola chiave END_FOR_NOWAIT per iterare senza la scansione degli I/O.

Formato

```
FOR VariabileIntera := Espressione TO Espressione [BY Espressione]
DO
    ListaDiFrase;
END_FOR;
```

Dove:

VariabileIntera Una variabile di tipo integer.

Espressione Un singolo valore, un'espressione o un'espressione complessa dello stesso tipo di dato di **VariabileIntera**.

ListaDiFrase Una serie di frasi in testo strutturato.

Esempi

- Il seguente frammento di codice inizializza un array di 100 elementi assegnando il valore 10 a tutti i suoi elementi. Dato che questa operazione non dipende dagli I/O, viene usata la parola chiave **END_FOR_NOWAIT**.

```
FOR indice := 1 TO 100
    Array01[indice] := 10;
END_FOR_NOWAIT;
```

- Il seguente frammento di codice agli elementi di un array i valori assunti da un punto di I/O in dieci scansioni consecutive. Il valore più recente è assegnato all'elemento dell'array con indice più piccolo. Dato che si desidera eseguire una scansione degli I/O dopo ciascuna iterazione, viene usata la parola chiave **END_FOR**.

```
FOR indice := 10 TO 1 BY -1 DO
    ArrayInput[indice] := Input01;
END_FOR;
```

Richiamo di funzione

Nel linguaggio ST un richiamo di funzione invoca un algoritmo predefinito che esegue un'operazione aritmetica, la manipolazione di una stringa di caratteri, la manipolazione di una stringa di bit, o un'operazione di altro tipo. Il richiamo di funzione consiste nel nome della funzione o del blocco funzionale seguito dai parametri richiesti (in ingresso o in uscita), racchiusi tra parentesi tonde..

Formato del richiamo di funzione

```
NomeDellaFunzione(Parametro1, Parametro2, . . .);
(*Parametri non denominati*)

or

NomeDellaFunzione(P1:=Parametro1, Parametro1, . . .);
(*Parametri denominati*)
```

Formato del richiamo del blocco funzionale

```
NomeDelBloccoFunzionale(P1:=Parametro1,
P2:=Parametro1, . . .);
```

Esempio 1

Questo frammento di codice illustra il richiamo della funzione TAN.

```
Risultato := TAN( AnyReal );
```

Esempio 2

Questo frammento di codice illustra il richiamo di una funzione con parametri denominati.

```
StringB := LEFT(IN:= StringA, L:= VarI);
```

ETICHETTA

L'etichetta si usa solo all'interno di un passo SFC. Se ad una azione è associata un'etichetta contenuta nel codice del passo SFC, l'azione non viene eseguita finché nel codice ST non viene incontrata quella particolare etichetta.

Format

EtichettaAzione:: Frase;

Dove:

EtichettaAzione Il nome di etichetta specificato nell'azione.

Frase Una qualsiasi frase ST.

Esempio

Il seguente frammento di codice provoca l'inizio dell'esecuzione dell'azione con etichetta `EtichettaAzione`.

```
EtichettaAzione:: VarString1:= "Inizio esecuzione azione";
```

Frase REPEAT

L'iterazione REPEAT esegue ripetutamente la lista di frasi contenuta entro il costrutto REPEAT...END_REPEAT, finché non è soddisfatta la condizione di uscita. Questo tipo di iterazione è utile quando la lista di frasi deve essere eseguita almeno una volta.

Formato

```
REPEAT
    ListaDiFrase;
UNTIL EspressioneBooleana  END_REPEAT;
```

Dove:

EspressioneBooleana Qualsiasi espressione il cui risultato è di tipo booleano.

ListaDiFrase Un insieme qualsiasi di frasi del linguaggio ST.

Esecuzione

Viene eseguita la **ListaDiFrase**. Se **EspressioneBooleana** è FALSE, la **ListaDiFrase** viene ripetuta, altrimenti l'iterazione ha fine. La lista di frasi viene eseguita almeno una volta, dato che **EspressioneBooleana** è valutata alla fine del ciclo.

La parola chiave **END_REPEAT** causa una scansione degli I/O da parte del sistema alla fine di ciascun ciclo REPEAT. In alternativa è possibile usare la parola chiave **END_REPEAT_NOWAIT** per iterare senza eseguire la scansione degli I/O.

Nota

E' possibile che si crei un'iterazione infinita che, dato che il sistema di comando ha la priorità più elevata, non restituisce mai il controllo al sistema operativo, specialmente quando si usa la parola chiave **END_REPEAT_NOWAIT**. Evitare questa situazione accertandosi che l'espressione booleana determini sempre la condizione di uscita.

Esempio

Il seguente frammento di codice legge i valori di un array finché non trova un valore maggiore di 5.0 (o non raggiunge il limite superiore dell'array). Dato che deve essere letto almeno un elemento dell'array, è utilizzato il costrutto REPEAT.

```
REPEAT
    Valore:=Array01[Indice];
    Indice:=Indice+1;
UNTIL Valore > 5.0 OR Indice >= LimiteSuperiore
END_REPEAT_NOWAIT;
```

Frase SCAN

La frase **SCAN** sospende l'esecuzione del programma mentre ha luogo una scansione degli I/O.

Formato

```
SCAN;
```

Esempio

```
Frase1;
SCAN;      (*Le frasi 2 e 3 sono eseguite solo dopo
una scansione degli I/O
Frase2;
Frase3;
```

Frase WHILE

L'iterazione **WHILE** esegue ripetutamente la lista di frasi contenuta entro il costrutto **WHILE...END_WHILE**, finché la condizione specificata è **TRUE**. La condizione viene valutata prima di eseguire lista di istruzioni. Questo costrutto è utile quando la lista di frasi non deve necessariamente essere eseguita

Formato

```
WHILE EspressioneBooleana DO
    ListaDiFrase;
END_WHILE;
```

Dove:

EspressioneBooleana Qualsiasi espressione il cui risultato è di tipo booleano.

ListaDiFrase Un insieme qualsiasi di frasi del linguaggio ST.

Esecuzione

Se `EspressioneBooleana` è `FALSE`, l'iterazione ha termine immediatamente, altrimenti, se `EspressioneBooleana` è `TRUE`, viene eseguita la lista di frasi e il ciclo si ripete. Dato che la condizione è valutata all'inizio dell'iterazione, la lista di frasi può anche non essere mai eseguita.

La parola chiave `END_WHILE` causa una scansione degli I/O da parte del sistema alla fine di ciascun ciclo `WHILE`. In alternativa è possibile usare la parola chiave `END_ WHILE _NOWAIT` per iterare senza eseguire la scansione degli I/O.

Nota

E' possibile che si crei un'iterazione infinita che, dato che il sistema di comando ha la priorità più elevata, non restituisce mai il controllo al sistema operativo, specialmente quando si usa la parola chiave `END_ WHILE _NOWAIT`. Evitare questa situazione accertandosi che l'espressione booleana determini sempre la condizione di uscita.

Esempio

Il seguente frammento di codice mantiene attiva l'uscita `Avanti` finché l'ingresso `Limitefinale` rimane `FALSE`. In questo esempio è usato il costrutto `WHILE` perché non c'è motivo di eseguire la lista di istruzioni se l'ingresso `Limitefinale` è `TRUE`. Dato che deve essere eseguita la scansione degli I/O viene utilizzata la parola chiave `END_WHILE`.

```
WHILE NOT LimiteFinale DO
    Avanti := TRUE;
END_WHILE;
Avanti := FALSE;
```

Operatori del linguaggio ST

La tabella che segue elenca gli operatori che possono essere usati in un'espressione. L'ordine di precedenza determina la sequenza con cui sono valutati all'interno dell'espressione. L'operatore con la precedenza più elevata è applicato per primo, seguito dall'operatore con la precedenza più elevata tra i restanti. Gli operatori con eguale precedenza sono valutati da sinistra a destra.

Operatore	Simbolo	Precedenza
Parentesi	()	1
Valutazione di funzione	Identificatore (lista argomenti) es.: LN (A), ABS (X)	2
Elevazione a potenza	** , POW	3
Negazione	-	4
Complemento	NOT	4
Moltiplicazione	*	5
Divisione	/	5
Modulo	MOD	5
Addizione	+	6
Sottrazione	-	6
Confronto	<, >, <=, >=	7
Uguaglianza	=	8
Disuguaglianza	<>	8
Booleo/AND bit a bit	AND	9
Booleo/OR esclusivo bit a bit	XOR	10
Booleo/OR bit a bit	OR	11

Questi simboli hanno le seguenti funzioni.

- := assegna un'espressione a un simbolo.
- ; indica la fine di una frase (obbligatorio)
- [] sono usate per l'indirizzamento di un array; l'indice dell'array è un intero. Ad esempio, questa frase assegna il valore j+10 all'iesimo elemento dell'array "intarray": intarray[i] := j + 10 ;
- :: si usano per designare un'etichetta. Ad esempio, questa è un'etichetta: spray_on:: Un'etichetta deve essere seguita da una frase, nella stessa riga.
- (* *) si usano per designare un commento. Ad esempio: (*Questo è un commento*).

Sommario dei tipi di frasi

Il linguaggio ST dispone dei seguenti tipi di frasi:

- **Assegnazione** - Attribuisce ad un oggetto uno specifico valore.
- **CASE** - Permette l'esecuzione condizionata di un gruppo di frasi.
- **Commento** - Permette di documentare il programma al suo interno.
- **EXIT** - Termina un'iterazione prima che se ne verifichi la condizione di fine.
- **FOR** - Indica che una sequenza di frasi deve essere eseguita ripetutamente sulla base del valore del simbolo di controllo.
- **Richiamo di funzione** - Richiama una funzione per l'esecuzione.
- **IF** - Specifica che una o più frasi devono essere eseguite al verificarsi di una certa condizione.
- **INCLUDE** - Esegue un gruppo di frasi contenute in un file esterno.
- **REPEAT** - Indica che una sequenza di frasi deve essere eseguita ripetutamente, finché un'espressione booleana non diventa TRUE.
- **SCAN** - Ferma l'esecuzione del testo strutturato mentre viene eseguita una scansione degli I/O.
- **WHILE** - Indica che una sequenza di frasi deve essere eseguita ripetutamente finché un'espressione booleana è TRUE.

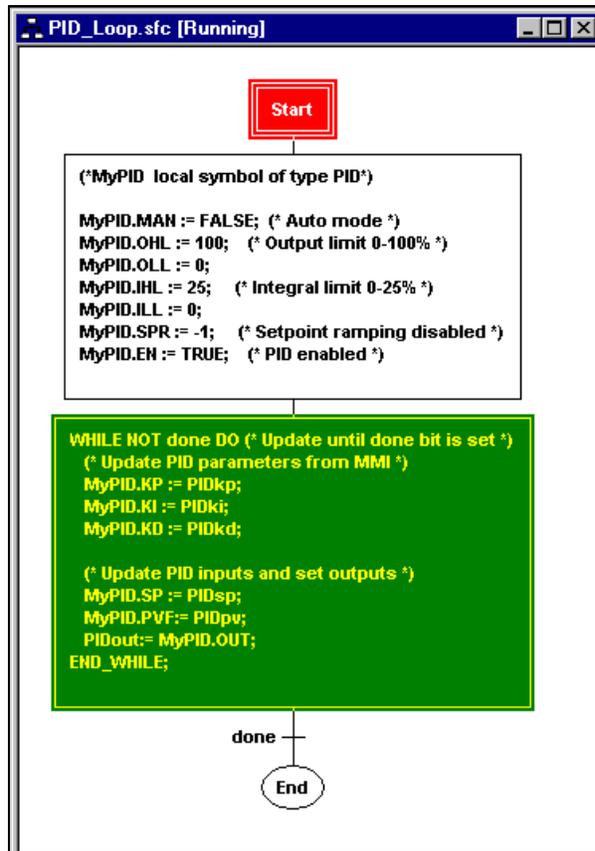
Uso di Pid in un programma applicativo

Che cosa è un Pid

Un PID è un'istruzione che permette la regolazione automatica in anello chiuso di anelli di controllo per processi continui. Per ciascun anello l'istruzione esegue il controllo proporzionale, e, facoltativamente, il controllo integrale, il controllo derivativo o entrambi:

- Controllo proporzionale – causa una variazione del segnale in uscita direttamente proporzionale alla variazione del segnale di errore.
- Controllo integrale – causa una variazione del segnale in uscita che è funzione dell'integrale del segnale di errore rispetto al tempo.
- Controllo derivativo – causa una variazione del segnale in uscita che è funzione della velocità con cui cambia il segnale di errore.

Esempio di PID



Note sull'impiego di PID

1. Un simbolo di tipo PID deve essere definito esclusivamente come variabile locale. Un simbolo PID globale verrebbe risolto una volta per ogni scansione degli I/O per ogni programma in esecuzione. Questo causerebbe instabilità del sistema e i risultati delle uscite PID sarebbero imprevedibili.
2. Se $SPR < 0$, il SET POINT ramping non ha luogo e l'anello PID usa il valore grezzo di SP.
3. Se $MAN = FALSE$ (modo AUTO), OVR è posto uguale a OUT ad ogni valutazione del PID.
4. IHL, ILL e IHLD influenzano l'effetto del trasferimento bumpless.
5. FF può essere usato come polarizzazione statica, come termine del feedforward dinamico o con ambedue i significati.
6. Per usare un PID in un programma ST:
 - A. creare una variabile locale di tipo PID.

- B. fissare i valori iniziali di KP, KI, KD, OHL, OLL, IHL, ILL, IHLD, SPR, MAN, FF.
- C. creare un programma che aggiorni costantemente SP, PVF, e FF (opzionale) e memorizzi OUT.
- D. settare EN per avviare l'aggiornamento del PID

Controllo del flusso dei programmi applicativi RLL e ST

Il PRGCB (Program Control Block = blocco di controllo del programma) permette ad un programma applicativo SFC di compilare e controllare l'esecuzione di altri programmi applicativi SFC e RLL.

Per usare il PRGCB

1. Creare un programma SFC e definire una variabile locale (ad esempio prgcb) di tipo PRGCB.
2. Assegnare a prgcb.NAME una stringa contenente il percorso e il nome di file del programma da controllare.
3. Usare gli ingressi booleani del PRGCB per controllare il programma.
4. Usare le uscite del PRGCB per ottenere informazioni sul programma.

Uso del codice di stato del PRGCB

PRGCB.STATUS permette di monitorare i vari stati di esecuzione del programma controllato. Questa variabile restituisce un valore intero che può essere usato all'interno del programma controllato stesso e come indicatore sullo schermo dell'interfaccia operatore. PRGCB.STATUS indica:

Valore	Definizione
0	Inizializzato
1	File non trovato
2	File aperto
3	File in compilazione
4	Errori di compilazione
5	Compilazione completata
6	Programma in esecuzione
7	Programma fermato

Valore	Definizione
8	Programma completato
9	Programma in interruzione
10	Programma interrotto
11	Programma in stato di errore
12	Programma a un punto di sospensione
13	Programma sospeso
14	Programma riavvolto (funzione REWIND)
15	Programma non riavvolto (funzione REWIND)

Uso della funzione Rewind del PRGCB

Arrivato alla fine della sua esecuzione, il programma controllato non verrà più rieseguito finché non sarà settata la variabile locale di controllo del programma prgcb.REWIND, riavvolgendo così il programma. lo stato di riavvolgimento del programma può essere monitorato mediante la variabile prgcb.STATUS.

Esempio di PRGCB

Il seguente frammento di codice SFC avvia la demo Cookie e la riavvia ogni volta che finisce. Questo esempio usa gli ingressi NAME, REWIND, e RUN, e l'uscita INCYCLE. Per gli altri ingressi e uscite booleani, fare riferimento all'aiuto in linea per il PRGCB.

```
cookie.NAME := "c:\simplicity\pccontrol\cookie\cookie.sfc";
cookie.REWIND := TRUE;
cookie.RUN := TRUE;
WHILE TRUE DO
    (* Quando il programma cookie termina viene riavviato *)
    IF NOT cookie.INCYCLE THEN
        cookie.REWIND := TRUE;
        cookie.RUN := TRUE;
    END_IF;

    (*SCAN;*)
    buzzer := FALSE;
    switch1 := TRUE;
END_WHILE;
```

Monitoraggio e prova dei programmi applicativi e dei simboli

Compilazione di un programma

I programmi possono essere compilati solo con il sistema operativo Windows NT e a patto che sia installato il software esecutivo di PC Control.

Per compilare il programma attivo:

Cliccare su *Execute* e selezionare *Parse*.

Se il programma contiene un errore, compare un messaggio di avvertimento e l'errore è evidenziato. Alla fine della compilazione, il titolo della finestra del file attivo viene aggiornato per indicare lo stato di compilazione.

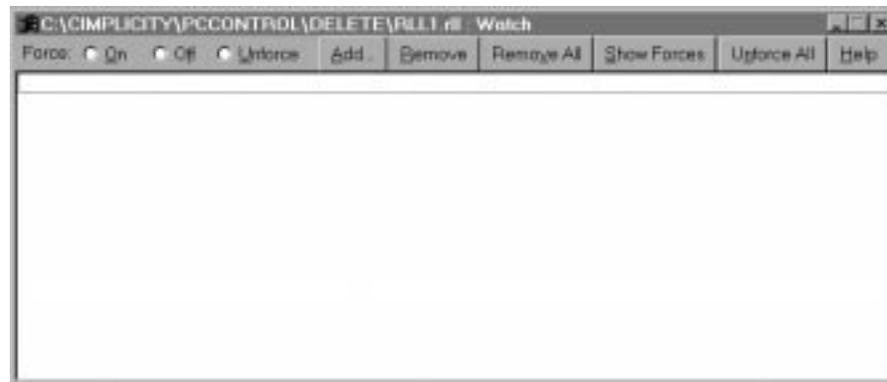
Controllo e forzatura dei simboli

La finestra di controllo, illustrata nelle figure che seguono, visualizza i simboli locali e globali e il loro stato al runtime.

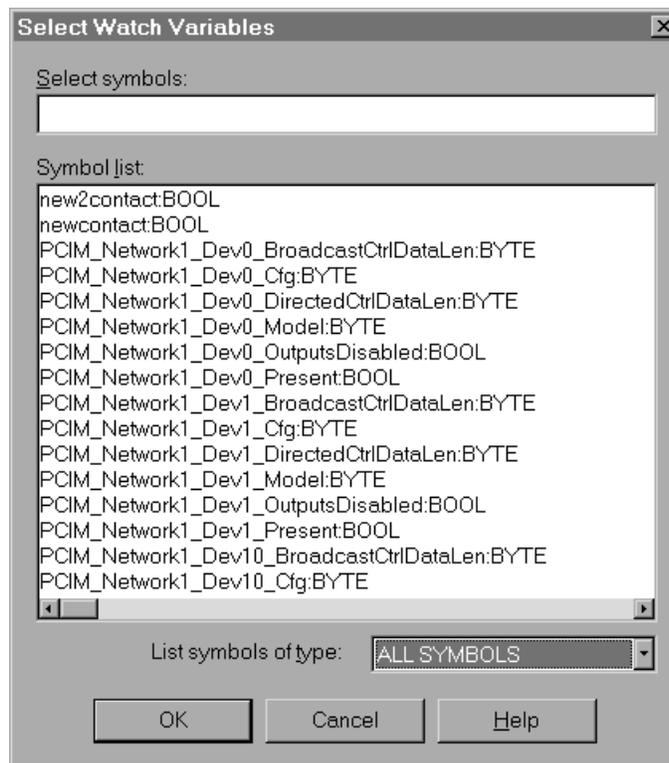
Per visualizzare o nascondere la finestra di controllo:

Cliccare su *View* e selezionare *Watch/Force Variables* oppure cliccare

sull'icona finestra di controllo. 



Per selezionare le variabili da visualizzare nella finestra di controllo, cliccare sul pulsante *Add*. Comparirà la finestra di dialogo *Select Watch Variables*.



Selezionare le variabili da controllare e cliccare su OK. Le variabili selezionate saranno visualizzate nella finestra di controllo.

Esecuzione con debug (RUN with Debug)

L'opzione RUN with Debug permette di eseguire normalmente un programma finché non viene incontrata la frase BREAK. A quel punto il programma si ferma. Dopo essere stato sospeso, il programma può essere definitivamente interrotto o eseguito passo - passo.

Esecuzione passo – passo di un programma

L'opzione Single Step Program permette di eseguire un programma per una scansione completa alla volta. Per poter essere eseguito passo – passo, un programma deve essere stato fermato. Il modo degli altri programmi del progetto non ha importanza.

Per eseguire un programma passo – passo:

1. Selezionare il programma nel Program Editor. Se il programma è in esecuzione, cliccare su *Execute* nella barra del menu e selezionare *Stop*.
2. Aprire una finestra di controllo per il programma e aggiungere tutti i simboli da monitorare durante l'esecuzione del programma. Questo passo è facoltativo.
3. Sulla barra del menu, cliccare su *Execute* e selezionare *Single Step*. Il programma entra nel modo Run per una scansione, poi entra nel modo Break. Durante la scansione del programma sono aggiornati tutti gli I/O controllati dal programma.

Cancellazione del modo Fault e delle condizioni di errore

Per cancellare l'arresto in emergenza Estop e gli eventuali errori di I/O, selezionare *Reset Estop* nel menu *Execute*.

Sezione 5: Programmazione con il linguaggio IL (Lista di istruzioni)

Introduzione

Il linguaggio IL (Instruction List = Lista di istruzioni) è un linguaggio testuale di programmazione definito nello standard IEC 1131-3. Il suo formato è simile a quello dei tradizionali linguaggi assemblativi. Per creare programmi IL standalone si usa l'editor IL. A questo editor, che dispone delle normali funzioni di un editore di testo, quali taglia, copia, incolla, trova, sostituisci, si accede dal Program Editor. L'editor dispone anche di strumenti e di comandi per l'inserimento automatico di frasi e funzioni IL.

Questa sezione descrive l'uso dell'editor IL. Si assume che il lettore abbia familiarità con il funzionamento del Program Editor e che abbia una certa conoscenza del linguaggio IL.

Nota: I programmi IL sono programmi ad esecuzione continua (una volta per scansione).

Apertura di un documento IL

Per aprire un documento esistente, *Open Editor* nel menu *file* del Program Editor e localizzare il documento nella finestra di dialogo *Open*, che compare di conseguenza.

Per aprire un nuovo documento, selezionare *New Editor* nel menu *File* del Program Editor e scegliere *Instruction List Document* nella finestra di dialogo *New*, che compare di conseguenza.

Introduzione delle istruzioni

Introduzione manuale

Le istruzioni possono essere introdotte digitando gli operatori o i richiami di funzione e gli operandi o i parametri associati. La sintassi, gli operatori e i modificatori degli operatori sono descritti nel paragrafo **Descrizione del linguaggio**; per la sintassi dei richiami di funzione si rinvia al **Language Reference**.

Barra degli accessori

La barra degli accessori presenta i comandi in forma grafica. E' un metodo alternativo di introduzione delle istruzioni. La barra degli accessori compare solo se è abilitata commutando *Accessory Bar* nel menu *View*. La figura qui sotto presenta la barra degli accessori, che può essere posizionata dall'utente in base alla propria convenienza.



Menu Insert IL Statements (inserisci frasi IL)

Quando è attivo l'editor IL, il menu *Edit* del Program Editor include una voce *Insert IL Statements* che elenca le frasi (operatori) del linguaggio. Selezionare la frase necessaria, che comparirà automaticamente nella posizione del cursore.

Menu Insert IL Functions (inserisci funzioni IL)

Quando è attivo l'editor IL, il menu *Edit* del Program Editor include una voce *Insert IL Function Calls* che elenca le funzioni standard utilizzabili nel linguaggio Lista di istruzioni. Selezionare la funzione necessaria, che comparirà automaticamente nella posizione del cursore, con la sintassi appropriata. Sostituire i parametri con quelli definiti nel Symbol Manager.

Fare riferimento a **Descrizione del linguaggio** per ulteriori informazioni sull'uso di funzioni e di blocchi funzionali nel linguaggio Lista di istruzioni.

Editazione delle istruzioni

L'editor IL dispone delle normali funzioni di un editor di testo, quali taglia, copia, incolla, trova e sostituisci. Questi comandi si trovano nel menu *Edit* e nel menu contestuale.

Descrizione del linguaggio

Sintassi IL

Un programma IL è costituito da una lista di istruzioni. Ogni istruzione inizia su una nuova riga e può contenere i campi etichetta, operatore, modificatori dell'operatore, operandi e commento, come si vede qui sotto.

Etichetta	Operatore	Operando	Commento
Esempio:	LD	Sym01	(*Carica il valore di Sym01 in un accumulatore*)
	ADD	Sym02	(*Gli somma Sym02*)
			(*La somma è ora il risultato corrente*)

I commenti possono trovarsi solo alla fine della riga. Tra le righe contenenti istruzioni possono trovarsi righe vuote. Il risultato corrente è mantenuto in un accumulatore. Le istruzioni sono interpretate come segue:

risultato_corrente:= risultato_corrente OPERATORE operando

dove risultato_corrente è sempre a sinistra dell'operatore.

Operatori

La sottostante tabella elenca gli operatori del linguaggio Lista di istruzioni

Operatore	Modificatore	Operando	Descrizione
LD	N		Pone il risultato corrente uguale all'operando.
ST	N		Memorizza il risultato corrente nella locazione dell'operando.
S	Nota 1	BOOL	Mette a 1 l'operando booleano.
R	Nota 1	BOOL	Mette a 0 l'operando booleano.
AND	N, (BOOL	AND booleano.
OR	N, (BOOL	OR booleano.
XOR	N, (BOOL	OR esclusivo booleano.
ADD	(Addizione.
SUB	(Sottrazione.
MUL	(Moltiplicazione.
DIV	(Divisione.
GT	(Comparazione per maggiore (>).
GE	(Comparazione per maggiore o uguale (>=).
EQ	(Comparazione per uguale (=).
NE	(Comparazione per non uguale (<>).
LE	(Comparazione per minore (<).
LT	(Comparazione per minore o uguale (<=).
JMP	C, N	LABEL	Salto a etichetta.
CAL	C, N	NAME	Richiamo di funzione o blocco funzionale.
)			Valutazione di operazione differita.
Nota 1. Eseguita solo se il valore di risultato_corrente è 1 booleano. Fonte della tabella: IEC 1131-3 (Parte 3 dello standard IEC 1131 per controllori programmabili), International Electrotechnical Commission.			

Modificatori

Gli operatori possono essere accompagnati dai seguenti modificatori:

N Negazione booleana dell'operando. Ad esempio,

```
ORN Bool1
```

è equivalente a: risultato:= risultato OR NOT Bool1.

C L'istruzione viene eseguita solo se il valore di risultato corrente è 1 booleano. (O 0 booleano se è usato il modificatore N. Ad esempio:

```
LD Valore  
JMPC Sort
```

è equivalente a "se Valore è TRUE, salta a Sort, altrimenti continua normalmente".

(Differisce la valutazione dell'operatore finché non viene incontrato l'operatore parentesi destra ")". Ad esempio:

```
MUL( Num1  
ADD Num2  
)
```

è equivalente a

risultato:= risultato * (Num1 + Num2).

Funzioni e blocchi funzionali

Richiami di funzione

I richiami di funzione possono essere eseguiti usando uno dei seguenti formati:

CALC Richiama una funzione se il valore di risultato_corrente (valore dell'accumulatore) è TRUE (1).

CAL Richiama la funzione in ogni caso.

CALCN Richiama la funzione se il valore di risultato_corrente (valore dell'accumulatore) è FALSE (0).

Qui sotto è illustrata la sintassi del richiamo di funzione mediante **CALC**:

Esempio 1:

CALC ROL(OUT:= VarBit, IN:= BitString, N:= RotateNum)

Esempio 2:

CALC ADD(OUT:= VarNum, Num1, Num2)

Blocchi funzionali

I blocchi funzionali si richiamano con il seguente formato:

CAL Richiama sempre il blocco funzionale.

Qui sotto si possono vedere alcuni esempi della sintassi usata per richiamare un blocco funzionale standard; ctu1 e ton1 sono istanze denominate dei blocchi funzionali CTU e TON.

LD	10	LD	TRUE
ST	ctu1.PV	ST	ton1.IN
LD	TRUE	ST	ton1.EN
ST	ctu1.EN	LD	t#30s
LD	In3	ST	ton1.PT
ST	ctu1.CU	CAL	ton1
CAL	ctu1	LD	ton1.Q
LD	ctu1.Q	ST	Out2
ST	Out3		

Per la descrizione e i parametri (PV, EN, CU, Q, ecc.) dei blocchi funzionali si rinvia al **Language Reference** .

Relazioni con l'accumulatore

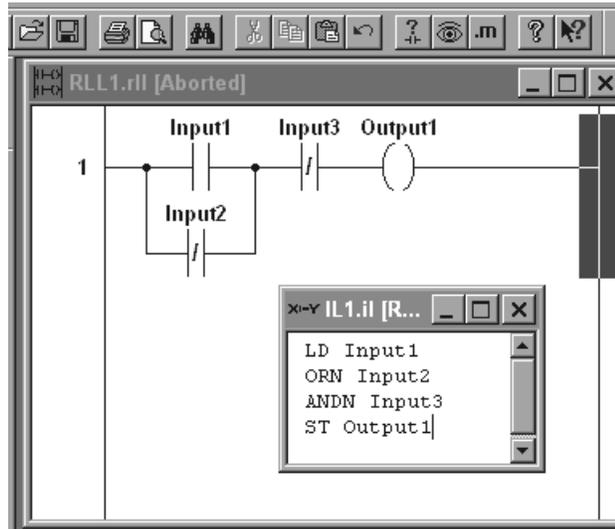
Per i blocchi funzionali, non esistono relazioni fra l'accumulatore e il blocco funzionale. Il blocco funzionale viene sempre richiamato, l'accumulatore non viene passato al blocco funzionale e, dopo il ritorno da un blocco funzionale, l'accumulatore contiene lo stesso valore che conteneva prima del richiamo stesso.

Per le funzioni l'accumulatore non ha effetti sugli ingressi della funzione. Se si usa `CALC`, la funzione non viene richiamata se l'accumulatore è `FALSE`. Se si usa `CALCN`, la funzione non viene richiamata se l'accumulatore è `TRUE`.

Al ritorno dalla funzione, se il valore restituito dalla funzione è di tipo `BOOL`, esso viene caricato automaticamente nell'accumulatore. Se il valore restituito dalla funzione non è di tipo `BOOL`, esso deve essere salvato in una variabile con la sintassi `func1 (OUT:=outvar1)`. In questo caso, il valore di ritorno viene salvato in `outvar1` solo se la funzione viene effettivamente chiamata (`CAL`, `CALC` con accumulatore `TRUE`, o `CALCN` con accumulatore `FALSE`). Se il valore restituito dalla funzione non è di tipo `BOOL`, l'accumulatore assumerà automaticamente il valore invertito del simbolo booleano di errore del sistema (`RTERROR`). Se la funzione viene richiamata e genera un errore, al ritorno dalla funzione l'accumulatore sarà `FALSE`. Se la funzione viene richiamata e non genera errori, l'accumulatore sarà `TRUE`.

Esempi di programmazione

La seguente figura presenta un programma RLL e un programma IL tra loro equivalenti.



Per vari esempi di utilizzo degli operatori del linguaggio Lista di istruzioni, fare riferimento all'aiuto in linea del PC Control.

Esempi di uso degli operatori

```
LD   Format      (* 0: CONVERTE F IN C, 1: CONVERTE C IN F *)
JMPC CtoF       (* SALTA ALLA FUNZIONE CORRISPONDENTE *)

LD   .555        (* CARICA IL MOLTIPLICATORE *)
MUL( Fahrenheit (* EQUIVALE A: (Fahrenheit-32)*.555) *)
SUB  32
)

ST   Celsius     (* MEMORIZZA IL RISULTATO IN Celsius *)
JMP  Done        (* OPERAZIONE COMPLETA, VA ALLA FINE DEL FILE *)
CtoF: LD 32      (* CARICA L'OPERANDO *)
ADD( Celsius     (* EQUIVALE A: (Celsius * 1.8)+32 *)
MUL  1.8
)

ST   Fahrenheit  (* MEMORIZZA IL RISULTATO IN Fahrenheit *)

Done: LD 1       (* CONVERSIONE COMPLETATA *)

LD   Run_Timer   (* CARICA TRUE NEL REGISTRO *)
ANDN FALSE      (* QUESTO VALE TRUE E NON FALSE *)
ST   Enable      (* MEMORIZZA IN Enable *)
```

Esempio di richiamo di blocco funzionale

```
LD Enable      (* CARICA Enable NEL REGISTRO *)
ST ton1.IN     (* MEMORIZZA IL VALORE DEL REGISTRO IN ton1.IN *)
ST ton1.EN     (* MEMORIZZA IL VALORE DEL REGISTRO IN ton1.EN *)
LD t#30s       (* CARICA t#30s NEL REGISTRO *)
ST ton1.PT     (* MEMORIZZA IL VALORE DEL REGISTRO IN ton1.PT *)
CALton1        (* RICHIAMA ton1, CHE GIRERA' PER 30s *)
LD ton1.Q      (* CARICA L'USCITA BOOLEANA NEL REGISTRO *)
ST Enable_Out  (* MEMORIZZA IL VALORE DEL REGISTRO IN Enable_Out *)
```

Sezione 6: Editazione in linea

Che cosa è l'editazione in linea

Il termine editazione in linea si riferisce alla modifica di un programma in esecuzione. Esistono due modi di editazione in linea: senza interferenza e con interferenza. Il modo di editazione in linea che viene stabilito dipende dal tipo delle modifiche eseguite, come è detto nel paragrafo "Regole".

Nell'editazione in linea senza interferenza le modifiche eseguite si riflettono sul programma in esecuzione senza disturbare il funzionamento in runtime. Assumiamo che un programma sia in esecuzione e che a quel programma venga apportata una modifica: se la modifica è eseguibile in linea senza interferenza, l'editor va nel modo editazione in linea senza interferenza. La finestra di controllo dell'editazione in linea è visualizzata con quattro pulsanti attivi:

- | | |
|-------------------------|--|
| Restart Program | Riavvia il programma e resetta gli I/O. Termina l'esecuzione del programma corrente e avvia l'esecuzione del nuovo programma, chiudendo gli I/O nel corso del processo. |
| Activate Changes | Sostituisce la vecchia versione del programma con la nuova senza interferire con la sua esecuzione; la scansione degli I/O continua indisturbata. Se necessario, compila prima le modifiche. Il programma è compilato solo se la data del file SFC o RLL è successiva a quella del programma compilato. Se è stato modificato, il programma viene salvato automaticamente.

Durante il processo di attivazione delle modifiche, i pulsanti della finestra dell'editazione in linea sono disabilitati (durante la compilazione delle modifiche).

Se l'attivazione delle modifiche ha avuto successo, la finestra dell'editazione in linea viene rimossa. Se la compilazione delle modifiche non ha avuto successo, compare il corrispondente messaggio di errore e i pulsanti della finestra dell'editazione in linea sono di nuovo abilitati. |
| Parse Changes | Compila le modifiche, ma non ne avvia l'esecuzione. Se la compilazione delle modifiche ha successo, i pulsanti della finestra sono di nuovo abilitati. Se si verifica un errore di compilazione, compare un messaggio di errore e i pulsanti della finestra sono di nuovo abilitati.

Durante la compilazione delle modifiche, i pulsanti della finestra dell'editazione in linea sono disabilitati. |
| Cancel Changes | Riporta il sorgente allo stato del programma in esecuzione e riparte evidenziando il programma attivo. |

Se un programma è in esecuzione e le modifiche eseguite non possono essere implementate senza interferenza, la finestra dell'editazione in linea è visualizzata con i soli pulsanti *Restart Program* e *Cancel Changes* attivi.

La prima volta che sta per essere implementata una modifica con interferenza, si riceve una richiesta di conferma. Dopo la prima conferma, non vengono dati altri avvertimenti e i pulsanti *Activate Changes* e *Parse Changes* sono disabilitati. Se era stata precedentemente eseguita una modifica senza interferenza, la nuova modifica con interferenza forza la ripartenza del programma.

Regole

Se vengono eseguite modifiche che non possono essere implementate senza interferenza nel programma in esecuzione, viene attivato il modo di editazione in linea con interferenza.

Regola generale

Se sono soddisfatte le seguenti condizioni:

sono apportate modifiche a un programma attivo

le modifiche vengono salvate senza attivarle (modifiche senza o con interferenza)

l'editor viene chiuso

quando l'editor viene nuovamente avviato, esso va direttamente nel modo di editazione in linea con interferenza (la finestra dell'editazione in linea compare con i pulsanti *Activate Changes* e *Parse Changes* disabilitati). Questo significa che il tipo delle modifiche in linea eseguite (con o senza interferenza) non è ricordato da una sessione di editazione all'altra, mentre l'avvenuta modifica del programma è indicata dalle date dei file (file SST rispetto a file SCF, o file RST rispetto a file RLL).

Simboli

E' possibile aggiungere senza interferenza nuove variabili globali. Per farlo, aggiungere le nuove variabili globali mediante il Symbol Editor e cliccare su Apply – le nuove variabili globali diventano attive nel motore di runtime, senza interferenza. Qualsiasi cancellazione o modifica di variabili di memoria globali o qualsiasi aggiunta di strutture globali definite dall'utente, richiederanno la riattivazione della configurazione, terminando tutti i programmi e chiudendo gli I/O. Quando si tenta di cancellare o modificare una variabile di memoria globale o di cambiare le strutture globali definite dall'utente, compare un messaggio che avverte che tutti i programmi in esecuzione verranno interrotti.

- E' possibile aggiungere simboli globali o locali.
- La cancellazione o la modifica di simboli forzano la ripartenza del programma.
- La modifica di strutture locali definite dall'utente forza la ripartenza del programma.

I/O

- La modifica dei driver degli I/O richiederà la riattivazione della configurazione.

Programmi RLL

- Se vengono editati programmi RLL, può essere fatta qualsiasi modifica. L'intero programma viene ricompilato e il programma viene sostituito senza interferenza. I simboli mantengono il loro valore corrente. La scansione degli I/O continua senza interferenza.

Nota

I contatti rilevatori di transizione positiva o negativa non possono causare una transizione la prima volta che l'elemento viene valutato. Per l'editazione in linea questo significa che la prima scansione dopo l'editazione non sentirà la transizione (dato che per il programma modificato questa sarà la scansione iniziale). Questo può causare un problema quando l'elemento è basso e va alto durante la scansione dell'editazione in linea. Questa transizione non verrà rilevata.

Programmi SFC

- Quando si editano programmi SFC, il contenuto di passi, azioni e transizioni può essere cambiato liberamente. La struttura dell'SFC (la rete SFC) non può essere cambiata (questo forzerà la ripartenza del programma), incluso il cambiamento del nome di un elemento (passo, azione o transizione). Nel caso di una editazione in linea senza interferenza, l'intero SFC (inclusi gli SFC macro) è ricompilato e i segmenti di programma che rappresentano i passi, le azioni e le transizioni sono sostituiti senza interferenza; i simboli mantengono il loro valore corrente e i passi, le azioni e le transizioni che sono attivi rimangono tali. La scansione degli I/O continua senza interferenza.
- Per i qualificatori delle azioni SFC è possibile cambiare solo la durata. Qualsiasi altra variazione forzerà la ripartenza del programma.
- Per i passi SFC contenenti testo strutturato, se il passo è completo (è finita l'esecuzione del testo strutturato) esso rimane tale, con le eventuali azioni attive ancora in scansione. Se il passo non è completo (il testo strutturato è ancora in esecuzione) l'esecuzione del testo strutturato ripartirà dall'inizio.

Operazioni su file

L'editazione in linea e le operazioni su file hanno alcune incompatibilità fondamentali. La natura dell'editazione in linea assicura che tutte le variabili, inclusi i blocchi di controllo file mantengono, durante l'editazione in linea, il loro stato corrente. Durante l'editazione in linea, il testo strutturato contenuto in un passo riparte dall'inizio. Se durante l'editazione in linea era in esecuzione un gruppo di comandi ST per la gestione di file, i comandi verranno rieseguiti, ma il file non sarà stato chiuso e riavvolto. Dopo l'editazione in linea, le operazioni su file falliranno o non saranno eseguite correttamente.

Esempio 1

```
STEP1:  
  
FILE_OPEN (fcb, "test.dat");  
  
WHILE (NOT fcb.EOF) DO  
  
    FILE_READ (fcb, struct1);  
  
END_WHILE;
```

Se durante il ciclo WHILE viene eseguita un'editazione in linea, STEP1 sarà resettato e il comando FILE_OPEN fallirà, perché troverà il file già aperto.

Esempio 2

```
STEP1:  
  
FILE_OPEN (fc~b, "test.dat");  
  
STEP2:  
  
WHILE (NOT fc~b.EOF) DO  
  
    FILE_READ (fcb, struct1);  
  
END_WHILE;
```

Se durante il ciclo WHILE viene eseguita l'editazione in linea, STEP2 sarà resettato e il comando FILE_READ funzionerà correttamente. In questo caso esisterà probabilmente un problema di tempificazione, perché il comando FILE_READ non viene interrotto e potrà essere ancora attivo al momento dell'esecuzione dell'editazione in linea

Programmi ST

- I programmi ST non supportano l'editazione in linea senza interferenza. Se vengono apportate modifiche a un file ST mentre il programma è in esecuzione e il file viene salvato, sono abilitati solo i pulsanti dell'editazione in linea con interferenza.

Programmi IL

- Se un file IL viene modificato mentre il programma è in esecuzione e il file viene salvato, sono abilitati i pulsanti dell'editazione in linea.

Capitolo 5

Esecuzione dei programmi applicativi

Dopo aver creato un programma applicativo, per eseguire e monitorare l'applicazione occorre avviare il sistema di Run-time.

Questo capitolo fornisce le seguenti informazioni:

- Come avviare il sistema di Run-time
- Come eseguire un programma applicativo
- Come configurare i programmi per l'esecuzione automatica
- Come monitorare il flusso di corrente
- Come vedere lo stato di un programma applicativo

Sottosistemi di runtime

I sottosistemi di runtime sono il Program Manager, il Program Execution, l'I/O Scanner e l'Event Log. Program Manager, Program Execution e I/O Scanner sono rappresentati visivamente dall'icona Run Time del PC Control. Event Log ha la propria icona.

I sottosistemi Program Execution e I/O Scanner hanno priorità Real-time process; questo significa che ad essi viene dato tempo di CPU prima che alle applicazioni normali, all'aggiornamento del mouse e agli accessi al disco.

Per avviare il sottosistema runtime:

Avviare il Program Editor, poi eseguire una delle seguenti operazioni:

- Selezionare *Startup Runtime Subsystems* dal Program Editor o dal menu *Execute* dall'editor dell'interfaccia operatore.
- Selezionare *Runtime Engine* dal menu *PC Control Applications* nel menu Avvio di Windows.

Esecuzione di un singolo programma

Per eseguire un programma occorre che il sistema operativo sia Windows NT e che sia installato il software esecutivo. Se i sottosistemi di runtime non sono attivi, compare la richiesta di avviare questi sottosistemi.

Esecuzione del programma attivo

Per eseguire il programma attivo:

1. Cliccare su *Execute*.
2. Selezionare *Run*.

Quando il programma comincia a girare, i suoi elementi costitutivi vengono via via evidenziati nella visualizzazione del programma stesso, con il colore selezionato a questo scopo, per indicarne lo stato.

Per eseguire il file attivo con il debug abilitato:

1. Cliccare su *Execute*.
2. Selezionare il comando *Run with Debug*.

Per eseguire un passo del file attivo:

1. Cliccare *Execute*.
2. Selezionare *Single Step*. Se il passo contiene più righe di comandi, viene eseguita la riga di comandi successiva all'interno del passo.

Per eseguire il file attivo con restart:

1. Cliccare su *Execute*.
2. Selezionare *Run with Restart*. Un programma eseguito con restart, inizia a girare automaticamente quando vengono avviati i sottosistemi di runtime.

Cancellazione dell'esecuzione di un programma

Per cancellare l'esecuzione di un programma SFC:

1. Cliccare su *Execute*.
2. Selezionare *Abort*. L'esecuzione del programma viene interrotta e il programma ritorna all'inizio.

Nota

Per inserire un punto di sospensione in un passo SFC, usare la funzione BREAK del linguaggio ST in quel passo.

Configurazione dei programmi per l'esecuzione automatica

In alcune applicazioni può essere necessario avviare automaticamente un programma SFC ogni volta che viene acceso il controllore o ogni volta che vengono avviati i sottosistemi di runtime. Una prima soluzione consiste nel creare un file batch con i comandi appropriati e nell'aggiungere questo file alla cartella Startup di Windows NT. Un'altra soluzione consiste nell'usare il comando Run with Restart.

Avvio dei programmi con un file batch

Nota

Con questo metodo è possibile avviare solo un programma applicativo SFC.

1. Da Windows NT, lanciare un semplice editore di testi per creare un file di testo. Dopo aver creato il file, salvarlo con l'estensione .bat. Il file di testo dovrà contenere le seguenti due righe:

```
start c:\CIMPLICITY PC Control \bin\runtime.exe/RUN
      c:\CIMPLICITY PC Control
\MyProject\Main.sst

start c:\CIMPLICITY PC Control \bin\oicfg.exe
```

Nota

Il comando /RUN deve essere scritto tutto in lettere MAIUSCOLE. Con questo metodo è possibile avviare solo un programma applicativo SFC.

2. Accertarsi che il percorso della cartella \bin sia corretto.
3. Accertarsi che il percorso della cartella del progetto e il nome del programma SFC siano corretti. Per il file del programma usare l'estensione .SST anziché .SFC, perché questa è la versione in testo strutturato del programma SFC di cui ha bisogno il compilatore di runtime invece del file binario .SFC usato dal Program Editor.
4. Usare il Program Editor per accertarsi che il progetto corrente e la configurazione attiva siano correttamente selezionati per questa applicazione. Compilare il file .SFC per accertarsi che il corrispondente file .SST esista e sia aggiornato.
5. Usare l'utilità Interfaccia operatore per selezionare il corretto file .OPI da usare all'accensione.
6. Seguire le istruzioni fornite dall'aiuto in linea di Windows NT per aggiungere il file batch alla cartella NT Startup.
7. Riavviare il computer per provare.

Avvio di programmi con il comando Run With Restart

Per eseguire il file attivo con restart:

1. Cliccare su *Execute*.
2. Sul menu, selezionare il comando *Run with Restart*. Un programma eseguito con restart inizia automaticamente a girare ogni volta che vengono avviati i sottosistemi di runtime del software PC CONTROL. Se viene interrotto o genera errori, il programma non sarà più contrassegnato per girare all'avvio del runtime.

Quando il programma comincia a girare, i suoi elementi costitutivi vengono via via evidenziati nella visualizzazione del programma stesso, con il colore selezionato a questo scopo, per indicarne lo stato.

Monitoraggio del flusso di corrente

Durante l'esecuzione di un programma, i suoi elementi costitutivi vengono via via evidenziati nella visualizzazione del programma stesso, con il colore selezionato a questo scopo, per indicarne lo stato.

Programmi RLL attivi

Durante l'esecuzione di un programma RLL, o quando un'azione e/o una transizione RLL entrano in esecuzione all'interno di un programma SFC, i contatti, le bobine e i blocchi funzionali del programma vengono evidenziati per indicarne lo stato. Un contatto normalmente aperto è evidenziato quando il simbolo booleano ad esso associato è TRUE. Un contatto normalmente chiuso è evidenziato quando il simbolo booleano ad esso associato è FALSE. Un contatto rilevatore di transizione positiva sarà evidenziato al verificarsi di una transizione positiva sul simbolo ad esso associato. Un contatto rilevatore di transizione negativa sarà evidenziato al verificarsi di una transizione negativa sul simbolo ad esso associato. Una bobina di uscita è evidenziata quando il simbolo booleano ad essa associato è TRUE (L'evidenziazione della bobina riflette lo stato del simbolo della bobina stessa). Un blocco funzionale è evidenziato quando è attivo.

Nota

Se il simbolo booleano associato ad un contatto è TRUE, il contatto è evidenziato indipendentemente dalla sua posizione nel rung. In altre parole, un contatto evidenziato non implica necessariamente che anche i contatti che lo precedono siano TRUE.

Programmi SFC attivi

Durante l'esecuzione di un programma SFC, i passi e/o le transizioni attivi sono evidenziati. Se un passo attivo è visualizzato con comandi di processo, a sinistra del comando in esecuzione compare un indicatore di comando attivo.

Visualizzazione dello stato dei programmi applicativi

Per visualizzare lo stato di un programma:

1. Cliccare su *View*.
2. Selezionare *Program Status*. Comparirà la finestra Program Status, che contiene l'elenco dei programmi attivi e il loro stato.

Per visualizzare il programma:

Fare doppio click sul programma da visualizzare.

Per dare un comando relativo ad un programma:

Selezionare il programma sulla lista visualizzata e cliccare sul pulsante corrispondente al comando desiderato: View, Run, Abort, Stop, o Step.

Capitolo 6

Creazione di applicazioni di interfaccia con l'operatore

Il PC Control contiene il software integrato Operator Interface (PC Control GUI), che può essere configurato in modo da permettere il controllo e il monitoraggio di applicazioni di automazione. Data la sua stretta integrazione con il Program Editor, tutti i simboli globali della configurazione attiva sono immediatamente disponibili nel software Operator Interface.

Questo capitolo fornisce le seguenti informazioni:

- Descrizione generale del software Operator Interface
- Come creare gli schermi di interfaccia con l'operatore
- Operazioni sui simboli
 - Editazione dei simboli
 - Attivazione di configurazioni
- Creazione di applicazioni di interfaccia con l'operatore
 - Controlli standard
 - Controlli ActiveX

Per le informazioni relative al Motion Control, fare riferimento all'appendice E.

Sezione 1: Descrizione generale dell'Operator Interface Editor

Avvio del software Operator Interface

Il software Operator Interface permette agli sviluppatori di creare le interfacce operatore di una applicazione e permette all'operatore di usare queste interfacce.

Per avviare il software Operator Interface:

- Selezionare il menu *PC Control Applications* dal menu Avvio di Windows e scegliere *Operator Interface*.
- Se il Program Editor è aperto, scegliere *Operator Interface* nel menu *Tools* del Program Editor.

Il software Operator Interface parte nel modo attivazione. L'interfaccia con l'operatore è definita dall'ultimo file di interfaccia aperto per il progetto corrente. Lo schermo iniziale definito in quel file è il primo schermo di interfaccia visualizzato.

Se quando viene avviato il software Operator Interface i sottosistemi di runtime non stanno girando, compare un prompt che dà l'opportunità di avviarli.

Livelli di accesso

I livelli di accesso e i codici di accesso sono utilizzati per prevenire l'accesso non autorizzato ai programmi applicativi e ai dati di configurazione. Esistono cinque livelli di accesso, da 0 a 4, con un diverso codice di accesso per ciascuno di essi. La seguente tabella descrive i privilegi di ciascun livello:

Con il livello di accesso:	E' possibile:
0	Attivare gli schermi dell'operatore e usare i controlli. Selezionare ed eseguire i programmi SFC.
1	Eseguire e arrestare i programmi RLL continui per l'ambiente e la logica di macchina. Visualizzare i programmi RLL e SFC.
2	Editare i programmi SFC. Eseguire le funzioni di gestione progetto.
3	Editare i programmi RLL. Modificare l'interfaccia operatore. Modificare i file di configurazione del sistema.
4	Cambiare le password per l'accesso ai livelli da 1 a 4..

I valori per difetto dei codici di accesso si trovano nelle istruzioni di installazione (su carta) consegnate insieme al software PC Control.

Introduzione del codice di accesso

Il Program Editor e l'Operator Interface richiedono un codice di accesso. Quando viene avviato, il Program Editor presenta automaticamente la tastiera per la specifica del codice di accesso.

Per introdurre il codice di accesso:

Software	Procedura
Program Editor	Cliccare sui numeri della tastiera visualizzata che rappresentano il proprio codice di accesso e cliccare su OK.
Operator Interface	Cliccare su <i>Access</i> nella barra del menu e selezionare <i>Password</i> . Comparirà una tastiera. Cliccare sui numeri della tastiera visualizzata che rappresentano il proprio codice di accesso e cliccare su OK.

Per mettere a 0 il codice di accesso:

Cliccare sul pulsante *Cancel* della tastiera di accesso.

Variazione dei codici di accesso

I codici di accesso possono essere cambiati solo dall'interno dell'interfaccia operatore, specificando un codice di accesso al livello 4. Esiste un codice di accesso per ciascun livello.

Per cambiare la password di un livello di accesso:

1. Sulla barra del menu selezionare *Access - Password*. Comparirà una tastiera.
2. Cliccare quattro volte sul carattere '*'. Comparirà il messaggio: "Enter Access Level to Change".
3. Cliccare sul numero del livello di accesso per il quale si desidera eseguire la modifica e cliccare su *OK*.
4. Cliccare il nuovo numero di quattro cifre. Comparirà il messaggio: "Enter new password again".
5. Cliccare di nuovo la nuova password e cliccare su *OK*. Se la password cliccata corrisponde a quella specificata nel passo 4, essa diventa la nuova password del livello di accesso selezionato.

Nota

Per azzerare il codice di accesso cliccare sul pulsante CANCEL della tastiera visualizzata.

Se, nel modo operator interface edit, ai controlli dell'operatore si associano le funzioni di controllo, è possibile specificare un livello di accesso per controllare l'uso di ciascun controllo dell'operatore.

Modi attivazione e edit

Il software Operator Interface dispone di due modi: il modo attivazione e il modo edit. Nel modo attivazione, questo software esegue e controlla le funzioni dell'applicazione. Nel modo edit, esso permette di editare, creare, copiare, rinominare o cancellare schermi, di posizionare controlli su questi schermi e di collegare i controlli a programmi e a simboli. Per selezionare il modo edit è necessario avere un livello di accesso 3 o 4.

Il software Operator Interface parte nel modo attivazione.

Commutazione dei modi del software Operator Interface

Il software Operator Interface parte nel modo attivazione.

Per passare dal modo attivazione al modo edit:

La creazione e la modifica degli schermi dell'interfaccia con l'operatore richiedono un livello di accesso 3 o 4.

1. Nel menu *Tools*, scegliere *Operator Interface Screen Edit*.
2. Introdurre la password per l'accesso al livello 3 o 4.

Nel modo edit è possibile inserire, editare, spostare e dimensionare i controlli.

Per passare dal modo edit al modo attivazione:

Cliccare su *Execute* e selezionare *Activate Screens*.

Dopo l'attivazione degli schermi dell'operatore, è possibile usare i controlli per gestire le funzioni dell'applicazione. Quando sono attivi gli schermi non è possibile editare i controlli.

Controllo di programmi RLL mediante l'interfaccia con l'operatore

Il Continuous Logic Manager è un programma eseguibile che può essere utilizzato per controllare i programmi RLL di un progetto. Per accedere al Continuous Logic Manager da uno schermo dell'interfaccia operatore, aggiungere allo schermo un pulsante associato alla funzione EXECUTE. Selezionare il Continuous Logic Manager come file eseguibile da lanciare. Fare riferimento a "Pulsanti" per i dettagli.

Durante il runtime, il Continuous Logic Manager visualizza la lista dei programmi RLL del progetto attivo che sono in esecuzione e la lista dei programmi RLL del progetto attivo che non sono in esecuzione. I programmi RLL in esecuzione sono visualizzati nella lista *Running RLL Programs*. I programmi RLL non in esecuzione sono visualizzati nella lista *Project RLL Programs*.

Per mettere in esecuzione un programma RLL

1. Selezionare il programma nella lista *Project RLL Programs*.
2. Cliccare su *Run*. Il programma selezionato viene tolto dalla lista *Project RLL Programs* e inserito nella lista *Running RLL Programs*.

Per mettere in esecuzione con restart un programma RLL

1. Selezionare il programma nella lista *Project RLL Programs*.
2. Cliccare su *Run with Restart*. Il programma selezionato viene tolto dalla lista *Project RLL Programs* e inserito nella lista *Running RLL Programs*. Un programma lanciato con restart inizierà automaticamente a girare all'avvio del Runtime.

Per cancellare l'esecuzione di un programma RLL

1. Selezionare il programma nella lista *Running RLL Programs*.
2. Cliccare su *Abort*. Il programma selezionato viene tolto dalla lista *Running RLL Programs* e inserito nella lista *Project RLL Programs*.

Per editare un programma RLL

1. Selezionare il programma nella lista *Project RLL Programs* o nella lista *Running RLL Program*.
2. Cliccare su *Edit*. Il Program Editor si attiva con il programma selezionato in primo piano. Se il programma RLL è in esecuzione, il Program Editor lo evidenzierà.

Sezione 2: Schermi dell'interfaccia con l'operatore

Funzioni del software Operator Interface

Creazione di un nuovo file interfaccia operatore

Un file interfaccia operatore contiene uno o più schermi. Un progetto può contenere più di un file interfaccia operatore. Lo schermo iniziale dell'ultimo file interfaccia operatore è lo schermo attivo quando viene avviato il software Operator Interface.

Per creare un nuovo file interfaccia operatore:

- Selezionare *New* nel menu *File* o usare il corrispondente pulsante della barra degli strumenti.

Il nome file per difetto di un file interfaccia operatore è *OPI* e il nome assegnato per difetto allo schermo è *Operator Interface*.

Apertura di un file interfaccia operatore

Per aprire un file interfaccia operatore esistente:

- Selezionare *Open* nel menu *File* o usare il corrispondente pulsante della barra degli strumenti.

Salvataggio di un file interfaccia operatore

Il salvataggio di un file interfaccia operatore salva tutti gli schermi contenuti nel file.

Per salvare il file:

- Selezionare *Save* nel menu *File* o usare il corrispondente pulsante della barra degli strumenti.

Quando un file viene salvato per la prima volta, compare la finestra di dialogo *Save As*, che permette di attribuire un nome al file.

Per salvare il file con un nuovo nome o in un'altra cartella:

- Selezionare *Save as* nel menu *File*. Nella finestra di dialogo che compare specificare la nuova cartella e/o il nuovo nome del file.

Operazioni relative agli schermi

Creazione di un nuovo schermo di interfaccia con l'operatore

Per creare un nuovo schermo dell'interfaccia con l'operatore:

1. Selezionare *New Screen* nel menu *Edit*. Comparirà la finestra di dialogo *Enter New Screen Name*.
2. Digitare il nome del nuovo schermo e cliccare su *OK*. Viene creato uno schermo operatore in bianco il cui nome compare nella barra del titolo.

Cancellazione di uno schermo

Un file interfaccia operatore deve contenere almeno uno schermo. Se si tenta di cancellare l'ultimo schermo di un file compare un messaggio di errore.

Per cancellare uno schermo dell'interfaccia con l'operatore:

1. Selezionare *Delete Screen* nel menu *Edit*. Comparirà la finestra di dialogo *Select Screen to Delete*.
2. Digitare il nome dello schermo che si desidera cancellare o selezionarlo sulla lista che compare e cliccare su *OK*.

Se lo schermo cancellato era lo schermo iniziale, viene visualizzata la finestra di dialogo *Select New Startup Screen*. Digitare il nome del nuovo schermo iniziale o selezionarlo nella lista che compare.

Copia di uno schermo

Per copiare uno schermo dell'interfaccia con l'operatore:

1. Selezionare *Copy Screen* nel menu *Edit*. Comparirà la finestra di dialogo *Select Screen to Copy*.
2. Selezionare lo schermo e cliccare su *OK*. Comparirà la finestra di dialogo *Enter New Screen Name*.
3. Digitare il nome del nuovo schermo e cliccare su *OK*. Viene creato un nuovo schermo con lo stesso contenuto dello schermo originale.

Rinominazione di uno schermo

Per rinominare un schermo dell'interfaccia con l'operatore:

1. Selezionare *Rename Screen* nel menu *Edit*. Comparirà la finestra di dialogo *Select Screen to Rename*.
2. Selezionare lo schermo da rinominare e cliccare su *OK*. Comparirà la finestra di dialogo *Enter New Screen Name*.
3. Digitare il nuovo nome dello schermo e cliccare su *OK*.

Selezione dello schermo iniziale

Lo schermo iniziale è lo schermo che compare (per la configurazione attiva del progetto corrente) quando viene avviato il software Operator Interface.

Per selezionare lo schermo iniziale:

1. Selezionare *Start Screen* nel menu *Edit*. Comparirà la finestra di dialogo *Select New Startup Screen*.
2. Selezionare lo schermo desiderato e cliccare su *OK*. Lo schermo selezionato diventa lo schermo iniziale.

Selezione dello schermo da editare

Per poter editare uno schermo, occorre prima selezionarlo.

Per selezionare lo schermo da editare

1. Cliccare sulla voce di menu *Screens*. Comparirà l'elenco degli schermi contenuti nel file interfaccia operatore corrente.
2. Selezionare lo schermo desiderato. Lo schermo selezionato viene visualizzato per l'editazione.

Sezione 3: Operazioni relative ai controlli

Aggiunta di controlli

Per aggiungere un controllo:

1. Selezionare *New Standard Control* nel menu *Edit*. Comparirà una lista di controlli standard.
2. Selezionare il controllo da aggiungere. La forma del cursore cambia per indicare il controllo selezionato.
3. Posizionare il cursore sullo schermo e cliccare per aggiungere il controllo.

Nota

Molti controlli possono anche essere selezionati nella barra degli strumenti relativa ai controlli.

Per annullare l'operazione premere Esc.

Per informazioni relative all'aggiunta di controlli ActiveX, fare riferimento a "Inserimento di controlli ActiveX".

Editazione di controlli

Per editare un controllo standard:

1. Selezionare il controllo con uno dei seguenti metodi:
 - Fare doppio click sul controllo.
 - Selezionare il controllo desiderato e premere Invio.
 - Selezionare il controllo desiderato e scegliere *Edit Standard Control* nel menu *Edit* o nel menu contestuale.
2. Comparirà una finestra di dialogo corrispondente al controllo selezionato. Per le informazioni relative ai controlli standard, fare riferimento a "Controlli Standard".

Nota

Per le informazioni relative all'editazione dei controlli ActiveX, fare riferimento a "Editazione di controlli ActiveX" nella sezione "Controlli ActiveX".

Selezione di Controlli

Per selezionare un solo controllo:

- Posizionare il cursore del mouse sul controllo e cliccare. Il controllo è evidenziato.

Per selezionare più controlli con una cornice di selezione:

Questo metodo consiste nel tracciare una cornice rettangolare intorno ai controlli da selezionare.

1. Posizionare la freccia di selezione su uno dei vertici del rettangolo. Cliccare e trascinare fino al vertice opposto.
2. Rilasciare il pulsante del mouse. Verranno selezionati tutti i controlli che sono interamente entro l'area rettangolare.

Per selezionare più controlli con la tastiera e il mouse:

- Tenere premuto il tasto Ctrl o il tasto Shift e selezionare individualmente i controlli desiderati.

Per deselectare un controllo in un gruppo di controlli selezionati:

- Tenere premuto il tasto Ctrl o il tasto Shift, portare il cursore del mouse sul controllo e cliccare.

Spostamento di controlli

Per spostare i controlli con il mouse:

1. Selezionare i controlli.
2. Portare il cursore su uno dei controlli selezionati. Cliccare e trascinare il controllo selezionato nella posizione desiderata. Un bordo nero indica il nuovo perimetro dell'area del gruppo di controlli.
3. Quando i controlli sono nella posizione desiderata, rilasciare il pulsante del mouse.

Per annullare l'operazione, premere Esc.

Per spostare i controlli con la tastiera:

1. Selezionare i controlli.
2. Usare i tasti freccia per spostare i controlli pixel per pixel nella direzione del tasto freccia.

Dimensionamento di controlli

Per dimensionare un controllo con il mouse

1. Selezionare il controllo. Se il controllo può essere dimensionato, sul suo perimetro compaiono dei quadratini di ridimensionamento.
2. Portare il cursore su uno di questi quadratini. Il cursore assume la forma della freccia di ridimensionamento, indicante la direzione nella quale il componente può essere ridimensionato.
3. Cliccare e trascinare il quadratino selezionato per allargare o restringere il controllo. Un bordo nero indicherà il nuovo perimetro del controllo.
4. Quando la dimensione del controllo è quella desiderata, rilasciare il pulsante del mouse.

Per annullare questa operazione, premere **Esc**.

Copia, taglia e incolla

Le operazioni copia, taglia e incolla sono spesso usate insieme. Copia salva gli oggetti selezionati negli appunti. Taglia cancella gli oggetti selezionati dopo averli salvati negli appunti. Incolla inserisce il contenuto degli appunti nello schermo corrente. Le operazioni copia, taglia e incolla possono essere eseguite tra schermi diversi.

Benché queste operazioni funzionino in modo simile a quello di altre applicazioni Windows, il formato degli appunti non è in questo caso compatibile con quello di altre applicazioni. I controlli possono solo essere copiati e incollati tra schermi.

Per copiare un controllo:

- Selezionare il controllo, poi scegliere *Copy* nel menu *Edit* o nel menu contestuale, oppure premere i tasti **Ctrl+C**.

Per tagliare un controllo:

- Selezionare il controllo, poi scegliere *Cut* nel menu *Edit* o nel menu contestuale, oppure premere i tasti **Ctrl+Z**.

Per incollare il contenuto degli appunti:

1. Visualizzare lo schermo nel quale deve essere incollato il controllo (se non è lo schermo corrente).
2. Scegliere *Paste* nel menu *Edit* o nel menu contestuale, oppure premere i tasti **Ctrl+V**.

I controlli copiati dagli appunti su uno schermo operatore, assumono la stessa posizione che occupavano quando sono stati copiati o tagliati.

Cancellazione di controlli

Per cancellare uno o più controlli da uno schermo operatore:

1. Selezionare i controlli.
2. Premere il tasto **Del** o selezionare *Delete* nel menu *Edit* o nel menu contestuale.

Allineamento di controlli

I controlli possono essere allineati sul lato sinistro, sul lato destro, sul lato superiore o sul lato inferiore.

Lato sinistro	I controlli selezionati si allineano sul lato sinistro del controllo più a sinistra.
Lato destro	I controlli selezionati si allineano sul lato destro del controllo più a destra.
Lato superiore	I controlli selezionati si allineano sul lato superiore del controllo più in alto.
Lato inferiore	I controlli selezionati si allineano sul lato inferiore del controllo più in basso

Per allineare i controlli:

1. Selezionare i controlli da allineare (devono essere selezionati almeno due controlli).
2. Selezionare *Align Control* nel menu *Layout*, poi scegliere l'allineamento desiderato: *Left*, *Right*, *Top* o *Bottom* (lato sinistro, lato destro, lato superiore, lato inferiore, rispettivamente).

Spostamento dei controlli davanti/dietro

Un controllo può essere spostato davanti o dietro ad altri controlli sovrapposti.

Per spostare un controllo davanti o dietro:

1. Selezionare il controllo.
2. Usare uno dei seguenti due metodi:
 - Per portare un controllo davanti ad un gruppo di controlli sovrapposti, scegliere *Move to Front* nel menu *Layout*.
 - Per portare un controllo dietro ad un gruppo di controlli sovrapposti, scegliere *Move to back* nel menu *Layout*.

Sezione 4: Operazioni sui simboli

Il Symbol Manager può essere usato direttamente dall'interno dell'Operator Interface Editor. Però, il Symbol Manager non può essere aperto contemporaneamente nell'Operator Interface Editor e nel Program Editor. Quando il Symbol Manager viene aperto nell'Operator Interface Editor, esso opera solo sui simboli globali. E' accessibile dalla barra degli strumenti, dal menu, e dalle finestra di dialogo che permettono di selezionare i simboli.

Nota

Nei controlli dell'interfaccia con l'operatore possono essere usati soltanto i simboli globali.

Editazione di simboli

Per editare i simboli globali:

- Selezionare *Symbol Manager* nel menu *Tools*. Fare riferimento a **Symbol Manager** nella **Programming Guide** per le informazioni relative all'uso del Symbol Manager.

Attivazione di configurazioni

Per rendere disponibili nel Program Editor le modifiche apportate ai simboli, deve essere attivata la configurazione.

Per attivare la configurazione usare uno dei seguenti metodi:

- Cliccare su *Apply* nel Symbol Manager .
- Selezionare *Activate Config* nel menu *Tools*.

Per salvare la configurazione:

- Selezionare *Save Config* nel menu *Tools*.

Sezione 5: Creazione di applicazioni di interfaccia con l'operatore

Controlli standard

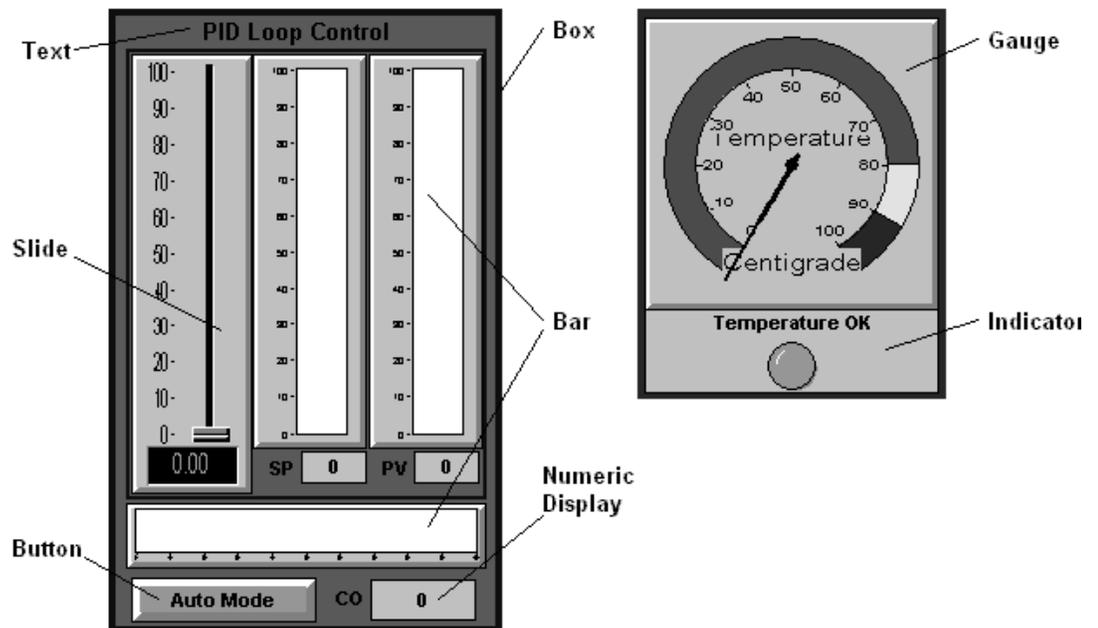
Introduzione

La tabella che segue elenca e descrive brevemente i controlli standard e gli elementi visuali utilizzabili in uno schermo dell'interfaccia con l'operatore.

Controllo	Descrizione
Bar	Questo controllo visualizza una scala e un indicatore a colonna. L'indicatore segue il valore corrente del simbolo assegnato al controllo. L'orientamento del controllo può essere orizzontale o verticale.
Bitmap	Questo è un elemento visuale che può essere usato per descrivere l'applicazione o aggiungere interesse visivo allo schermo.
Box	Questo è un elemento visuale che può essere usato per raggruppare controlli, dare un bordo a un testo o a una bitmap, mettere in evidenza parti dello schermo, o comunque rendere visivamente interessante uno schermo.
Click (multi-state) Button	Questo controllo esegue una funzione quando viene cliccato. Le funzioni includono il settaggio o l'azzeramento di un simbolo booleano; la visualizzazione di uno schermo; la selezione, l'esecuzione, la sospensione o l'interruzione di un programma; l'ottenimento di dati dall'operatore; il richiamo di un file eseguibile. Il pulsante può essere configurato a uno o a due stati, con stato sostitutivo. Possono essere programmate più funzioni per ciascuno stato e il pulsante eseguirà le funzioni in base al suo stato corrente. Il pulsante può essere configurato per il cambio di stato automatico o in base a un simbolo booleano.
Continuous Button	Questo controllo esegue una funzione quando viene cliccato. Le funzioni includono il settaggio o l'azzeramento di un simbolo booleano; la visualizzazione di uno schermo; la selezione, l'esecuzione, la sospensione o l'interruzione di un programma; l'ottenimento di dati dall'operatore; il richiamo di un file eseguibile. Le funzioni possono essere programmate in modo da essere eseguite quando il pulsante viene premuto e quando il pulsante viene rilasciato.
Gauge	Questo controllo visualizza una scala e un indicatore rotante. L'indicatore segue il valore del simbolo assegnato al controllo. Può essere visualizzato in quattro stili diversi.
Indicator	Questo è un indicatore di stato il cui aspetto (testo, colore, ecc.) cambia in base al valore di un simbolo di tipo BOOL o BYTE.
Numeric Display	Questo controllo si usa per visualizzare il valore di un simbolo.

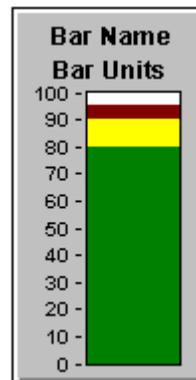
Pannello Selected Program Status	Questo pannello visualizza il nome e lo stato corrente (ad esempio: in esecuzione, sospeso, ecc.) del programma selezionato. Questo programma è quello selezionato da un comando SELECT PROGRAM associato a un pulsante (Click o Continuous Button).
Slide	Questo controllo dà la possibilità di variare in modo continuo il valore di un simbolo. La variazione del valore del simbolo assegnato al controllo si attua spostando l'indicatore di quest'ultimo.
Text	Questo elemento si usa per etichettare controlli o gruppi di controlli, documentare le funzioni dello schermo o altrimenti descrivere lo schermo e le sue funzioni.

La sottostante figura presenta un esempio di schermo che usa controlli standard ed elementi visuali.

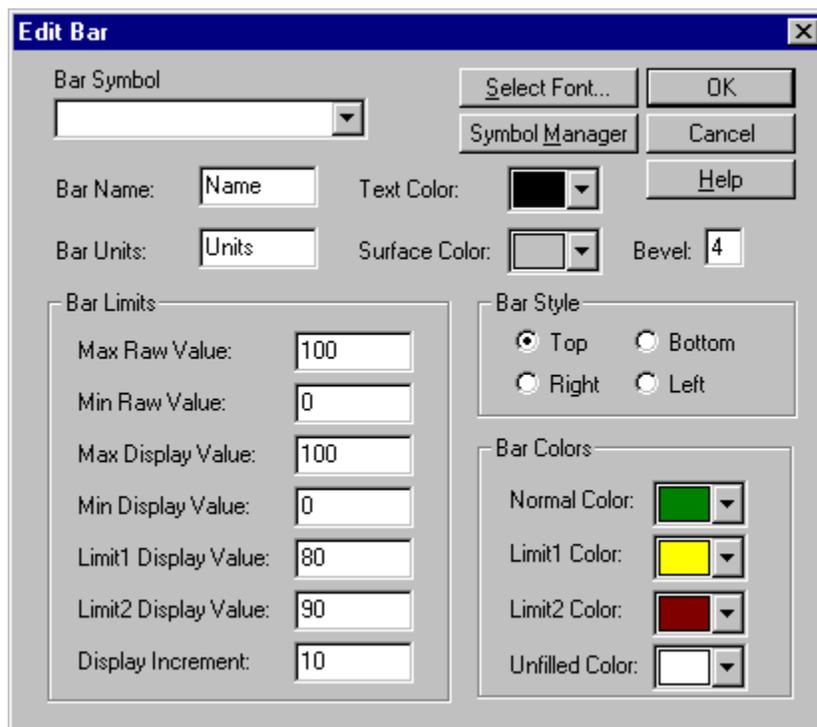


Bar

Questo controllo visualizza una scala e un indicatore a colonna. L'indicatore segue il valore corrente del simbolo assegnato al controllo. Il colore della colonna cambia con il variare del valore del simbolo, riempiendo l'area della colonna corrispondente ai valori minori di quello del simbolo. E' possibile stabilire dei limiti in modo che il colore della colonna cambi quando il valore del simbolo si avvicina a questi limiti. L'orientamento del controllo può essere orizzontale o verticale.



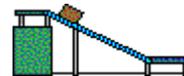
La finestra di dialogo *Edit Bar* è visibile nella sottostante figura ed è descritta nella tabella che segue.



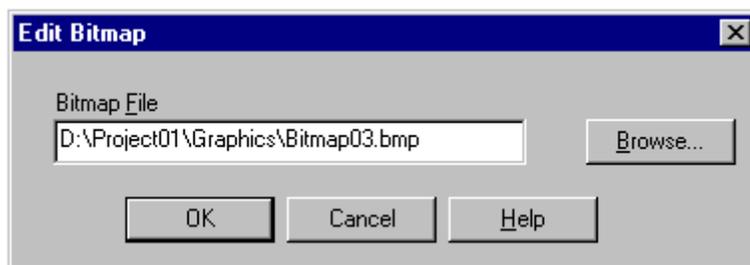
Campo	Descrizione
Bar Symbol	Specifica il simbolo il cui valore è seguito dall'indicatore a colonna. Se è necessario, è possibile definire ex novo questo simbolo aprendo il Symbol Manager con il corrispondente pulsante.
Bar Name	Specifica il nome del controllo. Serve solo per la documentazione dello schermo.
Bar Units	Specifica le unità del controllo. Serve solo per la documentazione dello schermo.
Text Color	Specifica il colore che deve essere usato per il nome, le unità e la numerazione della scala del controllo.
Surface Color	Specifica il colore dello sfondo del controllo.
Bevel	Questa proprietà influenza l'aspetto del bordo del controllo. I valori possibili di Bevel vanno da 0 a 6. Se si specifica 0, il controllo è piatto. L'aumento del valore di Bevel dà al controllo un aspetto tridimensionale.
Bar Limits	Limiti dell'indicatore a colonna.
Max Raw Value Min Raw Value	Specificano i valori massimo e minimo ai quali risponde l'indicatore a colonna. Essi limitano il campo dei valori rappresentabili dall'indicatore a colonna. Se il valore del simbolo associato al controllo supera il limite superiore, la colonna viene riempita completamente, mentre qualsiasi valore al di sotto del limite inferiore la vuota. Se si invertono i valori dei limiti massimo e minimo, può essere seguito l'inverso del valore del simbolo. La logica di massimo e minimo è invertita.
Max Display Value Min Display Value	Specifica i valori massimo e minimo della scala.
Limit1 Display Value Limit2 Display Value	Specifica i limiti ai quali cambia il colore dell'indicatore a colonna (rispetto ai valori visualizzati); ad esempio, per indicare una condizione di allarme.
Display Increment	Specifica l'incremento della numerazione della scala.
Bar Style	Lo stile specifica l'orientamento del controllo e il verso di riempimento dell'indicatore a colonna.
Bar Colors	Possono essere specificati quattro colori: Normal – il colore dell'indicatore a colonna che rappresenta il valore del simbolo prima che esso raggiunga un limite. Limit 1 – il colore dell'indicatore a colonna dopo che ha raggiunto il primo limite. Limit 2 - il colore dell'indicatore a colonna dopo che ha raggiunto il secondo limite. Unfilled – il colore dello sfondo dell'indicatore a colonna.
Select Font	Questo pulsante visualizza la finestra di dialogo <i>Font</i> , che permette di specificare il tipo, lo stile, la dimensione e il colore del testo del controllo (nome, unità, e numerazione della scala).
Symbol Manager	Questo pulsante apre il Symbol Manager.

Bitmap

Una bitmap è un elemento visuale che può essere usato per descrivere l'applicazione o per aggiungere interesse visivo allo schermo. Una bitmap non oscurerà mai un controllo o un testo.



La finestra di dialogo *Edit Bitmap* è visibile nella sottostante figura ed è descritta nella tabella che segue.



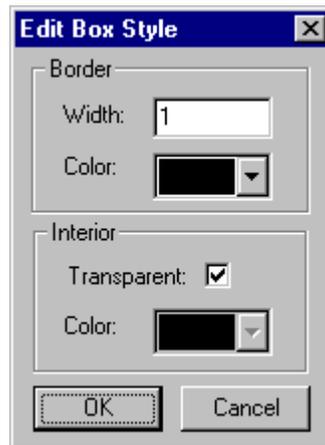
Campo	Descrizione
Bitmap File	Digitare il percorso e il nome del file bitmap, o cliccare su Browse per localizzare il file. E' possibile usare solo file bitmap di Windows (.BMP).

Box

Questo è un elemento visuale che può essere usato per raggruppare controlli, dare un bordo a un testo o a una bitmap, mettere in evidenza parti dello schermo, o comunque rendere visivamente interessante uno schermo. Un box ha un bordo e un colore di riempimento opzionali. Un box non oscurerà mai un controllo o un testo.



La finestra di dialogo *Edit Box* è visibile nella sottostante figura ed è descritta nella tabella che segue.



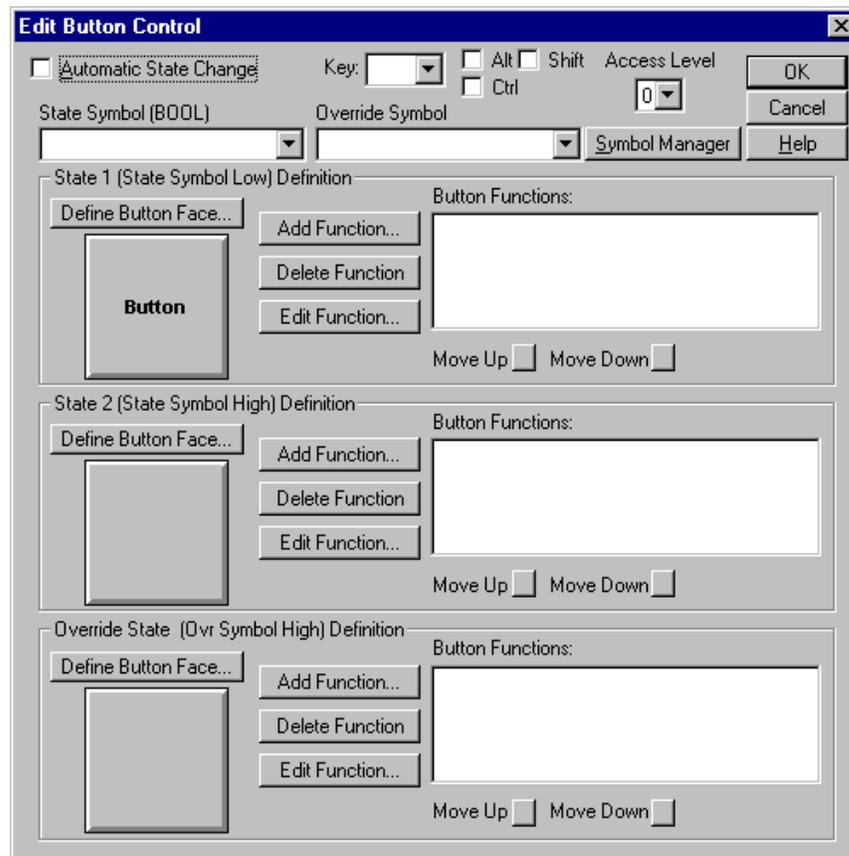
Campo	Descrizione
Border	Questo definisce l'aspetto del bordo. Digitare un valore numerico per la sua larghezza e selezionarne il colore nella lista visualizzata.
Interior	Se è selezionata la casella Transparent, l'interno del box è trasparente. Altrimenti, è possibile selezionare un colore di riempimento del box.

Click Button

Questo controllo esegue una funzione quando viene cliccato. Le funzioni includono il settaggio o l'azzeramento di un simbolo booleano; la visualizzazione di uno schermo; la selezione, l'esecuzione, la sospensione o l'interruzione di un programma; l'ottenimento di dati dall'operatore; il richiamo di un file eseguibile. Il pulsante può essere configurato a uno o a due stati, con stato sostitutivo. Possono essere programmate più funzioni per ciascuno stato e il pulsante eseguirà le funzioni in base al suo stato corrente. Il pulsante può essere configurato per il cambio di stato automatico o in base a un simbolo booleano. E' possibile assegnare al pulsante un livello di accesso e un tasto (o una combinazione di tasti).



La finestra di dialogo *Edit Click Button* è visibile nella sottostante figura ed è descritta nella tabella che segue.



Campo	Descrizione
Automatic State Change	<p>Se questa casella è selezionata, lo stato del pulsante cambia automaticamente ogni volta che esso viene cliccato. Il pulsante alternerà automaticamente lo stato 1 e lo stato 2. In questo caso, il campo State Symbol è disabilitato.</p> <p>Se questa casella non è selezionata e non è definito lo State Symbol, il pulsante è monostato e rimane sempre nello stato 1.</p>
State Symbol	<p>Definisce il simbolo booleano che determina lo stato del pulsante. Se è definito, il pulsante passa allo stato 1 quando il simbolo è basso e passa allo stato 2 quando il simbolo è alto.</p> <p>Digitare il nome del simbolo o selezionarlo nella lista visualizzata. Se è necessario è possibile definire un nuovo simbolo aprendo il Symbol Manager con il pulsante Symbol Manager.</p> <p>Se la casella Automatic State Change non è selezionata e non è definito lo State Symbol, il pulsante è monostato e rimane sempre nello stato 1.</p>
Override Symbol	<p>Definisce il simbolo booleano che seleziona lo stato sostitutivo del pulsante. Se è definito, il pulsante assume lo stato sostitutivo quando questo simbolo è alto.</p> <p>Digitare il nome del simbolo o selezionarlo nella lista visualizzata. Se è necessario è possibile definire un nuovo simbolo aprendo il Symbol Manager con il pulsante Symbol Manager.</p>
Key	<p>Specifica un tasto di scelta rapida per il pulsante. E' possibile selezionare un tasto funzione nella lista visualizzata, digitare un carattere negli intervalli A .. Z o 1 .. 9. E' possibile specificare la combinazione dei tasti Alt, Shift e Ctrl con il tasto selezionato.</p>
Access Level	<p>Specifica un livello di accesso per il pulsante. Se è definito un livello di accesso, per cliccare sul pulsante è necessario che il livello di accesso dell'interfaccia operatore sia posto al livello di accesso specificato o ad un livello più alto.</p>
Button Functions	<p>Visualizza l'elenco delle funzioni di controllo per ciascuno stato del pulsante. Quando il pulsante viene premuto e rilasciato, sono eseguite le funzioni di controllo assegnate allo stato corrente del pulsante.</p>
Add Function	<p>Aggiunge una funzione di controllo alla lista Button Functions. Fare riferimento alla tabella Funzioni dei pulsanti per l'elenco e la descrizione delle funzioni che possono essere associate ai pulsanti.</p> <p>Se la funzione aggiunta è una funzione RUN, STOP, o ABORT, ed è specificato il nome di un programma, sotto alla lista delle funzioni compare il pulsante Edit Program. Se viene premuto questo pulsante, si attiva il Program Editor e viene aperto il file specificato. Se il file non esiste, viene aperto un nuovo file..</p>
Delete Function	<p>Cancela dalla lista Button Functions la funzione evidenziata.</p>
Edit Function	<p>Edita i parametri della funzione evidenziata nella lista Button Functions.</p>
Move Up	<p>Sposta di una posizione verso l'alto la funzione evidenziata nella lista Button Functions.</p>

Campo	Descrizione
Move Down	Sposta di una posizione verso il basso la funzione evidenziata nella lista Button Functions.
Define Button Face	Aprire la finestra di dialogo Define Button Face.
Line 1 Title Line 2 Title	Al pulsante possono essere date due righe di testo per descriverne la funzione.
Bevel	Questa proprietà influenza l'aspetto del bordo del controllo. I valori possibili di Bevel vanno da 0 a 6. Se si specifica 0, il controllo è piatto. L'aumento del valore di Bevel dà al controllo un aspetto tridimensionale.
Surface Color	Specifica il colore di fondo del controllo.
Select Icon Remove Icon	Questi pulsanti permettono di assegnare al pulsante un'icona, o di cancellarla.
Select Font	Questo pulsante visualizza la finestra di dialogo <i>Font</i> , che permette di specificare il tipo, lo stile, la dimensione e il colore del testo del controllo.

Funzioni dei pulsanti

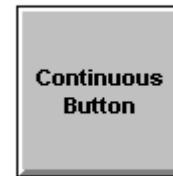
La tabella che segue elenca le funzioni di controllo che possono essere assegnate ai pulsanti. Una funzione assegnata ad un pulsante viene eseguita quando il pulsante viene premuto e lo schermo è attivo.

Funzione	Descrizione
SELECT PROGRAM (Selezione programma)	Quando lo schermo è attivo, visualizza un elenco di programmi SFC del progetto; l'operatore può selezionare il programma desiderato su questo elenco. Le funzioni Run Program, Stop Program e Abort Program possono essere predisposte per agire sul programma selezionato in questo modo (SELECTED PROGRAM).
RUN... (Esecuzione programma)	Esegue il programma specificato o quello selezionato dall'operatore con la funzione Select Program (SELECTED PROGRAM).
STOP... (Arresto programma)	Ferma il programma SFC specificato o quello selezionato dall'operatore con la funzione Select Program (SELECTED PROGRAM). L'esecuzione del programma può essere riavviata dal punto di arresto con la funzione Run Program. E' possibile fermare solo i programmi SFC.
ABORT... (Interruzione programma)	Interrompe l'esecuzione del programma specificato o di quello selezionato dall'operatore con la funzione Select Program (SELECTED PROGRAM).
SCREEN... (Attivazione schermo operatore)	Attiva uno specifico schermo dell'interfaccia con l'operatore.
SET... (Impostazione simbolo)	Imposta il simbolo specificato o il simbolo di sistema ESTOP.
CLEAR... (Azzeramento simbolo)	Azzerare il simbolo specificato o resetta il simbolo di sistema ESTOP.
OP INPUT... (Introduzione dato)	In fase di progetto chiede un simbolo e un messaggio. durante il runtime visualizza un campo per l'introduzione di un valore e il messaggio specificato. Il valore introdotto dall'operatore è memorizzato nel simbolo specificato.

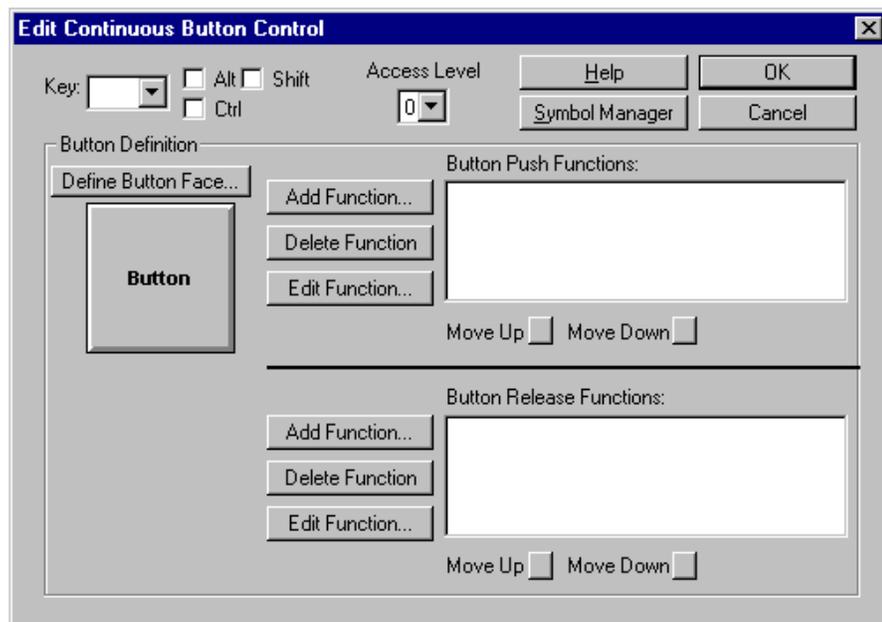
Funzione	Descrizione
EXECUTE... (Esecuzione file)	Esegue il file *.exe specificato. L'esecuzione di una delle utilità del prodotto richiede il livello di accesso appropriatamente autorizzato. Le utilità del prodotto includono il Continuous Logic Manager e il Program Editor. Nota: E' possibile specificare un percorso. Se il percorso contiene degli spazi, l'intera stringa deve essere scritta entro virgolette doppie; ad esempio: "C:\my .program folder\program1.exe"

Continuous Button

Questo controllo esegue una funzione quando viene cliccato. Le funzioni includono il settaggio o l'azzeramento di un simbolo booleano; la visualizzazione di uno schermo; la selezione, l'esecuzione, la sospensione o l'interruzione di un programma; l'ottenimento di dati dall'operatore; il richiamo di un file eseguibile. Le funzioni possono essere programmate in modo da essere eseguite quando il pulsante viene premuto e quando il pulsante viene rilasciato. E' possibile assegnare al pulsante un livello di accesso e un tasto (o una combinazione di tasti).



La finestra di dialogo *Edit Continuous Button* è visibile nella sottostante figura ed è descritta nella tabella che segue.

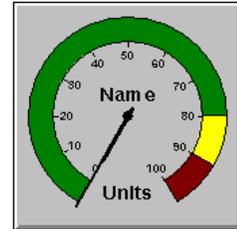


Campo	Descrizione
Key	Specifica un tasto di scelta rapida per il pulsante. E' possibile selezionare un tasto funzione nella lista visualizzata o digitare un carattere negli intervalli A .. Z o 1 .. 9. E' possibile combinare i tasti Alt, Shift e Ctrl con il tasto selezionato.

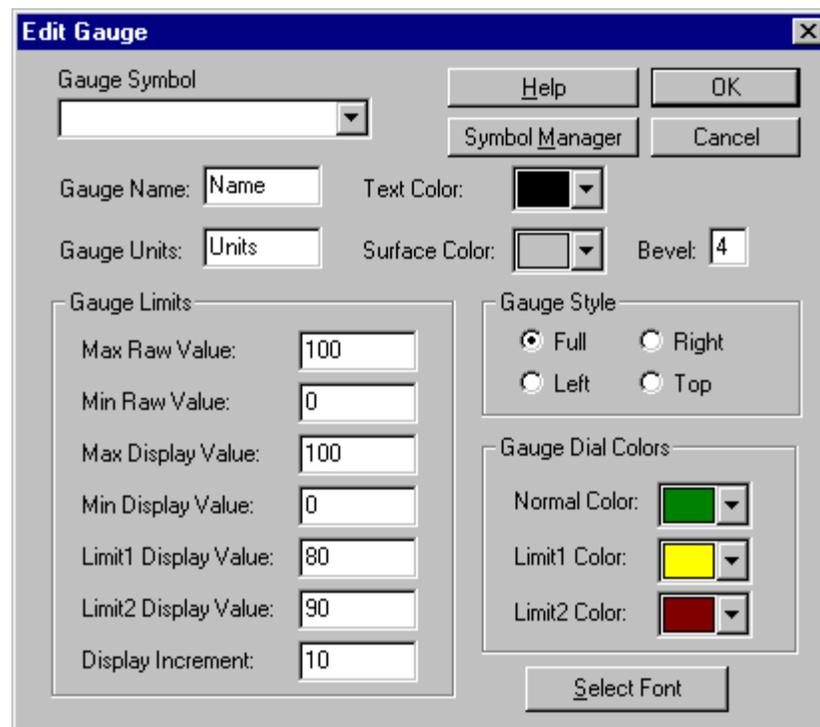
Campo	Descrizione
Access Level	Specifica un livello di accesso per il pulsante. Se è definito un livello di accesso, per cliccare sul pulsante è necessario che il livello di accesso dell'interfaccia operatore sia posto al livello di accesso specificato o ad un livello più alto.
Button Push Function	Visualizza la lista delle funzioni eseguite quando il pulsante viene premuto.
Button Release Function	Visualizza la lista delle funzioni eseguite quando il pulsante viene rilasciato.
Add Function	<p>Aggiunge una funzione di controllo alla lista Button Push Functions o alla lista Button Release Functions. Fare riferimento alla tabella Funzioni dei pulsanti per l'elenco e la descrizione delle funzioni che possono essere associate ai pulsanti.</p> <p>Se la funzione aggiunta è una funzione RUN, STOP, o ABORT, ed è specificato il nome di un programma, sotto alla lista delle funzioni compare il pulsante Edit Program. Se viene premuto questo pulsante, si attiva il Program Editor e viene aperto il file specificato. Se il file non esiste, viene aperto un nuovo file.</p>
Delete Function	Cancella dalla lista Button Push Functions o dalla lista Button Release Functions la funzione evidenziata.
Edit Function	Edita i parametri della funzione evidenziata nella lista Button Push Functions o nella lista Button Release Functions.
Move Up	Sposta di una posizione verso l'alto la funzione evidenziata nella lista Button Push Functions o nella lista Button Release Functions.
Move Down	Sposta di una posizione verso il basso la funzione evidenziata nella lista Button Push Functions o nella lista Button Release Functions.
Define Button Face	Apri la finestra di dialogo Define Button Face.
Line 1 Title Line 2 Title	Al pulsante possono essere date due righe di testo per descriverne la funzione.
Bevel	Questa proprietà influenza l'aspetto del bordo del controllo. I valori possibili di Bevel vanno da 0 a 6. Se si specifica 0, il controllo è piatto. L'aumento del valore di Bevel dà al controllo un aspetto tridimensionale.
Surface Color	Specifica il colore di fondo del controllo.
Select Icon Remove Icon	Questi pulsanti permettono di assegnare al pulsante un'icona, o di cancellarla.
Select Font	Questo pulsante visualizza la finestra di dialogo <i>Font</i> , che permette di specificare il tipo, lo stile, la dimensione e il colore del testo del controllo

Gauge

Questo controllo visualizza una scala e un indicatore rotante. L'indicatore segue il valore del simbolo assegnato al controllo. Al variare del valore del simbolo, l'indicatore ruota in senso orario o antiorario per indicarne il valore corrente. Possono essere stabiliti dei limiti intermedi, raggiunti i quali cambia il colore della scala. Sono disponibili quattro stili diversi.



La finestra di dialogo *Edit Gauge* è visibile nella sottostante figura ed è descritta nella tabella che segue.



Campo	Descrizione
Gauge Symbol	Specifica il simbolo il cui valore è seguito dall'indicatore rotante. Se necessario, è possibile definire un nuovo simbolo aprendo il Symbol Manager con il pulsante Symbol Manager.
Gauge Name	Specifica il nome del controllo. Serve solo per la documentazione dello schermo.
Gauge Units	Specifica le unità del controllo. Serve solo per la documentazione dello schermo.
Text Color	Specifica il colore che deve essere usato per il nome, le unità e la numerazione della scala del controllo.
Surface Color	Specifica il colore dello sfondo del controllo.

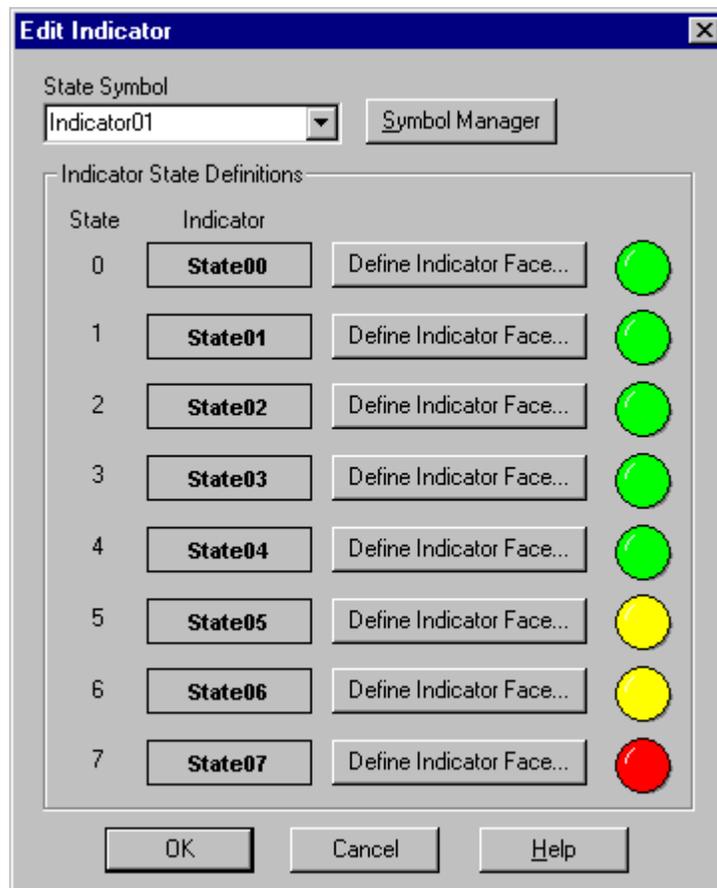
Campo	Descrizione
Bevel	Questa proprietà influenza l'aspetto del bordo del controllo. I valori possibili di Bevel vanno da 0 a 6. Se si specifica 0, il controllo è piatto. L'aumento del valore di Bevel dà al controllo un aspetto tridimensionale.
Gauge Limits Max Raw Value Min Raw Value Max Display Value Min Display Value Limit1 Display Value Limit2 Display Value Display Increment	Limiti dell'indicatore rotante. Specificano i valori massimo e minimo ai quali risponde l'indicatore rotante. Essi limitano il campo dei valori rappresentabili dall'indicatore a colonna. Se il valore del simbolo associato al controllo supera il limite superiore, l'indicatore rotante va a fondo scala, mentre qualsiasi valore al di sotto del limite inferiore lo posiziona a inizio scala. Normalmente l'indicatore ruota in senso orario all'aumentare del valore del simbolo. Per invertire il senso di rotazione dell'indicatore, scambiare i valori dei due limiti. Specifica i valori massimo e minimo della scala. Specifica i limiti ai quali cambia il colore della scala; ad esempio, per indicare una condizione di allarme. Specifica l'incremento della numerazione della scala.
Gauge Style	Cerchio intero. Semicerchio con il lato arrotondato in alto. Semicerchio con il lato arrotondato a sinistra. Semicerchio con il lato arrotondato a destra
Gauge Dial Colors	Normale – il colore della scala prima che venga raggiunto uno dei limiti intermedi. Limit 1 – il colore della scala a partire dal limite 1. Limit 2 - il colore della scala a partire dal limite 2.
Symbol Manager	Cliccare su questo pulsante per aprire il Symbol Manager.
Select Font	Questo pulsante visualizza la finestra di dialogo <i>Font</i> , che permette di specificare il tipo, lo stile, la dimensione e il colore del testo del controllo (nome, unità, e numerazione della scala).

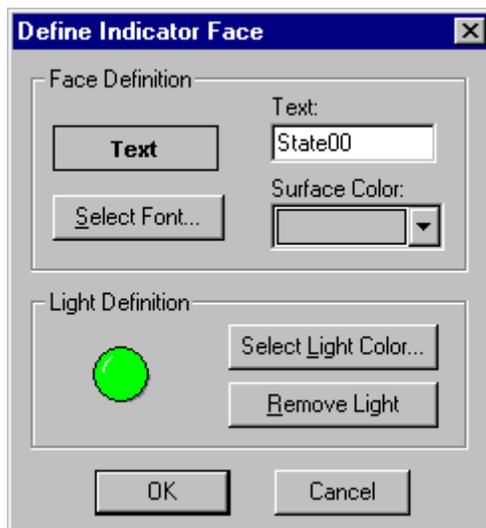
Indicator

Questo è un indicatore di stato il cui aspetto (testo, colore, ecc.) cambia in base al valore di un simbolo di tipo BOOL o di tipo BYTE. Un simbolo BOOL può selezionare fra due stati e un simbolo BYTE può selezionare fra un massimo di otto stati.



Le finestre di dialogo *Edit Indicator* e *Define Indicator Face* sono visibili nelle sottostanti figure e sono descritte nella tabella che segue.





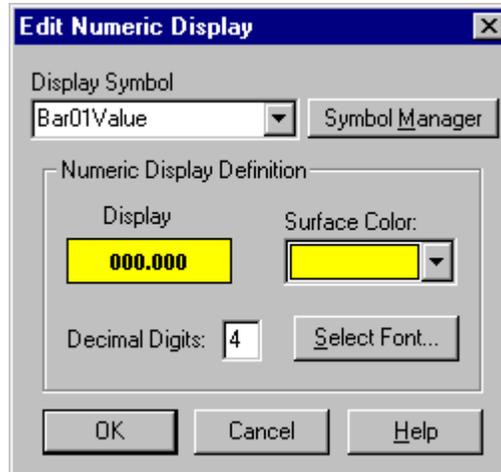
Campo	Descrizione
State Symbol	Specifica il simbolo al quale risponde l'indicatore. Questo simbolo può essere di tipo BOOL o di tipo BYTE. Se il simbolo è di tipo BOOL, l'indicatore può avere due stati. Se il simbolo è di tipo BYTE, l'indicatore può avere fino a otto stati. Se è necessario, è possibile definire un nuovo simbolo aprendo il Symbol Manager con il pulsante Symbol Manager.
Symbol Manager	Cliccare su questo pulsante per aprire il Symbol Manager.
Indicator State Definitions	Sono indicati il testo, il colore di fondo e la spia opzionale dell'indicatore corrispondenti a ciascuno dei possibili stati binari (da 0 a 7).
Define Indicator Face	Cliccare su questo pulsante per visualizzare la finestra di dialogo <i>Define Indicator Face</i> . Questa finestra serve per definire il testo, il colore del testo e il colore di fondo per un certo stato dell'indicatore e per definire il tipo e la dimensione dei caratteri per l'intero indicatore. Per ciascuno stato dell'indicatore può anche essere definita una spia colorata opzionale.
Face Definition	Definizione dell'aspetto dell'indicatore.
Text	La stringa da visualizzare quando l'indicatore è in questo stato.
Selezionare Font	Definisce il tipo e la dimensione del testo dell'indicatore e il colore del testo per questo stato dell'indicatore.
Surface Color	Definisce il colore di fondo per questo stato dell'indicatore..
Light Definition	Definizione della spia.
Select Light	Visualizza un gruppo di spie colorate selezionabili per questo stato dell'indicatore.
Remove Light	Rimuove la spia definita per questo stato dell'indicatore.

Numeric Display

Questo controllo si usa per visualizzare il valore di un simbolo (numerico o di tipo STRING)..

79.6460

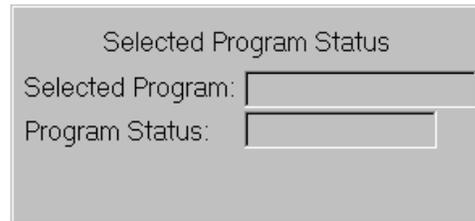
La finestra di dialogo *Edit Numeric Display* è visibile nella sottostante figura ed è descritta nella tabella che segue.



Campo	Descrizione
Display Symbol	Specifica il simbolo il cui valore è visualizzato dal controllo. Selezionare digitare il nome del simbolo o selezionarlo nella lista a discesa. Se è necessario, è possibile definire un nuovo simbolo aprendo il Symbol Manager con il pulsante Symbol Manager.
Symbol Manager	Cliccare su questo pulsante per aprire il Symbol Manager.
Display	Questo campo indica il colore del testo, il colore di fondo e il tipo di carattere selezionati.
Surface Color	Definisce il colore di fondo del controllo.
Decimal Digits	Definisce il numero delle cifre visualizzate a destra del punto decimale.
Select Font	Definisce il tipo, il colore e la dimensione dei caratteri.

Pannello Selected Program Status

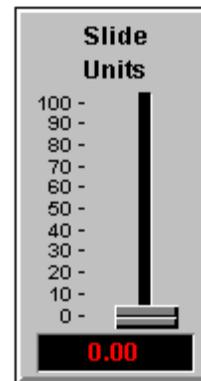
Questo pannello visualizza il nome e lo stato corrente (ad esempio: in esecuzione, sospeso, ecc.) del programma selezionato. Questo programma è quello selezionato da un comando SELECT PROGRAM associato a un pulsante (Click o Continuous Button).



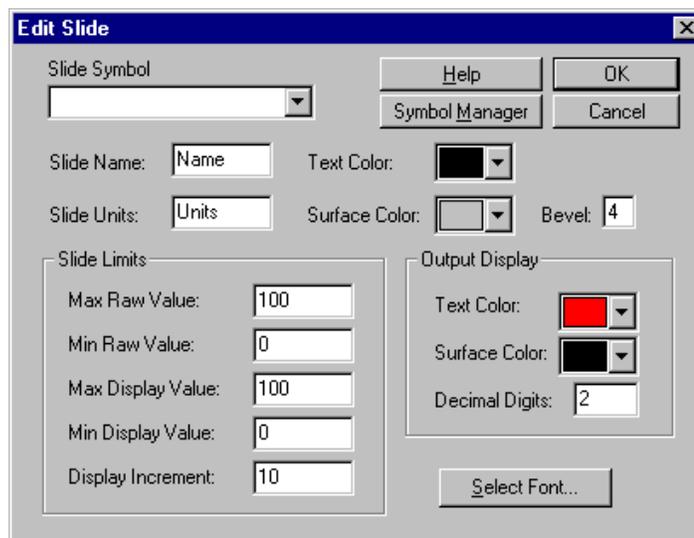
Questo pannello non ha opzioni di configurazione.

Slide

Questo controllo dà la possibilità di variare in modo continuo il valore di un simbolo. La variazione del valore del simbolo assegnato al controllo si attua spostando l'indicatore di quest'ultimo. Sotto al controllo è visualizzato il valore corrente del simbolo ad esso associato. Una approssimazione visiva è data dalla scala visualizzata a fianco del percorso dell'indicatore.



La finestra di dialogo *Edit Slide* è visibile nella sottostante figura ed è descritta nella tabella che segue..

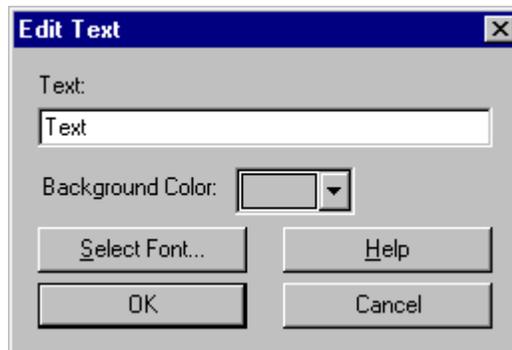


Campo	Descrizione
Slide Symbol	Specifica il simbolo associato al controllo.
Slide Name	Specifica il nome del controllo. Serve solo per la documentazione dello schermo.
Slide Units	Specifica le unità del controllo. Serve solo per la documentazione dello schermo.
Text Color	Specifica il colore che deve essere usato per il nome, le unità e la numerazione della scala del controllo.
Surface Color	Specifica il colore dello sfondo del controllo.
Bevel	Questa proprietà influenza l'aspetto del bordo del controllo. I valori possibili di Bevel vanno da 0 a 6. Se si specifica 0, il controllo è piatto. L'aumento del valore di Bevel dà al controllo un aspetto tridimensionale.
Slide Limits	I campi Max e Min raw value determinano l'interazione fra il controllo e il suo simbolo. I campi Max e Min Display Value determinano l'interazione fra il controllo e la sua scala.
Max Raw Value	Specificano i valori massimo e minimo che il controllo può assegnare al simbolo. Il controllo non assegnerà al simbolo valori al di fuori di questi limiti
Min Raw Value	
Max Display Value	Specificano i valori massimo e minimo della numerazione della scala.
Min Display Value	
Display Increment	Specifica l'incremento della numerazione della scala.
Output Display	Questo è il valore visualizzato dal controllo (non il valore del simbolo). Cioè, indica la posizione dell'indicatore scorrevole rispetto alla scala.
Text Color	Definisce il colore del testo visualizzato in Output Display.
Surface Color	Definisce il colore di fondo di Output Display.
Decimal Digits	Definisce il numero delle cifre visualizzate dopo il punto decimale.
Symbol Manager	Cliccare su questo pulsante per visualizzare il Symbol Manager.
Selezionare Font	Questo pulsante visualizza la finestra di dialogo <i>Font</i> , che permette di specificare il tipo, lo stile, la dimensione e il colore del testo del controllo (nome, unità, e numerazione della scala).

Text

Questo elemento si usa per etichettare controlli o gruppi di controlli, documentare le funzioni dello schermo o altrimenti descrivere lo schermo e le sue funzioni. **Text**

La finestra di dialogo *Edit Text* è visibile nella sottostante figura ed è descritta nella tabella che segue..



Campo	Descrizione
Text	Digitare la stringa da visualizzare nella finestra di testo.
Background Color	Definisce il colore di fondo dell'elemento Text.
Select Font	Definisce il colore, il tipo e la dimensione del testo visualizzato.

Controlli ActiveX

Introduzione

I controlli ActiveX sono DLL con una interfaccia OLE2; l'estensione file per difetto è OCX, anziché DLL. I controlli ActiveX includono controlli semplici, come pulsanti e finestre di edit, controlli più sofisticati, come gli indicatori, e componenti DLL senza interfaccia utente. I controlli ActiveX richiedono un contenitore, che è fornito dall'interfaccia operatore.

I controlli ActiveX hanno tre tipi di funzioni:

- **Proprietà** – queste funzioni interne al controllo possono essere richiamate dal contenitore per configurare l'aspetto del controllo stesso (colori, tipi di carattere, testi, bitmap, ecc.).
- **Metodi** – Queste funzioni interne al controllo permettono al contenitore di inviare o ricevere informazioni relative al controllo. Ad esempio, un metodo può essere usato per ottenere il valore di una cella in un controllo di tipo griglia o per ordinare ad un controllo di eseguire il repaint di se stesso.
- **Eventi** – queste funzioni sono richiamate quando all'interno del controllo si verifica un evento di Windows, ad esempio, un click del mouse, o la pressione di un tasto. Il contenitore viene informato del verificarsi di questi eventi.

Le proprietà, i metodi e gli eventi sono configurati tramite l'Operator Interface Screen Editor dopo che il controllo è stato aggiunto a uno schermo.

Limitazioni dei controlli ActiveX

Quando si usano i controlli ActiveX nell'interfaccia operatore, valgono le seguenti limitazioni:

1. Non sono supportati i controlli composti (controlli che contengono altri controlli).
2. I tipi non sono forzati (ad esempio da DWORD a INT).
3. Non sono supportati i controlli privi di una propria pagina di proprietà.
4. Non è supportato il cambio del tipo di carattere durante il runtime. Il software Operator Interface non ha un tipo "font".
5. Non sono supportati parametri di tipo VT_VARIANT per metodi ed eventi.
6. Non sono supportati tipi di dati complessi (strutture) per i parametri di metodi ed eventi .

Controlli ActiveX e controlli standard

I controlli standard forniti insieme al prodotto forniscono le funzioni fondamentali per l'interfaccia operatore di un sistema a controllore programmabile. Questi controlli hanno una conoscenza interna dei sistemi a controllore programmabile. Questa conoscenza permette loro di nascondere all'utente finale molti dettagli complessi.

Per estendere la funzionalità o per soddisfare particolari requisiti del sistema, l'interfaccia operatore può servire come contenitore di controlli ActiveX. Qualsiasi controllo ActiveX disponibile può essere incluso in uno schermo dell'interfaccia con l'operatore e comunicare con il motore di runtime (purché i tipi di dati siano compatibili con il software del sistema di comando).

Note:

- Gli utenti esperti possono scrivere i propri controlli ActiveX, oppure possono ottenerli da terze parti. E' loro responsabilità verificare l'adeguatezza all'applicazione di un controllo ActiveX di terza parte e di ottenerne la documentazione. Un controllo non ben progettato può causare problemi. Contattare chi ha implementato il controllo se il suo impiego genera problemi.
- I controlli ActiveX di terze parti **non** avranno alcuna conoscenza interna dei sistemi a controllore programmabile. Esempi di conoscenze che questi controlli **non** hanno possono essere: come comandare un arresto in emergenza (ESTOP) e come avviare, sospendere e interrompere i programmi di un sistema a controllore programmabile.
- I controlli ActiveX richiedono un processo di registrazione che permette loro di essere visti da Windows NT o Windows 95. La registrazione del controllo può essere stata predisposta da chi ha implementato il controllo, ma può anche non esserlo stata. Per tener conto di questo, l'Operator Interface Screen Editor implementa un processo di registrazione.

Fornitori di controlli ActiveX

Sono stati testati due package di controlli ActiveX di terze parti, che sono fonti raccomandate di controlli ActiveX:

VICOMponents™ Version 4.0 by ComputerBoards, Inc.
ComputerBoards Inc.
125 High Street
Mansfield, MA 02048
www.computerboards.com

Global Majic Software ActiveX controls.
Global Majic Software, Inc.
P.O. Box 322
Madison, Alabama 35758
gms@globalmajic.com
www.globalmajic.com

Registrazione di controlli ActiveX

Prima di poter essere usato in uno schermo dell'interfaccia con l'operatore, un controllo ActiveX deve essere registrato. Se il controllo ActiveX non è stato precedentemente registrato, può esserlo attraverso l'Operator Interface Screen Editor. I controlli registrati compaiono nella lista visualizzata quando nel menu *Edit* viene selezionato *Insert ActiveX Control*.

Per registrare un controllo ActiveX

1. Scegliere *Register ActiveX Control* nel menu *Edit*. Compare la finestra di dialogo *OCX Files to Register*.
2. Selezionare il controllo da registrare e cliccare su *Open* per registrarlo. Se la registrazione del controllo ha successo, comparirà un messaggio di conferma.

Nota

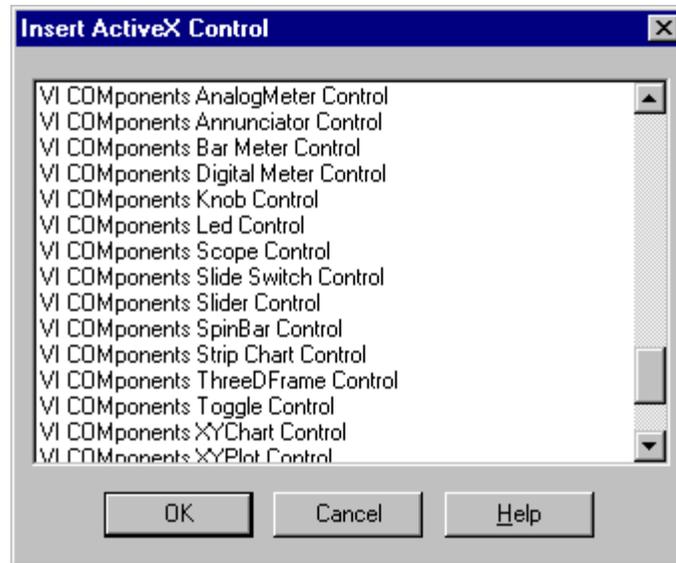
Questo processo non garantisce che il controllo potrà essere correttamente inserito in uno schermo. Ad esempio, se il controllo è privo di licenza o non è stato implementato correttamente, non si sarà in grado di inserirlo in uno schermo.



Inserimento di controlli ActiveX

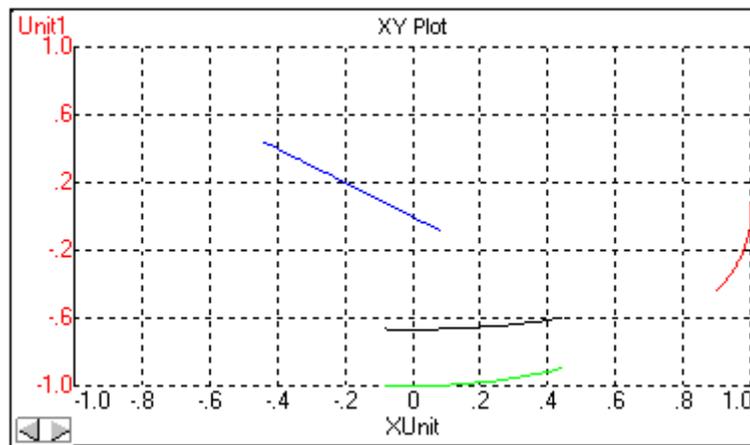
Per inserire un controllo ActiveX

1. Scegliere *Insert ActiveX control* nel menu *Edit*. Comparirà la finestra di dialogo *Insert ActiveX Control*.
2. Selezionare un controllo ActiveX e cliccare su *OK* per continuare.



In alternativa, il controllo può essere selezionato nella lista degli ultimi quattro controlli ActiveX inseriti, visualizzata in fondo al menu *Edit*.

Un esempio del componente VI XY Plot control è illustrato nella sottostante figura.



Nota

La lista contiene tutti i controlli ActiveX registrati sul PC. Alcuni di essi possono non essere stati concepiti per un ambiente di comando e controllo e può quindi non avere senso utilizzarli in tale ambiente.

Se viene distribuita un'applicazione che nell'interfaccia operatore usa un controllo ActiveX, il controllo deve essere registrato sul sistema di destinazione.

Fare riferimento a chi ha implementato il controllo per i diritti e gli accordi relativi all'uso e alla distribuzione del controllo stesso.

Editazione di controlli ActiveX

I controlli ActiveX possono essere spostati e ridimensionati come i controlli standard. Il risultato del ridimensionamento di un controllo dipende dal controllo stesso e non è necessariamente quello che ci si potrebbe attendere. Le altre operazioni, come cancella, taglia, copia e incolla funzionano come per i controlli standard.

Editazione delle proprietà

Per editare le proprietà, usare uno dei seguenti metodi:

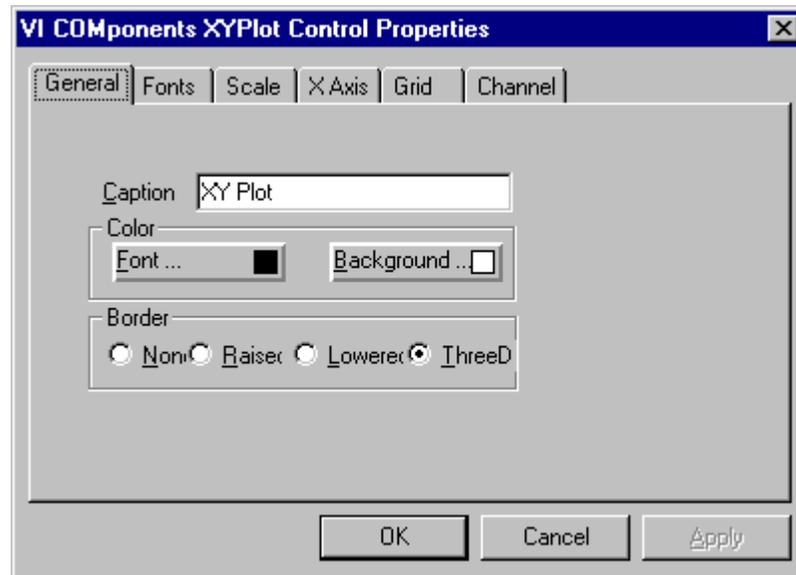
- Selezionare il controllo ActiveX e scegliere *Edit ActiveX Properties* nel menu *Edit* o nel menu contestuale, oppure usare la barra degli strumenti.
- Fare doppio click sul controllo ActiveX e scegliere *Edit Property Pages* nella finestra *Select ActiveX Editing Mode* visualizzata di conseguenza.

I dialoghi della pagina delle proprietà vengono dall'interno del controllo stesso. Questo significa che ogni controllo ha una diversa finestra di dialogo per l'editazione delle proprietà. Per i controlli che non hanno questa finestra di dialogo, compare un messaggio che indica questa condizione.

Nota

Per ottenere informazioni sulle proprietà di un controllo ActiveX rivolgersi al fornitore del controllo. Queste informazioni non sono contenute nell'Operator Interface Screen Editor. Tuttavia, se il controllo è accompagnato da un file di aiuto, a questo si potrà accedere dalla finestra di dialogo del controllo stesso.

La sottostante figura presenta una tipica finestra di dialogo per la pagina delle proprietà di un controllo ActiveX.



Editazione dei metodi

La configurazione dei metodi di un controllo ActiveX richiede in genere tre passi:

- Selezionare i metodi del controllo che saranno richiamati dall'interfaccia operatore.
- Assegnare i simboli necessari agli argomenti e al valore di ritorno del metodo (se necessario).
- Definire quando il metodo sarà richiamato.

I controlli hanno un numero variabile di metodi. Un metodo richiama una funzione interna del controllo. Gli argomenti e il valore di ritorno di una funzione possono essere di uno dei molti tipi di dati possibili (attualmente 38). Ciascuno di questi tipi di dati può o non può essere mappato su uno dei tipi di dati supportati dal sistema. Se non è possibile mappare un tipo di dati supportato sul tipo di dati usato dal metodo, sarà impossibile richiamare questo metodo. Se si tenta di configurarlo, compare il messaggio di errore appropriato.

I tipi degli argomenti dei Controlli ActiveX iniziano con *VT_*. La tabella che segue mostra come questi tipi sono mappati sui tipi di dati supportati.

Tipi di dati dei controlli ActiveX	Tipi di dati supportati dal sistema
VT_BOOL	BOOL
VT_UI1, VT_I1	BYTE
VT_UI2	WORD
VT_UI4, VT_UINT	DWORD
VT_I2, VT_INT	INT
VT_I4	DINT
VT_R4, VT_R8	REAL
VT_BSTR	STRING
VT_VOID, VT_EMPTY, VT_NULL	Non mappati
VT_PTR	Saranno supportati puntatori a tutti i precedenti tipi <i>VT_</i>

Un aspetto della configurazione di un controllo ActiveX consiste nel definire quando devono essere richiamati i metodi del controllo. Questo si fa sulla base del tempo con un temporizzatore da 200 millisecondi, che gira nell'interfaccia operatore. I metodi sono suscettibili di richiamo ogni 200 millisecondi o multiplo di 200 millisecondi. Il multiplo sarà definito dall'utente fino al valore di 100 (=20 secondi).

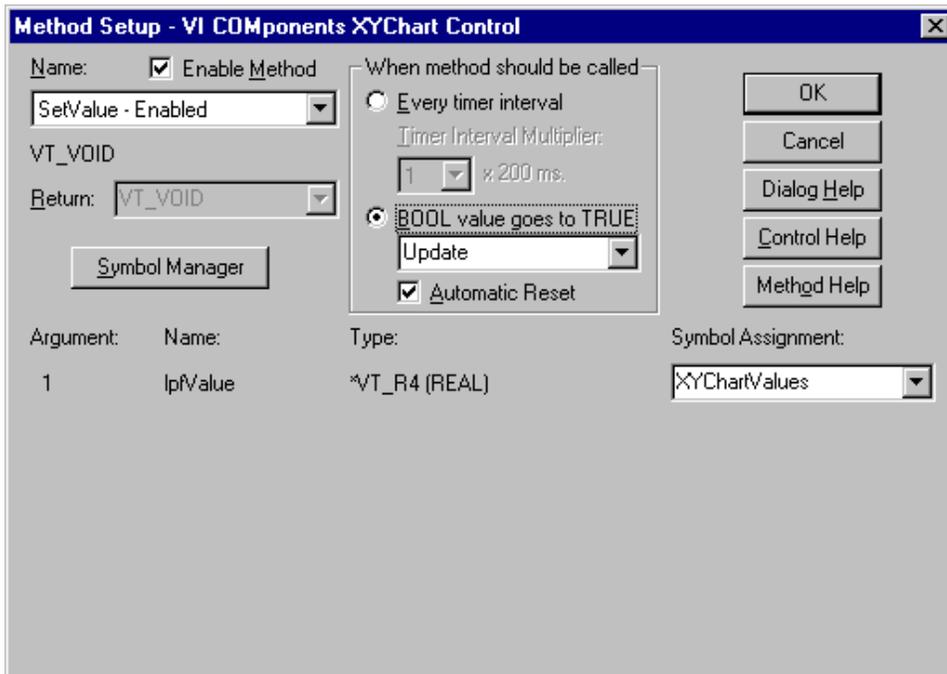
Un metodo può essere richiamato in due modi:

Ad ogni intervallo del temporizzatore – o ad ogni multiplo definito dall'utente dell'intervallo del temporizzatore.

Quando un simbolo di tipo BOOL diventa TRUE – quando un simbolo booleano definito dall'utente diventa TRUE, il metodo è richiamato entro il successivo intervallo del temporizzatore. Dopo il ritorno dal metodo, il simbolo booleano viene opzionalmente messo a FALSE.

Per editare i metodi

- Operare in uno dei seguenti modi:
 - Selezionare il controllo ActiveX e scegliere *Edit ActiveX Methods* nel menu *Edit* o nel menu contestuale, oppure usare la barra degli strumenti.
 - Fare doppio click sul controllo ActiveX e scegliere *Edit Methods* nella finestra di dialogo *Select ActiveX Editing Mode*, che compare di conseguenza.
- Compare la finestra di dialogo *Method Setup*. Configurare i metodi in base alle proprie necessità, facendo riferimento alla seguente tabella. Alla fine, cliccare su *OK* per salvare la configurazione e ritornare all'Operator Interface Screen Editor.



Campo	Descrizione
Name	Selezionare il metodo da configurare nella lista a discesa. I nomi dei metodi compaiono nella lista in ordine alfabetico.
Enable Method	Selezionare questa casella per abilitare il metodo. Per poter essere richiamato, un metodo deve essere stato abilitato.
Return	Se il tipo restituito dal metodo è VOID, nel campo Return compare VT_VOID, che non può essere cambiato; altrimenti viene indicato il tipo restituito e nel campo Return deve essere specificato un simbolo di tipo appropriato. Se necessario, cliccare su <i>Symbol Manager</i> per creare un nuovo simbolo.
Arguments	Se il metodo accetta argomenti, questi sono elencati con il loro numero, nome e tipo di dato. I metodi possono avere al massimo 16 argomenti. Se un metodo ha più di 16 argomenti, compare un messaggio di errore. Per ciascun argomento al quale deve essere assegnato un valore, selezionare un simbolo nella lista a discesa del corrispondente campo Symbol Assignment. Se necessario, cliccare su <i>Symbol Manager</i> per creare nuovi simboli. Al metodo possono essere passati puntatori ad array (definiti nel <i>Symbol Manager</i>). Specificare il nome dell'array senza indice per passare l'intero array. La dimensione dell'array deve essere quella richiesta dal metodo.

Campo	Descrizione
When method should be called	Se il metodo deve essere richiamato ad intervalli di tempo, cliccare su <i>Every Timer Interval</i> e selezionare un moltiplicatore. Se il metodo deve essere richiamato quando un simbolo booleano diventa TRUE, selezionare un moltiplicatore dell'intervallo di tempo, cliccare su <i>BOOL value goes TRUE</i> e selezionare un simbolo booleano nella lista a discesa. Se è selezionata la casella <i>Automatic Reset</i> , il simbolo booleano diventa FALSE dopo il richiamo del metodo.
Symbol Manager	Visualizza il Symbol Manager.
Dialog Help	Visualizza l'aiuto per la finestra di dialogo Method Setup.
Control Help	Visualizza l'aiuto per il controllo ActiveX (fornito da chi ha implementato il controllo).
Method Help	Visualizza l'aiuto del controllo ActiveX per questo metodo (fornito da chi ha implementato il controllo).

Editazione degli eventi

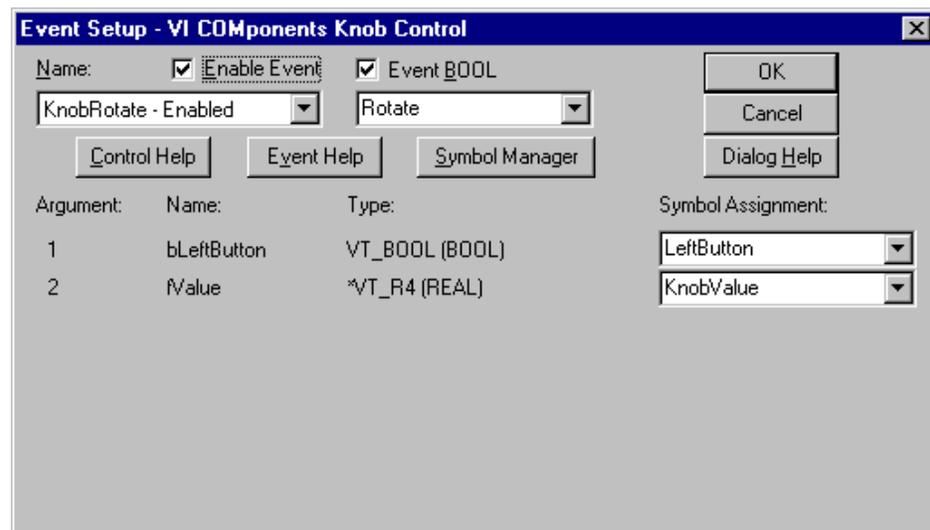
Nella maggior parte dei casi un controllo ActiveX ha una lista di eventi che notifica al suo contenitore, cioè all'interfaccia operatore. Esempi di questi eventi sono un click del mouse e la pressione di un tasto. Agli eventi possono o non possono essere associati dei parametri. Ad esempio, un evento *on_focus* non ha parametri, ma un evento *mouse_click* potrebbe avere due parametri di tipo intero, corrispondenti alle coordinate x e y. Quando si verifica un evento, l'interfaccia operatore può fare due cose distinte:

- Mettere a TRUE un simbolo booleano – questo è facoltativo se l'evento ha almeno un parametro (argomento). Se l'evento non ha parametri, questo è obbligatorio. (Il simbolo **non** viene messo a FALSE dall'interfaccia operatore dopo l'evento.)
- Assegnare a un simbolo il valore di un parametro (argomento) dell'evento – se nella configurazione è stato assegnato un simbolo a un parametro dell'evento, il valore del parametro viene assegnato al simbolo durante l'esecuzione.

Per editare gli eventi

1. Operare in uno dei seguenti modi:
 - Selezionare il controllo ActiveX e scegliere *Edit ActiveX Events* nel menu *Edit* o nel menu contestuale, o usare la barra degli strumenti.
 - Fare doppio click sul controllo ActiveX e scegliere *Edit Events* nella finestra di dialogo *Select ActiveX Editing Mode* visualizzata di conseguenza.
2. Comparirà la finestra di dialogo *Event Setup*. Configurare gli eventi in base alle proprie necessità, facendo riferimento alla seguente tabella. Alla fine, cliccare su *OK* per salvare la configurazione e ritornare all'Operator Interface Screen Editor.

Per i controlli privi della finestra di dialogo *Event Setup* comparirà un messaggio indicante questa situazione.



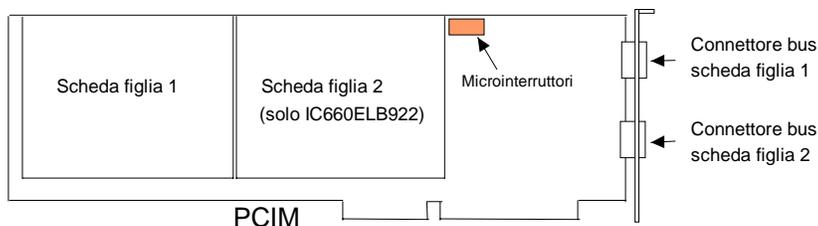
Campo	Descrizione
Name	Selezionare l'evento da configurare nella lista a discesa di questo campo.
Enable Event	Selezionare questa casella per abilitare l'evento. Per poter essere notificato all'interfaccia operatore, un evento deve essere stato abilitato.
Event BOOL	Se è selezionata questa casella ed è specificato un simbolo booleano, durante l'esecuzione l'evento metterà a TRUE quel simbolo. Se necessario, cliccare su <i>Symbol Manager</i> per creare un nuovo simbolo.
Arguments	Se l'evento accetta argomenti, questi sono elencati con il loro numero, nome e tipo di dato. Gli eventi possono avere al massimo 16 argomenti. Se un evento ha più di 16 argomenti, compare un messaggio di errore. Per ciascun argomento al quale deve essere assegnato un valore, selezionare un simbolo nella lista a discesa del corrispondente campo Symbol Assignment. Se necessario, cliccare su <i>Symbol Manager</i> per creare nuovi simboli.
Symbol Manager	Visualizza il Symbol Manager.
Dialog Help	Visualizza l'aiuto per la finestra di dialogo Event Setup.
Control Help	Visualizza l'aiuto per il controllo ActiveX (fornito da chi ha implementato il controllo).
Event Help	Visualizza l'aiuto del controllo ActiveX per questo evento (fornito da chi ha implementato il controllo).

Modulo di interfaccia con Personal Computer per Genius I/O

Questa appendice contiene una descrizione del GE Fanuc Genius® I/O Personal Computer Interface Module (PCIM). Essa contiene anche le procedure di installazione e di configurazione del PCIM.

Descrizione

Il modulo di interfaccia con PC monoslot (PCIM) è un punto di ingresso nel Genius I/O System per PC IBM. Il PCIM è una scheda di tipo "AT", concepita per essere integrata in un sistema a microprocessore sviluppato dall'utente. E' pienamente compatibile con tutti i protocolli Genius, dal punto di vista della meccanica, dei livelli elettrici e della tempificazione delle comunicazioni.



Il PCIM è disponibile con una o con due schede figlia (IC660ELB921/922). Ciascuna scheda figlia del PCIM fornisce una "presa" a basso costo su un bus Genius I/O, permettendo a un sistema host di controllare I/O remoti utilizzando l'estesa diagnostica, l'elevata affidabilità e l'immunità dai disturbi del sistema Genius I/O. Ciascuna scheda figlia è individualmente configurabile mediante il software di configurazione fornito con il PCIM.

Per collegare il PCIM al bus Genius si usano i connettori sul lato della scheda. Se il PCIM ha due schede figlia, esse possono essere collegate allo stesso bus o a bus indipendenti.

Scheda figlia

Una scheda figlia per PCIM è un controllore non specializzato di I/O per il sistema Genius I/O. Essa fornisce un metodo conveniente per il controllo di unità installate sul bus seriale Genius. Questa scheda esegue le operazioni di inizializzazione e di gestione degli errori per un massimo di 30 unità di bus, mantiene immagini aggiornate degli I/O controllati da ciascuna unità (Genius I/O Block o altre unità di bus) e può comunicare con altri controllori sul bus Genius, tramite messaggi in background non associati a comandi I/O o dati globali. L'interfaccia a questa RAM è ottimizzata per il bus del personal computer IBM.

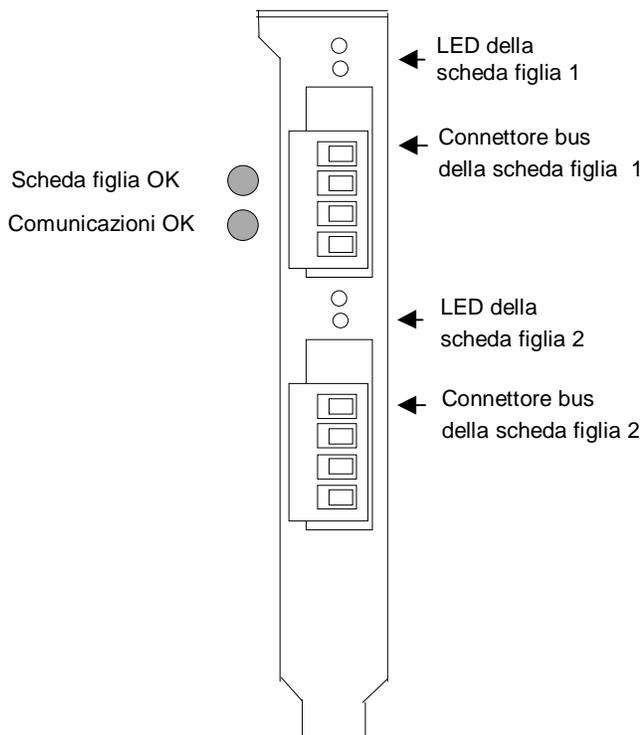
Scheda madre

La scheda madre del PCIM fornisce un conveniente metodo di interfacciamento fra una scheda figlia ad architettura aperta, come la scheda figlia del PCIM, e un sistema host compatibile con IBM. Tutti i segnali di comunicazione fra una scheda figlia e il bus dell'Host sono bufferizzati attraverso la scheda madre. Oltre alle normali linee di interfaccia, la scheda madre fornisce le seguenti funzioni di comando e monitoraggio della scheda figlia:

- “Unit load” standard verso il bus IBM
- Compatibilità con il backplane ISA.
- Rilevazione bassa tensione di alimentazione
- Reset dei segnali all'accensione
- Decodifica degli indirizzi del sistema Host su tutte le mappe della memoria PC AT
- Temporizzatore watchdog incorporato in grado di monitorare il funzionamento del sistema e di disabilitare la scheda figlia in caso di guasto del sistema host, evitando ogni conflitto sul bus Genius.

Frontalino

Sul frontalino del CPIM sono alloggiati due LED per ciascuna scheda figlia (scheda OK e comunicazioni OK), come si vede qui sotto:



Nelle aperture del frontalino trovano posto i connettori del bus seriale per la scheda (o per le schede) figlia del PCIM.

Installazione e Configurazione

Per interfacciare il PCIM con il bus seriale Genius, occorre prima eseguire i seguenti passi:

- Determinare gli indirizzi della porta I/O e della memoria condivisa, e l'IRQ che si utilizzeranno per il PCIM.
- Impostare l'indirizzo I/O della EEPROM mediante i microinterruttori (DIP switches).
- Installare il PCIM nel computer.
- Configurare il PCIM mediante l'utilità PCIM Configuration Utility (PCU).
- Connettere il PCIM al bus seriale.

Requisiti Hardware

I requisiti dell'hardware per l'impiego del PCIM sono gli stessi del software PC Control.

Scelta del calcolatore

Il PCIM è stato testato con successo su molti tipi di calcolatori compatibili con IBM. Esso è pienamente compatibile con il backplane ISA e fornisce la decodifica degli indirizzi del sistema host su tutte le mappe di memoria PC AT. Tuttavia, non è stato possibile testarlo su tutti i calcolatori che possono essere disponibili. Di conseguenza non può essere assicurato il corretto funzionamento del PCIM su qualsiasi tipo di host.

Risorse del calcolatore da riservare

Per installare e configurare il PCIM, occorre riservare specifiche risorse di sistema del calcolatore (porte I/O, memoria e IRQ). Questo viene fatto mediante i microinterruttori della scheda (indirizzo EEPROM) e l'utilità PCIM Configuration Utility. Prima di iniziare l'installazione e la configurazione del PCIM, è una buona idea determinare quali indirizzi è possibile usare senza causare conflitti con altri dispositivi già installati sul calcolatore.

E' opportuno annotare queste informazioni per potervi fare riferimento durante la configurazione del PCIM.

Le risorse da riservare per il PCIM sono:

- Una porta I/O per l'indirizzo EEPROM impostato mediante i microinterruttori (stesso valore dell'indirizzo Controller Port specificato con la PCIM Configuration Utility).
- Un blocco di quattro porte I/O per *ciascuna* scheda figlia GENI. Questo blocco deve iniziare a un multiplo di 4 byte (0, 4, 8, C esadecimale.) nell'intervallo da 100 a 3E4 esadecimale.
- Un blocco di 4000 (esadecimale) byte di memoria condivisa per *ciascuna* scheda figlia GENI. Questo blocco deve iniziare a un multiplo di 16 KB (0000, 4000, 8000, C000 esadecimale.)
- Un IRQ libero (3, 4, 5, 6, 9, 10, or 11) da condividere fra tutte le schede figlia GENI del sistema.

Indirizzi suggeriti per evitare conflitti

La PCIM Configuration Utility (PCU) inserisce automaticamente i valori suggeriti nei campi Controller Port, I/O Port, Memory Base Addresses e IRQ Value. Questi valori sono presi dai valori suggeriti elencati qui sotto. Controllare la configurazione del calcolatore utilizzato per verificare che non vi siano conflitti.

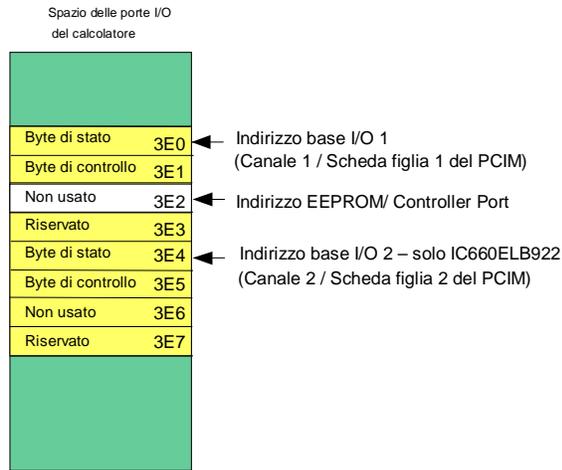
Indirizzo EEPROM (stesso valore di Controller Port Address)

Indirizzo di una porta I/O di un byte impostato mediante i microinterruttori del PCIM (EEPROM address) e configurato nella PCU (Controller Port address).

Questa porta I/O è utilizzata dalla PCU per scrivere i dati di configurazione del PCIM nella EEPROM del PCIM stesso e per permettere alle applicazioni di accedere a questi dati quando è necessario.

L'indirizzo di questa porta I/O è indipendente da quelli delle porte I/O usate dalle schede figlia (vedere indirizzo base I/O). Tuttavia, si suggerisce di impostarlo in una porzione inutilizzata dello spazio della porta I/O riservata alla prima scheda figlia GENI del PCIM (vale a dire 2 in più dell'indirizzo base I/O della scheda figlia). Questo byte non è utilizzato dal PCIM e il suo uso minimizzerà le probabilità di conflitti di indirizzi. **Vedere la sottostante figura.**

Esempio: Se si sceglie 3E0 come indirizzo base I/O per la prima scheda figlia GENI del PCIM (canale 1), usare 3E2 per l'indirizzo EEPROM (Controller Port).



Indirizzo base I/O

Questo è l'indirizzo iniziale dei quattro byte che devono essere riservati nel calcolatore per ciascuna scheda figlia GENI (canale). Questo indirizzo deve essere sulla delimitazione di un gruppo di 4 byte.

Gli indirizzi che sono normalmente OK per questo scopo sono i seguenti:

3E0	348	2E0	228
3E4	34C	2E4	22C
340		220	
344		224	

Controllare la configurazione di sistema del calcolatore per verificare che non vi siano conflitti.

Indirizzo base di memoria

Questo è l'indirizzo iniziale del blocco di 16 KB di memoria che devono essere riservati sul calcolatore per ciascuna scheda figlia GENI (canale) installata sul PCIM. Questo indirizzo deve essere sulla delimitazione di un blocco di 16 KB.

I seguenti indirizzi sono normalmente OK, se non sono utilizzati l'hardware o il software indicati.

B0000	Usato per la memoria del video monocromatico
B4000	Usato per la memoria del video monocromatico
C8000	Normalmente OK
CC000	Usato per la scheda GE Fanuc Parallel WSI
D0000	Usato per la memoria EMS
D4000	Usato per la memoria EMS
D8000	Normalmente OK
DC000	Normalmente OK
E0000	Normalmente OK
E4000	Normalmente OK
E8000	Normalmente OK
EC000	Normalmente OK

Controllare la configurazione di sistema del calcolatore per verificare che non vi siano conflitti.

IRQ

Interrupt per il PCIM.

Controllare se sono già utilizzati gli IRQ 10 e 11. Gli IRQ che si possono configurare per il CPIM (3, 4, 5, 6, 9, 10, or 11) devono essere condivisi fra tutte le schede figlia GENI installate sul sistema. *Controllare la configurazione di sistema del calcolatore per verificare che non vi siano conflitti.*

Alcune associazioni normali degli IRQ sono elencate qui sotto:

IRQ3 Porta seriale COM2:
IRQ4 Porta seriale COM1:
IRQ5 Rete/LPT2:
IRQ6 Floppy disk

Determinazione dei conflitti per le risorse porta I/O, memoria e Interrupt

Durante l'installazione e la configurazione del PCIM, si definiscono l'indirizzo EEPROM, l'indirizzo del controllore della porta, l'indirizzo base di memoria e l'IRQ. I valori scelti per questi parametri devono essere scelti in modo da non entrare in conflitto con quelli usati per gli altri dispositivi installati sul computer. La procedura da seguire a questo scopo è la seguente

Risorse di Windows NT 4.0

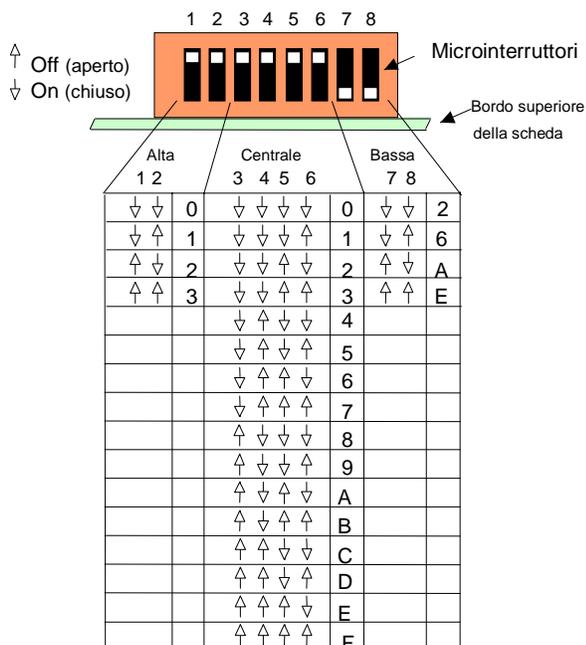
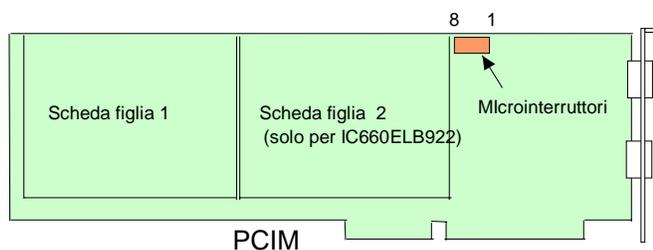
Eseguire il log in con i privilegi dell'amministratore del sistema.

1. Premere il pulsante Avvio.
2. Nel menu avvio, scegliere Programmi, Strumenti amministrativi, Diagnostica di Windows NT.
3. Cliccare sulla linguetta Risorse, poi ricorrere alle finestre di dialogo Porte I/O, Memoria e IRQ per individuare gli indirizzi non utilizzati.
4. Trovare un blocco libero (non elencato) di 4 porte nello spazio delle porte I/O per ciascuna scheda figlia GENI. L'indirizzo del primo byte del blocco è l'indirizzo base I/O (*I/O Base Address*) per un canale configurato mediante l'utilità di configurazione del PCIM (PCU).
5. Trovare un blocco libero (non elencato) di 4000 (esadecimale) byte di memoria per ciascuna scheda figlia GENI. L'indirizzo iniziale di questo blocco è l'indirizzo base di memoria (*Memory base address*) per un canale configurato mediante l'utilità di configurazione del PCIM (PCU).
6. Trovare un IRQ libero (non elencato) per l'*Interrupt* configurato nell'utilità di configurazione del PCIM (PCU).

Impostazione dei microinterruttori del PCIM

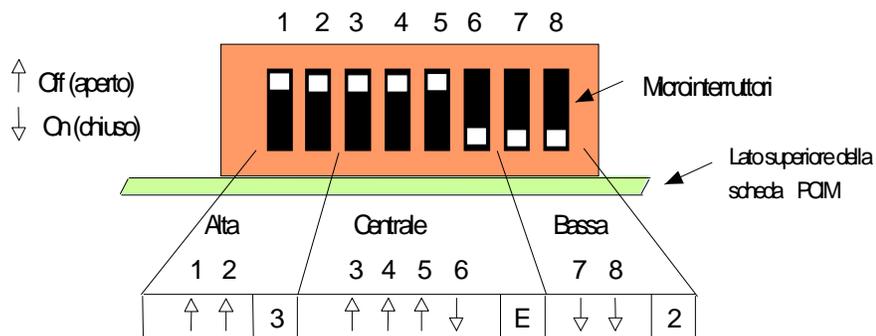
La porta I/O specificata mediante i microinterruttori è utilizzata dalla PCU (utilità di configurazione del PCIM) per scrivere nella EEPROM del PCIM i dati di configurazione del medesimo e per consentire alle applicazioni di accedere a questi dati.

I microinterruttori sono numerati da 1 a 8. Usare i microinterruttori 1 e 2 per impostare la cifra esadecimale più significativa, i microinterruttori 3, 4, 5 e 6 per impostare la cifra esadecimale centrale, e i microinterruttori 7 e 8 per impostare la cifra esadecimale meno significativa.



Esempio di impostazione dei microinterruttori

Se la prima scheda figlia GENI (canale 1) deve essere configurata a 3E0 esadecimale, si suggerisce di impostare i microinterruttori della EEPROM a. Questo esempio mostra come impostare a 3E2 l'indirizzo della EEPROM.



Installazione del PCIM sul Computer

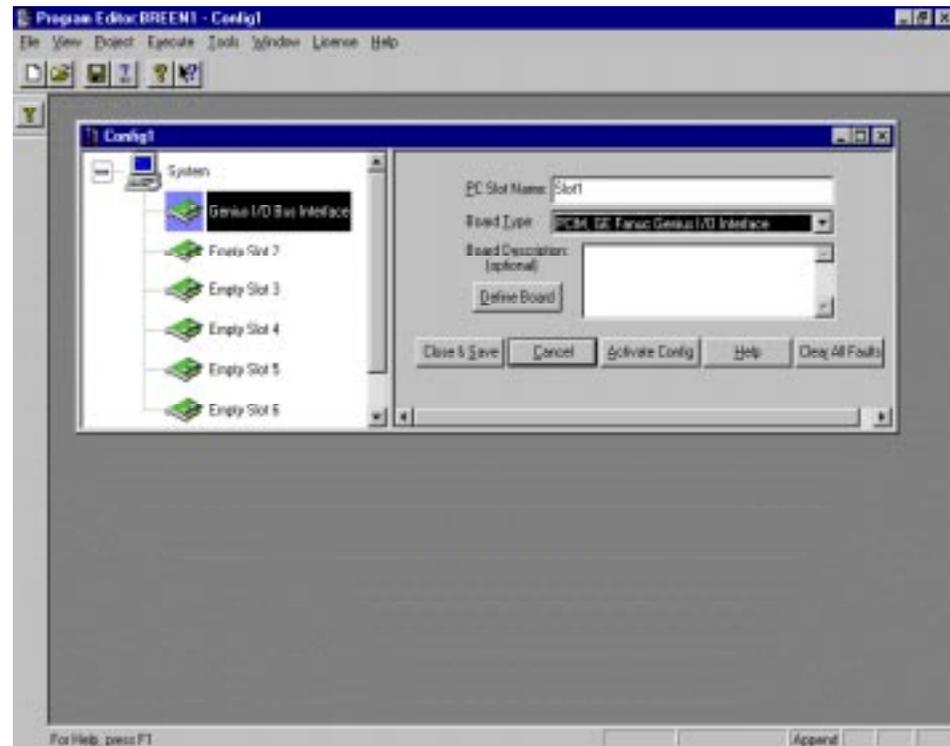
1. Spegner il computer e scollegare il cavo di alimentazione dalla presa di corrente.
2. Installare il PCIM seguendo le istruzioni fornite dal costruttore per l'installazione di schede opzionali.
3. Collegare il bus al PCIM (se lo si desidera, questo passo può essere eseguito dopo aver configurato il PCIM).
4. COSE DA NON FARE:
 - Montare il PCIM in una posizione in cui sia ostruito il flusso d'aria che lo attraversa.
 - Montare il PCIM a meno di 1/8 di pollice (0.318 cm) da qualsiasi altra scheda o componente del rack.
 - Usare adesivi o rivestimenti conformabili su qualsiasi parte del PCIM.

Configurazione del PCIM

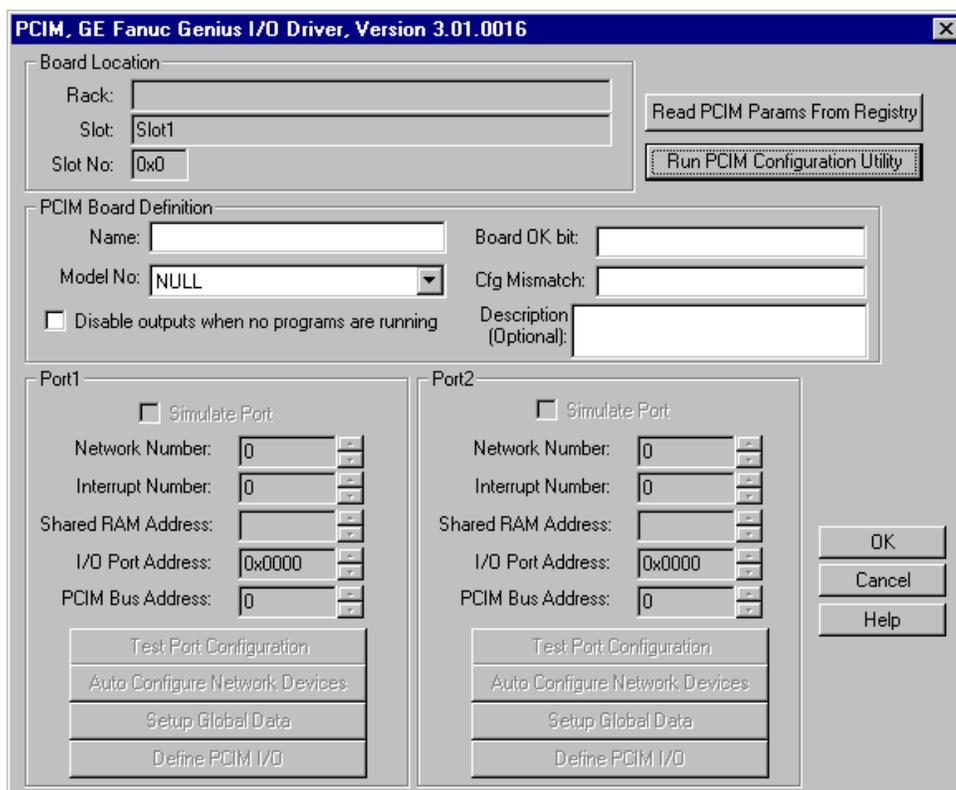
Prima di configurare il PCIM, accertarsi che le porte I/O, gli indirizzi base di memoria e l'IRQ selezionati siano effettivamente disponibili. Fare riferimento alla precedente sezione "Risorse del computer da riservare".

Per configurare il PCIM:

1. Avviare PC Control, Program Editor.
2. Nel menu File, scegliere New o Open Config per creare una nuova configurazione del sistema o per aprirne una esistente. Comparirà lo schermo della configurazione del sistema.



3. Cliccare su un'icona "Empty Slot" sotto l'icona "System".
4. Nella lista a discesa del campo Board Type scegliere PCIM, poi cliccare sul pulsante Define Board. Comparirà la finestra di dialogo PCIM-Genius I/O Driver.

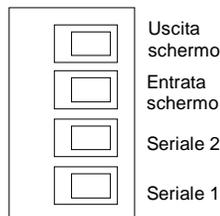


5. Avviare l'utilità PCU cliccando sul pulsante Run PCIM Configuration Utility. Nella PCU si dovranno specificare vari dati. Per questo, fare riferimento alla lista degli indirizzi e degli IRQ disponibili, preparata precedentemente. Durante il processo di configurazione è possibile ricorrere all'aiuto in linea.
6. Dopo aver configurato il PCIM ed essere usciti dalla PCU, cliccare sul pulsante Read PCIM Params From Registry. Comparirà una finestra contenente l'elenco dei PCIM configurati.
7. Selezionare il PCIM appropriato e cliccare su OK. Questo aggiornerà i parametri della porta nella finestra di dialogo Genius I/O Driver con i valori specificati per l'IRQ (Interrupt number), l'indirizzo base di memoria (Shared RAM Address), l'indirizzo base I/O (I/O Port Address) e l'indirizzo del bus del PCIM (PCIM Bus Address).
8. Alla fine cliccare sul pulsante Test Port Configuration. Dovrebbero accendersi i LED Scheda OK e Comunicazioni.

In caso di problemi, fare riferimento alla sezione "Ricerca guasti".

Connessione del PCIM al bus

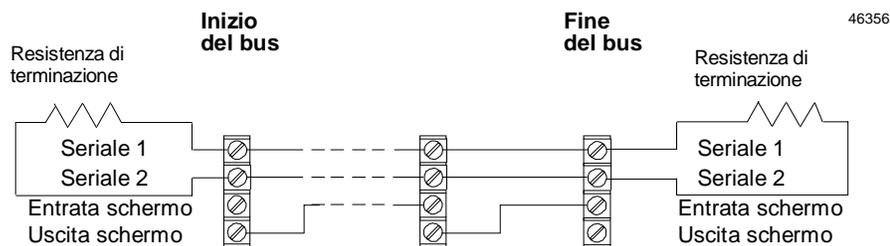
Le unità possono essere messe sul bus in qualunque sequenza fisica. Ciascun connettore del PCIM ha quattro terminali per il cavo del bus (Seriale 1, Seriale 2, Entrata schermo e Uscita schermo). Notare che la sequenza di questi terminali sul connettore del PCIM non è la stessa di altre unità collegate al bus (ad esempio, i blocchi di I/O).



Questi terminali accettano due fili AWG #20 (sezione di 0.54mm²) più un filo di connessione di una resistenza da 0.25 Watt (opzionale: usata per la terminazione del bus). La dimensione minima raccomandata del filo è AWG #22 (sezione di 0.36mm²).

Collegare il terminale Seriale 1 di ciascun connettore ai terminali Seriale 1 dell'unità precedente e di quella successiva. Collegare il terminale Seriale 2 di ciascun connettore ai terminali Seriale 2. Se il PCIM ha due schede figlia, esse possono essere collegate a bus differenti o allo stesso bus.

Il terminale Entrata schermo di ciascun connettore deve essere collegato al terminale uscita schermo dell'unità precedente. Per la prima unità del bus, il terminale Entrata schermo può non essere collegato. Per l'ultima unità può non essere collegato il terminale Uscita schermo.



Nell'esecuzione delle connessioni del bus, la lunghezza massima del filo nudo esposto deve essere di due pollici. Come ulteriore protezione, ciascun filo di scarico dello schermo dovrebbe essere protetto mediante guaina isolante per evitare che i fili di entrata e di uscita dello schermo entrino in contatto fra loro.

Terminazione del bus

Un bus deve essere terminato a ciascuna estremità con una resistenza di valore uguale all'impedenza caratteristica del cavo. L'impedenza sarà di 75, 100, 120, o 150 ohm. Se un connettore del PCIM si trova ad una o all'altra estremità del suo bus, installare la resistenza di terminazione appropriata attraverso i terminali Seriale 1 e seriale 2. Il *Genius I/O System and Communications User's Manual* elenca le resistenze di terminazione appropriate per ciascun tipo raccomandato di cavo del bus.

Rimozione del PCIM dal Bus

I connettori di bus del PCIM sono rimovibili. Essi possono essere rimossi durante il funzionamento del sistema senza compromettere l'integrità dei dati sul bus. Per rimuovere un connettore di bus Se un cavo operativo è attaccato al bus, fare molta attenzione ad evitare che i fili del bus si tocchino o tocchino qualsiasi altra cosa. Non appoggiare il connettore su una superficie conduttrice di elettricità.

Non si devono mai rimuovere *singoli fili del bus* dai connettori terminali mentre il bus è in funzione; la presenza sul di dati inaffidabili che ne risulterebbe potrebbe generare condizioni di grave pericolo.



Specifiche

Numeri di catalogo

PCIM a 1 canale IC660ELB921

PCIM a 2 canali IC660ELB922

LED (2 per ciascuna scheda figlia) GENI OK, COMMS OK (Comunicazioni OK)

Elettriche

Alimentazione 5 volt CC +/- 10%, 400 ma (massimo)

Carico d'ingresso 1 carico LS TTL per linea di ingresso

Capacità di pilotaggio 10 carichi LS TTL per linea di uscita

Meccaniche

Scheda PCIM Scheda monoslot tipo "AT"

Connessione monitor portatile Connettore esterno con HHM e terminali bus

Connessione bus seriale Terminali sul bordo della scheda o connettore esterno. I terminali sul bordo della scheda accettano due fili AWG#20 (sezione di 0.55mm²) o tre fili AWG #22 (sezione di 0.36mm²).

Interfaccia con il backplane dell'Host Piena compatibilità con ISA

Necessità di memoria

Scheda madre 4 byte

Ciascuna scheda figlia 16 KB

Requisiti ambientali – In funzione

Temperatura da 0 a 60 gradi C (temperatura ambiente alla scheda)

Umidità dal 5% al 95% senza condensa

Altitudine 10000 piedi (3048 metri)

Vibrazioni spostamento di 0.2" (5.08 mm) da 5 a 10 Hz

1 G da 10 a 200 Hz

Urti 5 G, durata 10 ms per MIL-STD 810C, metodo 516.2

Requisiti ambientali – Non in funzione

Temperatura da -40 a 125 gradi C (Temperatura ambiente alla scheda)

Umidità dal 5% al 95% senza condensa

Altitudine 40000 piedi (12192 metri)

Vibrazioni spostamento di 0.2 pollici (5.08 mm) da 5 a 10 Hz

1 G, da 10 a 200 Hz

Urti 5 G, durata 10 ms per MIL-STD 810C, metodo 516.2

Caratteristiche elettriche del PCIM

Requisiti dell'alimentazione

Il PCIM richiede una sorgente di 5 volt CC per l'alimentazione della logica. La tensione fornita non deve variare di oltre il 10% sopra e sotto al valore nominale (non meno di 4.5 V CC e non più di 5.5 V CC), o il PCIM non funzionerà correttamente. Il PCIM con una sola scheda figlia (PCIM a 1 canale) assorbe tipicamente 1.0 A. Il PCIM con due schede figlia (PCIM a due canali) assorbe tipicamente 1.5 A.

Carichi del bus/Capacità di pilotaggio

Tutte le linee in ingresso al PCIM presentano non più di un carico LSTTL standard al connettore di interfaccia con l'host.

Tutte le linee in uscita dal PCIM sono capaci di pilotare 10 carichi LSTTL standard. Queste linee, con l'eccezione delle linee /INT e /PCIM OK, sono uscite a tre stati. La linea /INT è un'uscita open collector che può essere OR cablata a un singolo ingresso di interrupt. Le linee /PCIM OK e /COMM OK sono uscite open collector attive basse con limitatore di corrente a 10 mA incorporato adatte al pilotaggio diretto di LED.

Tutti i segnali in ingresso al PCIM dal sistema host sono visti dall'host come un carico LSTTL. Questi segnali sono TTL compatibili e commutano a livelli TTL.

I segnali di controllo in uscita verso l'host sono piloti LSTTL di tipo open collector con pull-up resistivo di 10K, capaci di assorbire 4 mA mantenendo una tensione in uscita di 0.4V o meno.

Il ricetrasmittitore di dati è un dispositivo LSTTL a tre stati capace di fornire o assorbire 12 mA con $V_{OL} = 0.4V$ e $V_{OH} = 2.0V$.

Il PCIM è totalmente compatibile con i backplane ISA.

Condizionamento dei segnali

Il PCIM ha due connettori ai quali si può accedere quando esso è installato in un rack di tipo PC. Ambedue i connettori servono per la connessione standard con copie intrecciate a un bus seriale.

Il monitor portatile può essere collegato attraverso un cavo di interfaccia ai connettori Genius separati.

Tutte le linee in ingresso da ambedue i connettori sono isolate o limitate a impedenza per proteggere il PCIM da picchi transitori di tensione o dall'applicazione di alte tensioni alle connessioni del bus seriale.

Ricerca guasti

Come nella messa a punto dei programmi, la ricerca dei guasti hardware/firmware si esegue pensando logicamente alla funzione di ciascun componente e a come queste funzioni interagiscono. Una comprensione di base del significato delle varie spie sarà di aiuto nel ricondurre rapidamente il problema al PCIM, al controllore del bus, a un rack di I/O, a un blocco di I/O o alla CPU.

Quando si verifica un problema, occorre considerare l'intero sistema. La CPU, l'Host, i blocchi di I/O e le unità esterne collegate al sistema Genius I/O o da questo controllate devono essere tutti funzionanti e connessi correttamente. Tutte le connessioni dei cavi così come le connessioni a vite o saldate devono essere controllate accuratamente.

Concetto di sostituzione del modulo

Quando si manifesta un problema, isolare prima l'insieme principale che ne è responsabile, poi individuare il modulo difettoso in quell'insieme. Il modulo difettoso sarà quindi sostituito prelevandolo dal set duplicato di moduli mantenuto in sito. La linea di produzione, o il sistema, sarà rapidamente di nuovo in funzione.

Il modulo difettoso potrà essere restituito attraverso i normali canali per la sostituzione in garanzia o per la riparazione senza fermare la linea di produzione o il sistema per un lungo periodo di tempo. Il concetto di sostituzione minimizza il tempo di fermo per guasti riducendolo a minuti piuttosto che a possibili giorni. I risparmi potenziali superano di gran lunga il relativamente piccolo costo sostenuto per la duplicazione dei moduli.

Se insieme al sistema non è stato inizialmente acquistato un set duplicato di moduli, si raccomanda di farlo, contattando il distributore autorizzato GE Fanuc locale. Poi, con l'aiuto di questo manuale e del personale del distributore autorizzato GE Fanuc locale, sarà possibile individuare ed eliminare quasi tutti i problemi che potranno manifestarsi.

Ricerca guasti del PCIM

LED

Normalmente, un guasto del PCIM che ne provochi un funzionamento improprio può essere isolato controllando i LED di stato del PCIM stesso. La condizione normale dei LED è lo stato ON. Se un LED non è ON, eseguire la procedura di ricerca guasti descritta in questa sezione.

Indicatore	Stato	Definizione
BOARD OK	ON	Il PCIM è alimentato (per funzionare correttamente il PCIM deve essere adeguatamente alimentato) e il test autodiagnostico della scheda è stato superato.
	OFF	Timeout del temporizzatore watchdog. Questo indica un guasto della scheda, l'assegnazione errata di un indirizzo o lo stato basso della linea di ingresso /RST.
COMM OK	ON	L'alimentazione elettrica è disponibile, l'hardware per le comunicazioni del controllore è funzionante e può trasmettere dati (riceve il token) ad ogni scansione del bus seriale.
	OFF	(o LAMPEGGIANTE) significa che è stato rilevato un errore nell'hardware per le comunicazioni o nell'accesso al bus seriale Genius.

Isolamento e riparazione dei guasti

Se i LED di stato sono OK ma il bus non sta funzionando correttamente, il problema può corrispondere a una delle seguenti descrizioni. Se è così, seguire la procedura proposta per il sintomo appropriato.

- Uno dei LED non si accende quando il PCIM è inserito e alimentato.

BOARD OK OFF e COMM OK ON -

- Controllare i parametri introdotti con il software di configurazione.

BOARD OK non va ON -

- Controllare che il PCIM sia inserito completamente nel connettore del backplane dell'host e che tutti i pin del connettore siano correttamente allineati.

Se tutto appare in ordine sostituire il PCIM.

BOARD OK ON e COMM OK OFF -

- Controllare che il tipo e la lunghezza del cavo siano corretti (fare riferimento al *Genius I/O System and Communications User's Manual*, GEK90486-1).
- Controllare che alle due estremità del bus siano installare resistenza di terminazione appropriate (fare riferimento al *Genius I/O System and Communications User's Manual*, GEK90486-1).
- Controllare che il cablaggio del bus seriale sia stato completato e formi un collegamento a margherita.
- Accertarsi che i cavi non passino in prossimità di linee ad alta tensione.
- Controllare che i cavi non siano interrotti.

Ambedue i LED sono OFF -

- Controllare che il PCIM sia inserito, correttamente alloggiato e alimentato.

I due LED lampeggiano contemporaneamente -

- Due unità sullo stesso bus hanno probabilmente ricevuto lo stesso numero di unità (indirizzo del bus seriale).

Controllare mediante l'HHM.

- Errori ripetuti del bus
 - Assicurarsi che la schermatura del cavo sia installata e messa a terra correttamente (fare riferimento al *Genius I/O System and Communications User's Manual*, GEK904861).
 - Staccare dal PCIM il cavo delle comunicazioni con il bus, recuperare la tabella dei numeri di unità usata per configurare il sistema e usare l'HHM per leggere la configurazione e confrontare i numeri di unità e i numeri di riferimento degli I/O.
 - Se tutto appare in ordine, sostituire il PCIM
- Il sistema si ferma indicando un errore di parità

Riferimenti PCIM/I/O duplicati o sovrapposti.

- Ingresso duplicato sullo stesso bus.
- Sovrapposizione di riferimenti a ingressi da altri PCIM.
- Errori del bus -impossibile mettere in funzione il PCIM
 - Seriale 1/Seriale 2 incrociati.
- Mancanza intermittente o totale di comunicazioni
 - Baud rate diversi

Accendere i blocchi uno per volta e controllare il rispettivo baud rate. La variazione del baud rate di un blocco non ha effetti finché il blocco non viene spento e riacceso.

- Nessun dato globale
 - Unità di destinazione fuori linea.

Verificare che la destinazione sia in linea.

- Completamento datagramma senza successo.
 - Unità di destinazione fuori linea.

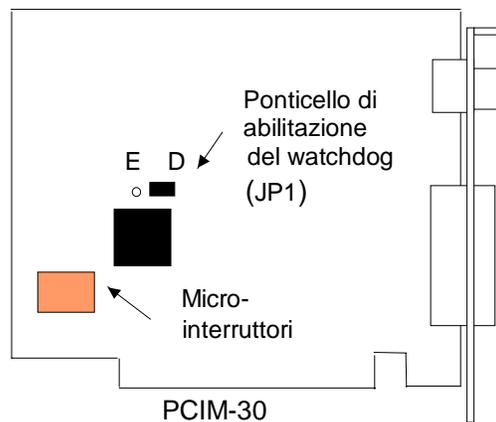
Verificare che la destinazione sia in linea.

Appendice **B**

Interfaccia con personal computer per I/O della Serie 90-30 (PCIF)

La scheda di interfaccia con personal computer per I/O della Serie 90-30 (IC693PIF301) fornisce l'accesso agli I/O di un massimo di quattro espansioni o rack remoti della Serie 90-30. Questa scheda half-slot è concepita per calcolatori compatibili con il PC/AT IBM. La scheda PCIF-30 supporta tutti i moduli di I/O discreti e analogici, oltre ad alcuni moduli speciali.

La connessione dalla scheda ai rack usa lo stesso cavo usato per un'espansione o un sistema remoto standard della Serie 90-30. Inoltre la scheda è dotata di un watchdog hardware che forza tutte le uscite nella condizione di reset e apre un relè se, per qualsiasi motivo, il PC smette di aggiornare le uscite.



Installazione della scheda PCIF-30

Per installare la scheda PCIF-30, si devono riservare otto porte I/O contigue nello spazio delle porte I/O del computer. Questo viene fatto impostando l'indirizzo della scheda mediante i microinterruttori. Prima di iniziare l'installazione, è una buona idea determinare quali porte si possono utilizzare senza causare conflitti con le altre unità installate sul computer.

Come iniziare

1. Leggere il paragrafo "Indirizzi suggeriti per evitare conflitti".
2. Determinare se gli indirizzi (risorse) suggeriti sono effettivamente disponibili nel sistema
3. Impostare i microinterruttori della scheda PCIF-30 e installarla sul calcolatore.
4. Mettere il ponticello di abilitazione del Watchdog su "D" (disabilitato).

Indirizzi suggeriti per evitare conflitti

Indirizzo della scheda

Questo è l'indirizzo iniziale degli otto byte che devono essere riservati alla scheda PCIF nello spazio delle porte I/O del calcolatore. Questo indirizzo deve essere su una delimitazione di 8 byte.

Normalmente, possono essere usati i seguenti indirizzi:

310	348	2E0	228
318	34C	2E4	22C
340	3E0	220	
344	3E4	224	

Controllare la configurazione di sistema del calcolatore per verificare che non vi siano conflitti.

Determinazione dei conflitti fra le porte I/O

Quando si installa la scheda PCIF-30, si definisce l'indirizzo della scheda mediante i microinterruttori. Questo indirizzo non deve essere in conflitto con quelli usati per altri dispositivi installati sul calcolatore. Per visualizzare le risorse usate dai dispositivi attualmente installati sul calcolatore procedere come segue.

Risorse di Windows NT 4.0

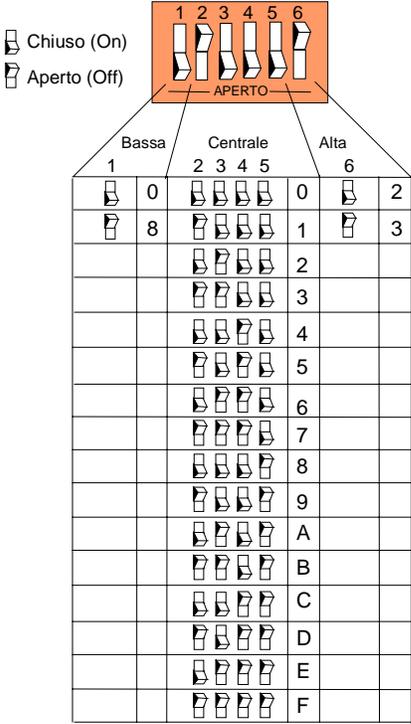
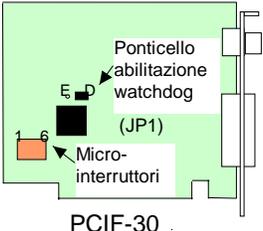
Eeguire il log in con i privilegi dell'amministratore del sistema.

1. Premere il pulsante Avvio
2. Nel menu avvio, scegliere Programmi, Strumenti amministrativi, Diagnostica di Windows NT.
3. Cliccare sulla linguetta Risorse, poi ricorrere alle finestra di dialogo Porte I/O per individuare gli indirizzi non utilizzati.
4. Trovare un blocco di otto porte I/O (otto byte) non utilizzato (non listato) per la scheda PCIF-30. La prima porta del blocco è l'indirizzo della scheda.

Impostazione dei microinterruttori e del ponticello di abilitazione del Watchdog sulla scheda PCIF-30

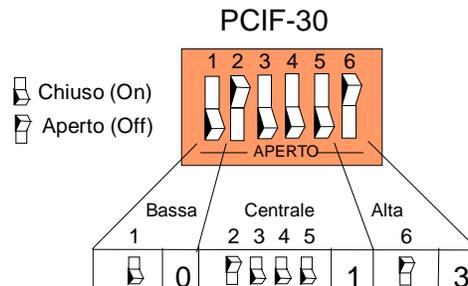
L'indirizzo impostato con i microinterruttori è utilizzato per riservare le porte I/O di cui la scheda PCIF-30 ha bisogno per funzionare.

I microinterruttori sono numerati da 1 a 6. **Usare il microinterruttore 6 per impostare la cifra esadecimale più significativa, i microinterruttori 5, 4, 3 e 2 per impostare la cifra esadecimale centrale e il microinterruttore 1 per impostare la cifra decimale meno significativa.**



Esempio di impostazione dei microinterruttori

Questo esempio mostra come mettere l'indirizzo della scheda a 310.



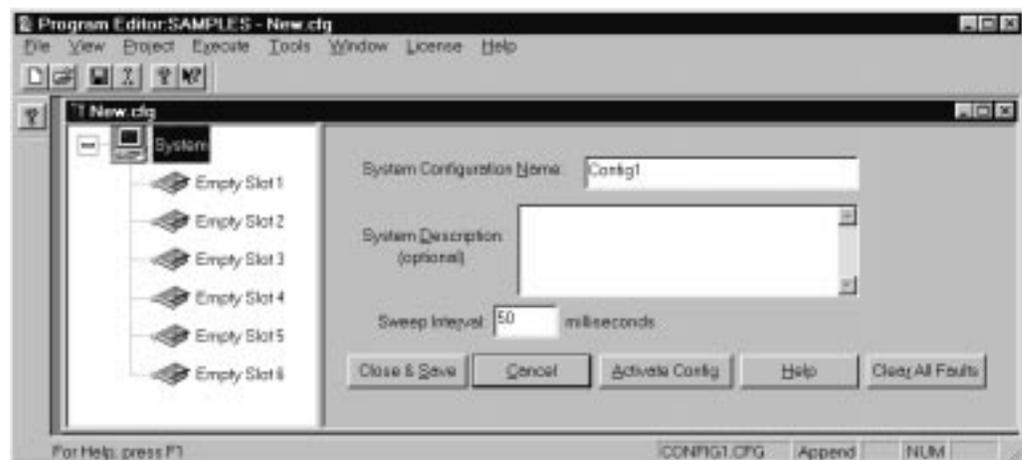
Configurazione della scheda PCIF-30

1. Avviare il Program Editor del software PC Control.

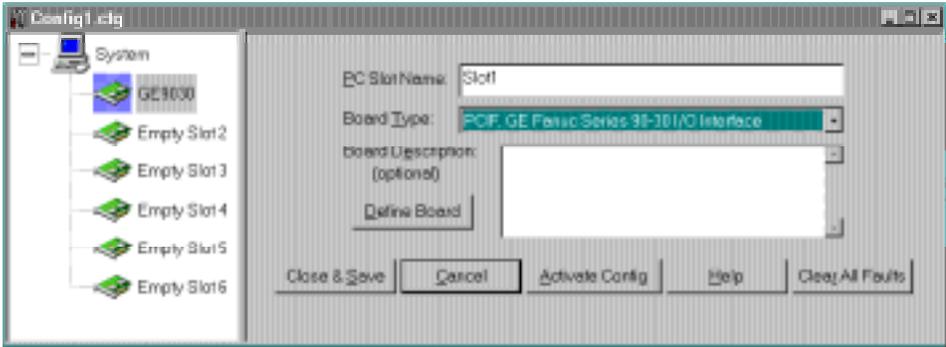
Nel menu Avvio, selezionare PC Control Applications, Program Editor.

2. Configurare la scheda PCIF-30. [IPC](#)

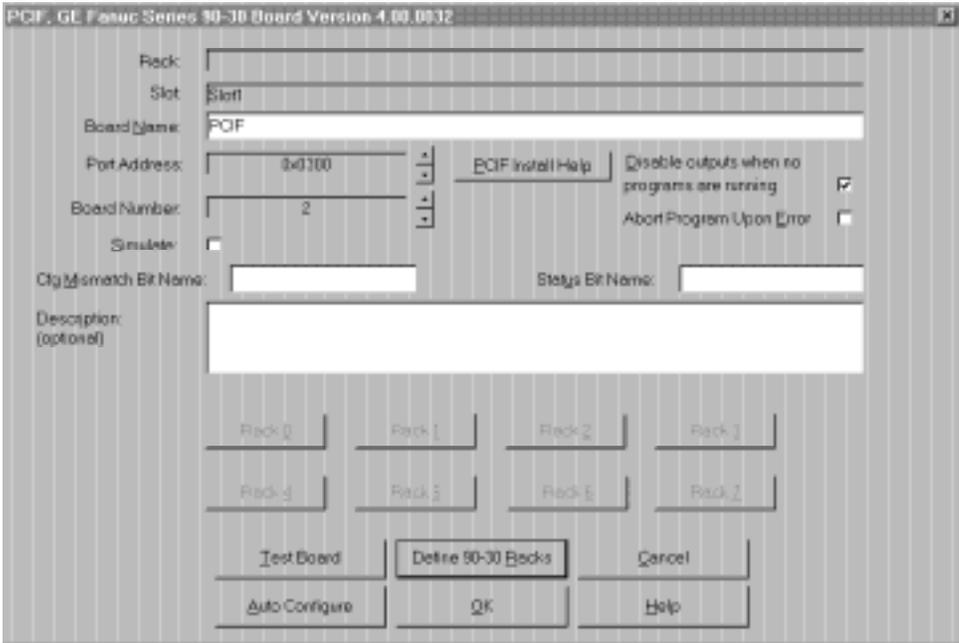
A. Nel menu file del Program Editor, selezionare Open Config (or New Config). Comparirà la seguente finestra di configurazione.



B. Nell'albero visualizzato nella parte sinistra della finestra, selezionare lo slot che si desidera configurare. Comparirà la seguente finestra:



C. -Nel campo Board Type, selezionare PCIF, GE Fanuc Serie 90-30 I/O Interface. Cliccare sul pulsante Define Board Comparirà la finestra di dialogo PCIF Board.



D. Impostare i parametri di configurazione della scheda PCIF-30. Accertarsi di avere impostato correttamente il campo Port Address (indirizzo porta). Il valore di questo campo deve corrispondere a quello impostato sulla scheda con i microinterruttori (pagina B-3).

Per ulteriori dettagli relativi alla configurazione, consultare l'aiuto in linea del PC Control.

E. Dopo aver impostato l'indirizzo della porta, cliccare sul pulsante Test Board per verificare che l'installazione sia corretta. (Il sistema di runtime non deve essere attivo.) Dovrebbe comparire la finestra di dialogo Board Test, contenente il messaggio " PCIF Board found, shared RAM access successfully found" (Scheda PCIF trovata, accesso alla RAM condivisa eseguito con successo). Se questo messaggio non compare, tentare di configurare un altro blocco di RAM condivisa.

2. Configurazione degli I/O della Series 90-30.

Dalla finestra di dialogo PCIF Board è possibile continuare il processo di configurazione.

- **Configurazione automatica:** Se i rack e i moduli I/O sono installati e alimentati, e se il sistema di runtime non è attivo, è possibile usare questa prestazione. Quando si clicca sul pulsante Auto Configure, il software attiva la scheda PCIF-30 e legge le informazioni relative ai rack e ai moduli collegati
- **Configurazione manuale:** Per usare questa prestazione, cliccare sul pulsante Define 90-30 Racks. Comparirà la finestra di dialogo Rack Definition.

Per ulteriori dettagli relativi alla configurazione, consultare l'aiuto in linea del PC Control.

Introduzione

La scheda di interfaccia con personal computer IC693PIF400 permette al software PC Control di comunicare con un massimo di sette espansioni o rack di I/O remoti della Serie 90-30. I rack di I/O possono essere installati fino a 700 piedi (circa 213 metri) dal personal computer. Attraverso la scheda PIF400, il software PC Control può gestire fino a 25886 byte di I/O. I limiti effettivi della capacità di I/O dipendono dai requisiti di scansione e dalla configurazione degli I/O.

La scheda PIF400 si inserisce in uno slot a 16 bit del bus PC/AT/ISA del personal computer. La connessione delle espansioni e dei rack remoti alla scheda si realizza con un collegamento a margherita usando un connettore D femmina a 25 pin. La scheda PIF400 fornisce un segnale di uscita con watchdog RUN e un relè che ne permettono l'integrazione con circuiti di sicurezza. Per le specifiche, fare riferimento al foglio dati fornito insieme alla scheda di interfaccia, GFK-1540.

Questa appendice fornisce le seguenti informazioni:

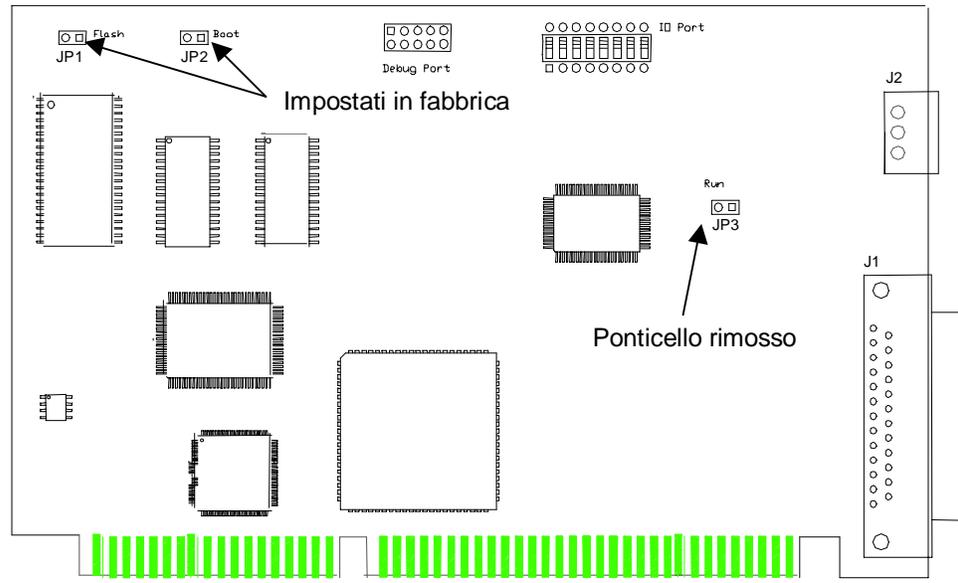
- Compatibilità
- Descrizione dell'hardware
- Ponticelli
- Connettori
- Microinterruttori
- Guida per l'avviamento rapido

Compatibilità

La scheda PIF400 supporta tutti i moduli di I/O discreti e analogici della Serie 90-30 che possono essere configurati dal PC Control.

Descrizione dell'hardware

La scheda PIF400 ha un ponticello configurabile, un connettore per il cavo degli I/O e un relè RUN.



Ponticelli

JP3 – Attivazione all'accensione

Questo ponticello può essere usato per determinare se la scheda PIF400 richiederà un segnale di Start dal PC per attivarsi.

Se J3 è rimosso, la scheda PIF400 non si attiverà finché non riceverà un segnale di Start dal PC. Se il PC viene resettato, la scheda PIF400 si resetterà e attenderà un nuovo segnale di Start dal PC

Se J3 è installato, la scheda PIF400 si attiva non appena il PC viene alimentato. Se il PC viene resettato, la scheda PIF400 *non* si resetta, ma rimane attiva.

Lo stato per difetto di JP3 è *rimosso*.

JP1 – Flash Protect e JP2 – Boot Protect

Questi ponticelli sono impostati in fabbrica e devono essere lasciati installati per il buon funzionamento della scheda PIF400.

Connettori

J1 – Connettore espansione / I/O remoti

Questo connettore D femmina a 25 pin si usa per comunicare con un massimo di sette rack di I/O. Questo connettore è collegato alla porta di espansione sul primo rack di I/O del sistema.

Assegnazione dei pin del connettore di espansione J1

Pin	Nome del segnale	Descrizione	Direzione
1	SHLD	Schermo	N/A
7	GND	Terra	N/A
2	DFRAME+	Coppia segnale Data frame	Uscita
3	DFRAME-		
8	RUN+	Coppia segnale RUN	Uscita
9	RUN-		
12	PERR+	Coppia segnale Parity error	Ingresso
13	PERR-		
16	DATA+	Coppia segnale Data	Ingresso e Uscita
17	DATA-		
20	RSEL+	Coppia segnale Bus select	Uscita
21	RSEL-		
24	IOCLK+	Coppia segnale Data clock	Uscita
25	IOCLK-		

Il cavo di connessione fra la scheda PIF400 e i rack di I/O deve essere fornito dal cliente. Questo cavo deve essere formato da sette coppie intrecciate, con schermatura totale e filo di scarico (Belden 8107 o equivalente). L'ultimo rack del bus I/O deve essere terminato con I/O Bus terminator plug, numero di catalogo IC693ACC307.

Per le interconnessioni si possono utilizzare cavi di espansione GE Fanuc di lunghezza standard, elencati qui sotto, o cavi di lunghezza variabile. I rack remoti possono essere posti fino a 700 piedi (213 metri) dal PC e i rack di espansione possono essere posti fino a 50 piedi (15 metri) dal medesimo.

Tipo di cavo	Codice parte
Cavo T, 1 metro	IC693CBL300
Cavo T, 2 metri	IC693CBL301
Cavo punto a punto, 15 metri	IC693CBL302

J2 – Connessione del relè RUN (Temporizzatore watchdog)

Quando è il segnale run verso il backplane 90-30, il relè che controlla questo connettore è eccitato.

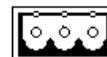
Il relè RUN può essere cablato in modo da comandare un dispositivo esterno di sicurezza nel caso di un errore del calcolatore o del software applicativo.

Durante il funzionamento normale con I/O attivi, il temporizzatore watchdog della scheda PIF400 viene continuamente resettato per mantenere lo stato di eccitazione (non normale) del relè. Se il software applicativo non aggiorna le uscite o accede al contatto RUN, il temporizzatore watchdog disattiva tutte le uscite dopo un secondo. Questo significa che tutti i circuiti dei moduli di uscita vanno nello stato off, il LED RUN dell'alimentatore di ciascun rack si spegne e il relè assume il suo stato normale (non eccitato).

Assegnazione dei pin di J2, Connessione del relè RUN

Pin	Nome del segnale
1	Comune
2	Contatto relè normalmente chiuso (NC)
3	Contatto relè normalmente aperto (NO)

COM NC NO



Specifiche dei contatti del relè RUN

Resistenza iniziale	50 milliohm
Massima potenza di commutazione	60 Watt, 62.5 VA
Massima tensione di commutazione	220 VCC, 250 VCAAC
Massima corrente di commutazione	2 Amp
Massima corrente sopportata	3 Amp
Valori UL/CSA	125 VAC @ 0.3 A 110 VDC @ 0.3 A 30 VDC @ 1.0 A
Operazioni minime	
Meccanica	100000000
Elettrica	500000 (30 VDC @ 1.0 A, resistivo) 100000 (30 VDC @ 2.0 A, resistivo)

P1 –Porta programmatore

Non ci devono essere ponticelli su questi pin.

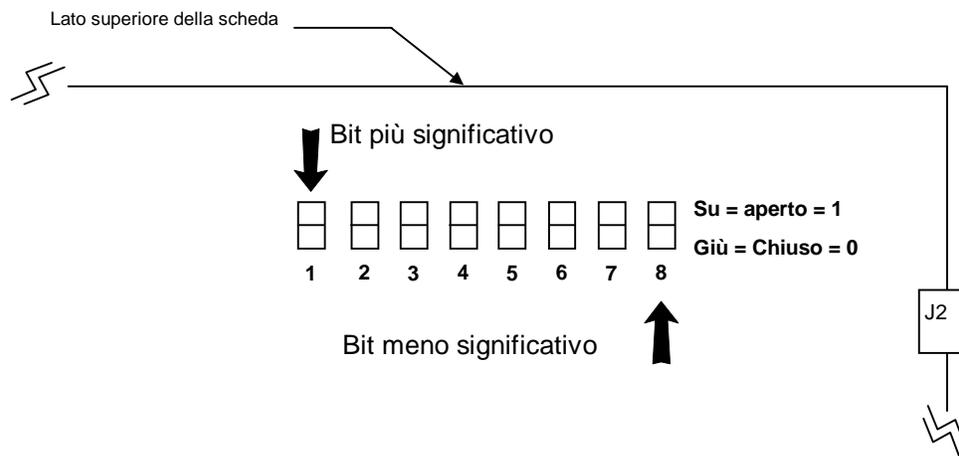
Microinterruttori

S1 – Interruttori di selezione indirizzo porta I/O Questi otto microinterruttori selezionano l'indirizzo base nello spazio degli indirizzi I/O del PC al quale risponde la scheda PIF400.

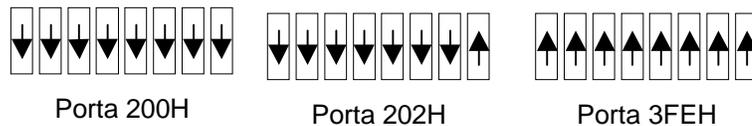
La scheda PIF400 usa due porte I/O adiacenti a partire da qualsiasi delimitazione di 2 byte da 200 esadecimale a 3FE esadecimale. L'impostazione standard eseguita in fabbrica è 200 esadecimale. Le impostazioni corrispondenti a tutti i possibili indirizzi sono elencate nel foglio dati fornito insieme alla scheda di interfaccia.

Nota

S1 può avere differenti stili, differenti marcature o ambedue le cose. **Basare l'impostazione degli interruttori sulla loro posizione fisica, come si vede qui sotto, e non sulla numerazione che compare sulla scheda.**



Posizione dei microinterruttori S1



Esempi di impostazione dei microinterruttori S1

Guida all'avviamento rapido

1. Controllo delle risorse del sistema

La scheda PIF400 richiede due porte adiacenti nello spazio delle porte I/O del PC. Questo si riservano impostando l'indirizzo della scheda mediante i microinterruttori S1. Prima di installare la scheda PIF400 nel PC occorre determinare quali porte è possibile usare senza generare conflitti con altri dispositivi già installati nel calcolatore.

La scheda PIF400 richiede 32Kbytes (7FFFH esadecimale) di RAM condivisa. L'indirizzo iniziale di questa memoria è configurato dal software PC Control (l'indirizzo per difetto è D0000 esadecimale). Sarà necessario verificare che questa memoria non sia già utilizzata da un altro dispositivo.

Per individuare le risorse già in uso, seguire le istruzioni fornite qui sotto.

- A. Eseguire il log in con i privilegi dell'amministratore del sistema.
- B. Nel menu avvio di Windows NT®, scegliere Programmi, Strumenti amministrativi, Diagnostica di Windows NT.
- C. Cliccare sulla linguetta Risorse, poi cliccare sul pulsante I/O per visualizzare la finestra di dialogo Porte I/O e individuare gli indirizzi non utilizzati.
- D. Trovare un blocco libero (non elencato) di 2 porte nello spazio delle porte I/O per la scheda PIF400. I microinterruttori S1 devono essere impostati in modo da selezionare la prima porta del blocco. L'impostazione standard eseguita in fabbrica di S1 è 200 esadecimale; se le porte 200—202 sono già utilizzate, sarà necessario cambiare impostazione di S1 (vedere il passo 2, "Impostazione dei microinterruttori").
- E. Per verificare la memoria disponibile cliccare sulla linguetta Memory. Trovare un blocco disponibile di 32 KB e prendere nota del suo indirizzo per usarlo durante la configurazione della scheda PIF400. Se il blocco specificato per difetto (D0000—D7FFF) è già utilizzato, sarà necessario cambiare l'impostazione per difetto per selezionare un altro blocco di memoria.

2. Impostazione dei microinterruttori

Nota

Non è necessario cambiare l'impostazione dei microinterruttori S1, a meno che essa non generi conflitti con altre schede di interfaccia installate sul calcolatore. L'impostazione standard eseguita in fabbrica è 200 esadecimale.

Se è necessario cambiare l'impostazione dei microinterruttori, vedere i dettagli a pagina C-6.

3. Installazione sul PC della scheda PIF400

Nota

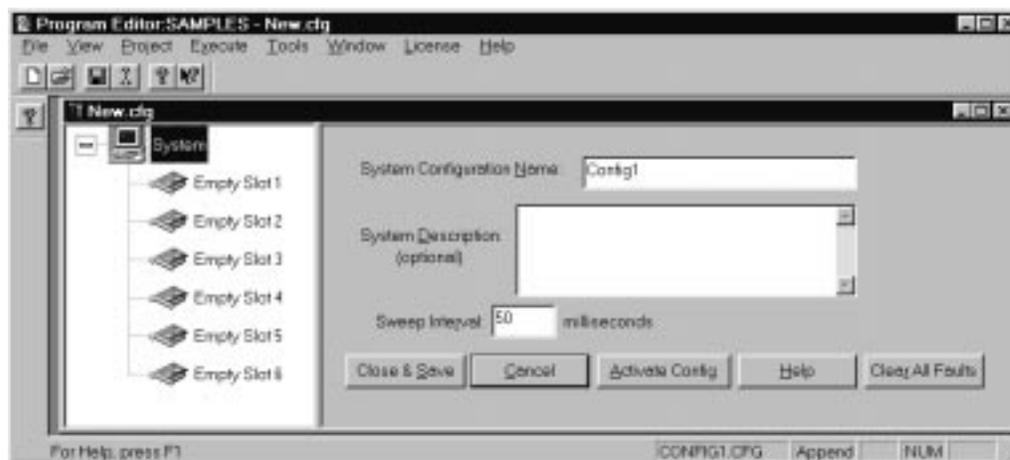
Il PC e i rack di I/O Serie 90 devono essere collegati a una connessione di terra comune. Normalmente questa connessione di terra comune si realizza collegando il PC e i rack degli I/O alla stessa fonte di alimentazione con lo stesso punto di riferimento a terra, ma questo deve essere verificato caso per caso.

- A. **Installare la scheda IC693PIF400** in uno slot di espansione PC/AT/ISA del PC, seguendo le istruzioni fornite dal costruttore del PC. Per assicurare la massima immunità del sistema ai disturbi, accertarsi che il contatto elettrico fra le due staffe metalliche della scheda PIF400 e lo chassis del PC sia buono (usando le viti rimosse per togliere la placchetta di protezione) e che la connessione a terra del computer sia affidabile.
 - B. Collegare l'espansione e/o i rack remoti della Serie 90-30 al connettore di espansione D a 25 pin della scheda (J1). Vedere i dettagli a pagina C-4.
 - C. Collegare il connettore del relè RUN (J2) al dispositivo esterno di controllo, facendo attenzione a non superare i valori di specifica dei contatti del relè. Vedere i dettagli a pagina C-5.
- 4. Avvio del *Program Editor* del software PC Control.**

Nel menu Avvio, selezionare PC Control Applications, Program Editor.

5. Configurazione della scheda PIF400.

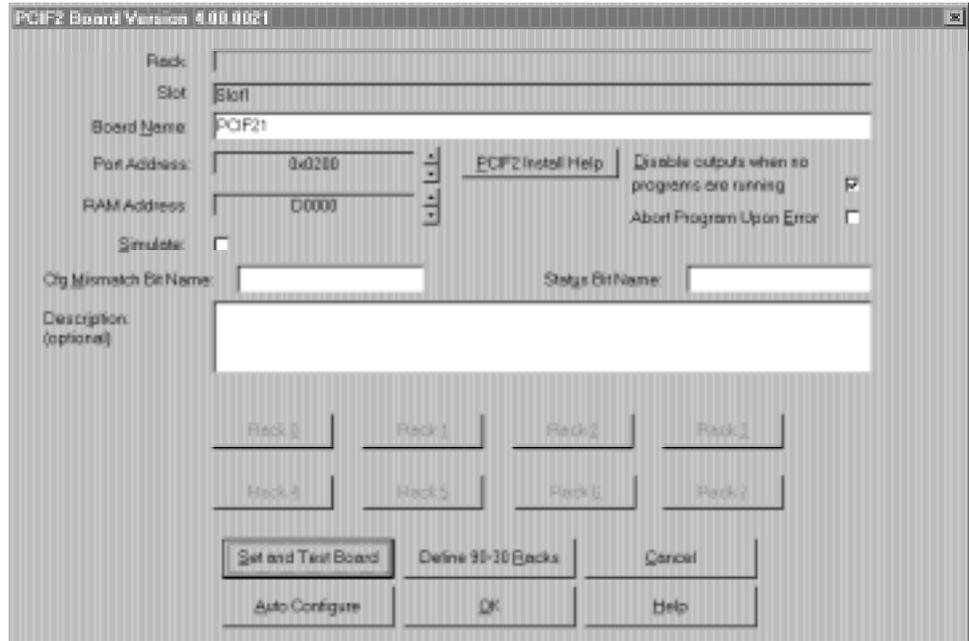
- A. **Nel menu file del Program Editor, selezionare Open Config (o New Config).** Comparirà la seguente finestra di configurazione.



B. Nell'albero visualizzato nella parte sinistra della finestra, selezionare lo slot che si desidera configurare. Comparirà la seguente finestra:



C. Nel campo Board Type, selezionare PCIF2 (per la scheda PIF400). Cliccare sul pulsante Define Board. Comparirà la finestra di dialogo PCIF2 Board.



D. Impostare i parametri di configurazione della scheda PIF400. Accertarsi che il contenuto dei campi Port Address e RAM address sia corretto. (Fare riferimento a “Controllo delle risorse del sistema” a pagina C-7).

- Il campo Port Address deve contenere lo stesso valore impostato con i microinterruttori S1.
- Il campo RAM Address deve selezionare un blocco disponibile di 32 KByte.

Per ulteriori dettagli relativi alla configurazione, consultare l'aiuto in linea del PC Control.

E. Dopo aver impostato gli indirizzi della porta e della RAM, cliccare sul pulsante Set and Test Board per verificare che l'installazione sia

corretta. (Il sistema di runtime non deve essere attivo.) Dovrebbe comparire la finestra di dialogo Board Test, contenente il messaggio “PCIF2 Board found, shared RAM access successfully found” (Scheda PCIF2 trovata, accesso alla RAM condivisa eseguito con successo). Se questo messaggio non compare, tentare di configurare un altro blocco di RAM condivisa.

6. Configurazione degli I/O della Serie 90-30.

Dalla finestra di dialogo PCIF2 Board è possibile continuare il processo di configurazione.

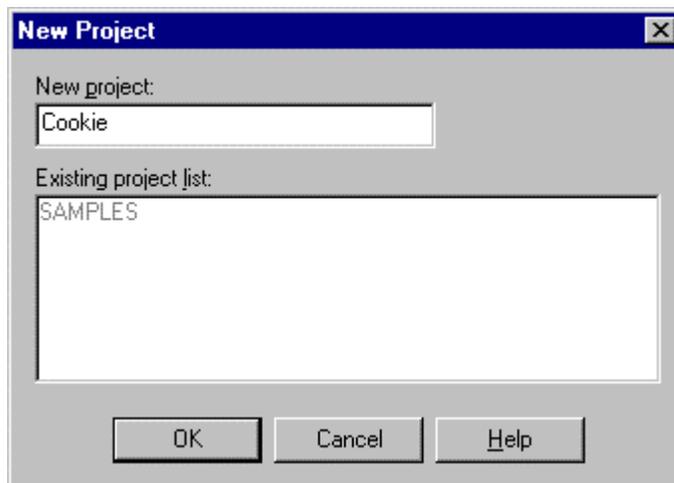
- **Configurazione automatica:** Se i rack e i moduli I/O sono installati e alimentati, e se il sistema di runtime non è attivo, è possibile usare questa prestazione. Quando si clicca sul pulsante Auto Configure, il software attiva la scheda PIF400 e legge le informazioni relative ai rack e ai moduli collegati.
- **Configurazione manuale:** Per usare questa prestazione, cliccare sul pulsante Define 90-30 Racks. Comparirà la finestra di dialogo Rack Definition.

Per ulteriori dettagli relativi alla configurazione, consultare l'aiuto in linea del PC Control.

Questa appendice accompagnerà il lettore passo passo attraverso un esempio di applicazione volto ad illustrare la creazione di un progetto, la configurazione degli I/O, la creazione di un programma applicativo mediante RLL, SFC e testo strutturato, la creazione di uno schermo di interfaccia con l'operatore, l'esecuzione e il monitoraggio dell'applicazione. Attraverso il menu Help del Program Editor è possibile accedere ad un altro esempio, che illustra la creazione di un'applicazione SFC.

Esercizio 1: Creazione di un nuovo progetto

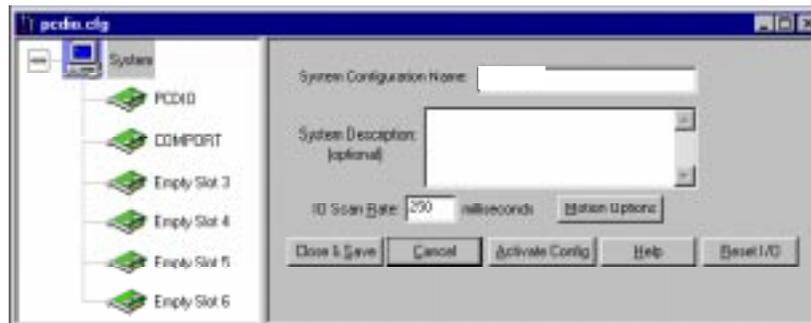
- Dal Program Editor cliccare sul menu **Project** e scegliere **New**.
- Digitare "Cookie" nel campo New project. Lo schermo dovrebbe presentarsi come quello illustrato qui sotto. Cliccare sul pulsante "OK" per creare il nuovo progetto.



Esercizio 2: Creazione di una nuova configurazione del sistema

Definizione del sistema

- Dal Program Editor selezionare il menu **File** e cliccare **NEW CONFIG**. Si aprirà una finestra di dialogo per la configurazione di un nuovo sistema, simile a quella illustrata qui sotto



- In questo esempio si useranno il nome per difetto della configurazione del sistema "Config1", e un intervallo di scansione degli I/O di 250ms.

Nota

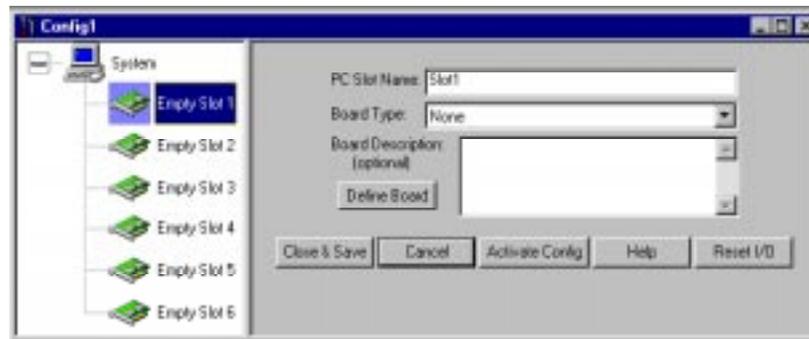
L'intervallo di scansione degli I/O determina la frequenza con cui il PC CONTROL legge e aggiorna gli I/O e risolve la logica. L'intervallo di scansione del PC CONTROL è asincrono rispetto all'intervallo di scansione degli I/O.

Definizione degli slot

Il passo successivo della configurazione degli I/O consiste nel definire le schede I/O locali, la schede I/O scanner e/o le schede motion che saranno utilizzate.

Il dialogo slot contiene varie informazioni riguardanti il tipo di scheda per le comunicazioni installata in uno specifico slot. In un PC i numeri degli slot non sono significativi, ma la numerazione degli slot può essere utile ai tecnici di manutenzione, che possono così più facilmente individuare la scheda appropriata per la connessione e la riparazione.

- Per configurare una scheda I/O cliccare sulla figura **empty slots 1** sul lato sinistro della finestra di dialogo. Dopo la selezione dello slot, il lato destro della finestra di dialogo cambia, diventando simile a quella illustrata qui sotto.



- Il nuovo dialogo include un menu a discesa (Board Type), la cui lista include tutti i driver I/O del PC CONTROL installati sul sistema.
- Nella lista del menu a discesa Board Type, selezionare il driver "**Industrial Computer Source PCDIO**". Se questo driver non appare nella lista, uscire da PC CONTROL e installarlo usando i dischi di installazione dei driver I/O.
- Dopo aver selezionato la scheda, cliccare sul pulsante **Define Board** per configurare il software per le comunicazioni con la scheda.

Definizione delle schede

La finestra di dialogo Define Board è unica per ciascuna scheda I/O. il dialogo si usa per configurare il PC CONTROL in modo che possa comunicare con la particolare scheda I/O selezionata.

- Per questo esempio, usare le informazioni per difetto che compaiono nella finestra di dialogo come si vede qui sotto. Se si dispone di un simulatore di I/O, configurarlo in maniera che lavori con queste impostazioni per difetto. Se non si dispone di un simulatore di I/O installato sul sistema, occorre simulare la scheda cliccando sulla casella di **Simulate Board**.

- Per assegnare nomi simbolici ai punti di I/O di questa scheda, cliccare sul pulsante **Def. Connectors**.

Definizione dei connettori

- Il pulsante **Def. Connectors** apre la finestra di dialogo Port Definition.
- Selezionare **Input** per la prima porta e introdurre i nomi simbolici dei primi sei punti, da "switch1" a "switch6". Questi simboli corrispondono ai sei interruttori del simulatore di I/O.
- Selezionare **Output** per la seconda porta e introdurre i nomi simbolici dei primi sette punti, da "light1" a "light6" e "buzzer" (nel settimo punto). Questi simboli corrispondono alle sei spie e al segnalatore acustico del simulatore di I/O. Nel campo "Port B Name" digitare "lights".

Nota

Per il software PC Control i nomi dei simboli fanno differenza tra lettere maiuscole e lettere minuscole. Per questo esempio, i nomi dei simboli sono tutti in lettere minuscole.

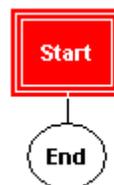
Salvataggio e attivazione della nuova configurazione

- Cliccare sul pulsante **OK** del dialogo Port Definition, poi cliccare su **Yes** quando compare la domanda "Do you wish to activate this configuration?"
- Quando compare la corrispondente domanda, cliccare su **Yes** se si desidera salvare la configurazione prima di attivarla.
- Comparirà il dialogo **Save As**, che permette di dare un nome alla configurazione. Digitare "**pcdio**" nel campo File name e cliccare su **OK**.
- Dopo l'attivazione della nuova configurazione, comparirà di nuovo la finestra di dialogo System Configuration. Cliccare su **Close and Save** per terminare la creazione della configurazione.

Esercizio 3: Creazione di un SFC

Se il Program Editor non è già stato avviato, aprire il gruppo di programmi PC Control Applications e cliccare sull'icona Program Editor per avviare questa utilità.

- Sulla barra degli strumenti dell'editor (nella parte alta dello schermo), cliccare sul pulsante corrispondente alla creazione di un nuovo file di programma. Selezionare SFC+/M nella finestra di dialogo e premere il pulsante OK per creare un nuovo diagramma funzionale sequenziale (SFC). Si aprirà una finestra contenente il nuovo programma SFC.

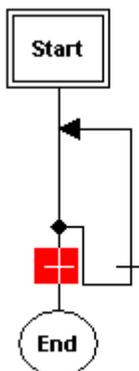


Creazione della struttura del programma

- Nella barra degli strumenti SFC, selezionare lo strumento anello.



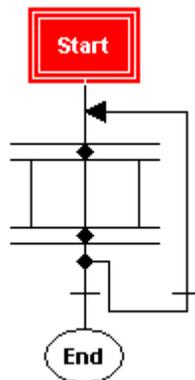
- Portare il cursore sul percorso tra il passo Start e il passo End. Premere il pulsante sinistro del mouse per inserire un elemento anello di sequenza nel programma SFC.



- Nella barra degli strumenti SFC, selezionare lo strumento divergenza simultanea.



- Portare il cursore al centro dell'anello e premere il tasto sinistro del mouse per inserire l'elemento divergenza simultanea nel programma SFC.
- Selezionare di nuovo lo strumento divergenza simultanea e aggiungere un terzo ramo alla divergenza, cliccando sulla doppia linea orizzontale superiore.



Linee guida per l'impiego di divergenze simultanee

Nel creare una divergenza simultanea, osservare le seguenti linee guida:

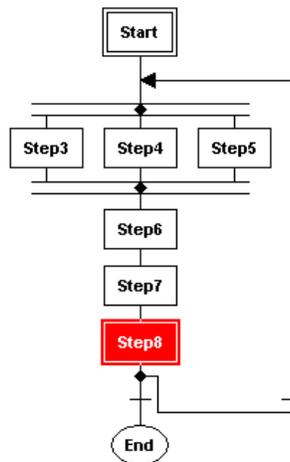
- Non fare riferimento alla stessa variabile in rami diversi di una divergenza simultanea.
- Non richiamare lo stesso SFC figlio da passi macro in rami diversi di una divergenza simultanea.
- Per non compromettere la correttezza della convergenza, non usare etichette per:
 - Saltare dall'interno all'esterno di una divergenza simultanea.
 - Saltare dall'esterno all'interno di una divergenza simultanea.
 - Saltare da un ramo all'altro della stessa divergenza simultanea..

Aggiunta dei passi

Selezionare lo strumento Passo della barra degli strumenti SFC.



Portare il cursore sul ramo sinistro della divergenza simultanea. Premere il pulsante sinistro del mouse per inserire l'elemento passo nel programma SFC. Portare il cursore sugli altri due rami e inserire un elemento passo in ciascuno di essi. Portare il cursore sul percorso di controllo successivo alla divergenza, al di sopra della connessione dell'anello di sequenza. Aggiungere tre elementi passo consecutivi in questo punto del programma SFC.

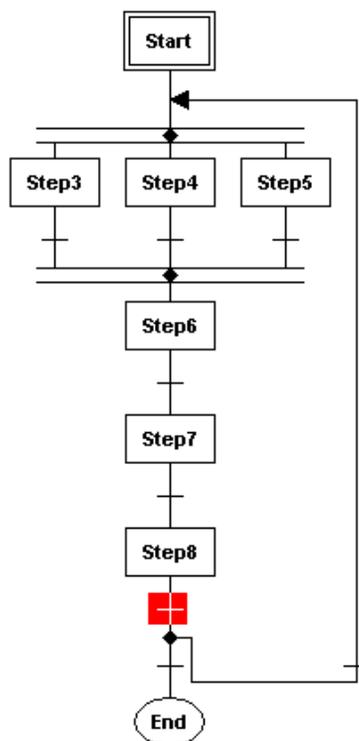


Aggiunta delle transizioni

Selezionare lo strumento transizione nella barra degli strumenti SFC.



- Portare il cursore sotto a ciascun ramo della divergenza simultanea e cliccare per inserire in ciascuno di quei punti un elemento transizione. Ripetere la stessa operazione dopo ciascuno dei tre passi successivi alla divergenza simultanea.



Salvataggio del programma applicativo



Prima di cominciare ad aggiungere i dettagli al programma SGC, è bene salvare il lavoro fatto fino a questo punto. Per salvare il file attivo con il suo nome e il suo percorso attuali, si usano il menu File/Save o il pulsante salva della barra degli strumenti dell'editor. Dato che questo è il primo salvataggio del file, il Program Editor visualizza la finestra di dialogo Save As per permettere di attribuire un nome al file. Digitare "cookie_maker" nel campo File Name e cliccare sul pulsante **Save**. La barra del titolo del programma SFC dovrebbe essere come questa:



Esercizio 4: Creazione dei simboli

Dopo aver selezionato il nuovo programma "cookie_maker.SCF" come finestra attiva, accedere al Symbol Manager premendo il corrispondente pulsante della barra degli strumenti dell'editor.



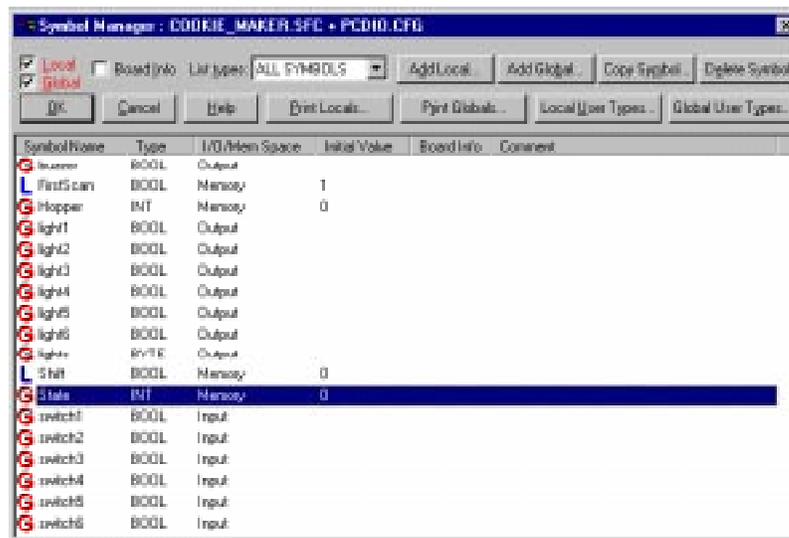
Il Symbol Manager visualizza i nomi simbolici degli I/O definiti con l'utilità di configurazione. Dato che ai simboli degli I/O è possibile accedere da tutti i programmi applicativi di un progetto, i simboli degli I/O sono contrassegnati da una "G" rossa davanti al loro nome. Questo indica che si tratta di simboli globali.

Aggiunta dei simboli locali

- Cliccare sul pulsante **Add Local**. Nella finestra di dialogo Symbol Detail digitare il nome "**FirstScan**". Accertarsi che il campo "Type" contenga **BOOL**, e introdurre "**1**" come valore iniziale. Cliccare su **OK** per finire e aggiungere il nuovo simbolo alla lista del Symbol Manager.
- Aggiungere un altro simbolo locale di nome "**Shift**", Tipo: **BOOL**, valore iniziale: **0**. Alla fine cliccare su **OK**.

Aggiunta dei simboli locali

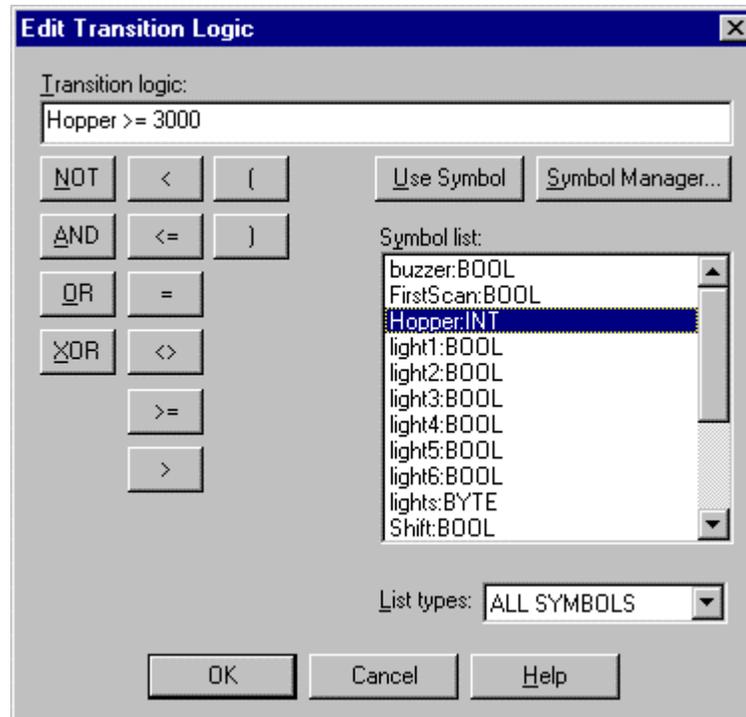
- I simboli globali si creano allo stesso modo dei simboli locali. Cliccare sul pulsante **Add Global**.
- Creare due nuovi simboli globali: "**Hopper**", di tipo **INT** e con valore iniziale **0** e "**State**", di tipo **INT** e con valore iniziale **0**.



- Cliccare su **OK** per chiudere la finestra e accettare i nuovi simboli. Poiché si tratta di simboli globali, il Program Editor chiede che cosa si vuole fare delle informazioni globali. Cliccare su **Save & Activate** per salvare i simboli globali nella configurazione del sistema e attivare la nuova configurazione.

Esercizio 5: Aggiunta della logica delle transizioni

- Fare doppio click sulla transizione del ramo sinistro della divergenza simultanea, per aprire la finestra di dialogo Edit Transition Logic.

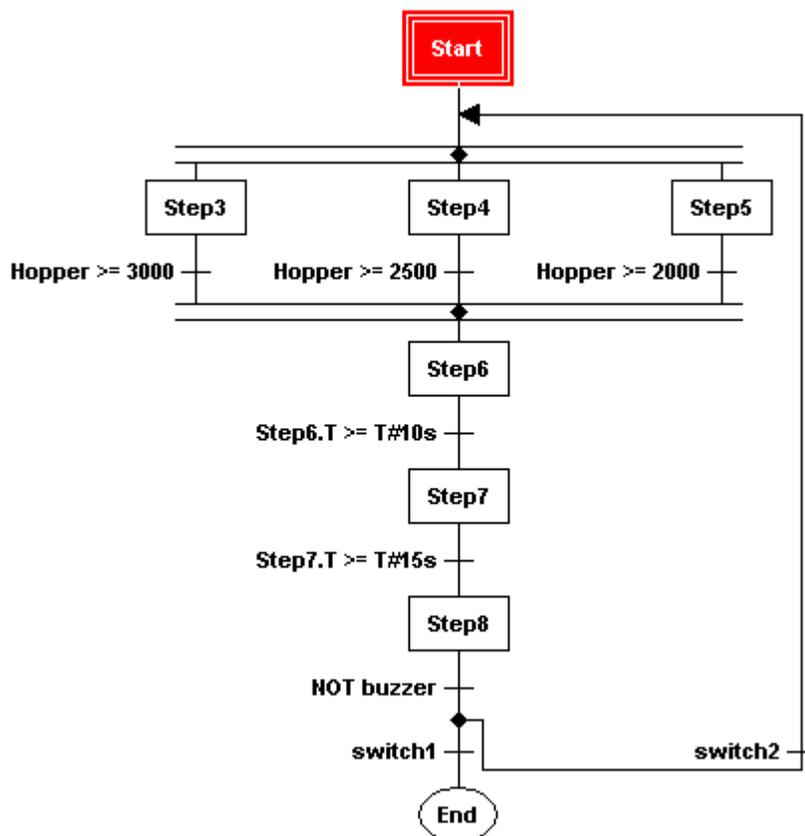


- Nella lista dei simboli, fare doppio click sul simbolo **Hopper** per copiarlo nel campo Transition Logic. Poi cliccare sul pulsante ">=" e digitare "**3000**". Cliccare su **OK** per chiudere la finestra di dialogo e accettare le modifiche.
- Ripetere questi passi per gli altri due rami della divergenza simultanea. Introdurre rispettivamente "**2500**" e "**2000**" come condizione di test per queste transizioni.
- Fare doppio click sulla transizione direttamente al di sotto del primo passo successivo alla divergenza. per la logica di questa transizione useremo una delle variabili step create automaticamente dal PC CONTROL. Nel campo Transition Logic, digitare il nome del passo che precede immediatamente questa transizione e aggiungere ".**T**" alla fine (in questo esempio: **Step6.T**). Poi cliccare sul pulsante ">=" e digitare "**T#10s**", questa è la sintassi IEC-1131 per 10 secondi. In questo modo la transizione sarà TRUE quando Step6 sarà stato attivo per 10 secondi o più.
- Ripetere questi passi sotto a **Step7**. Usare **T#15s** per rendere vera la transizione dopo 15 secondi.
- Editare la transizione successiva a **Step8**. Cliccare sul pulsante "**NOT**", poi fare doppio click sul simbolo "**buzzer**" nella lista dei simboli. Cliccare su "**OK**" per accettare queste modifiche..

Alla fine, editare le due condizioni di transizione dell'anello di sequenza.

- Fare doppio click sulla transizione connessa al passo **End**. Fare scorrere la lista dei simboli fino a visualizzare il simbolo **switch1**. Fare doppio click su questo simbolo per copiarlo nel campo Transition logic. Cliccare sul pulsante **OK** per chiudere la finestra di dialogo e accettare le modifiche.
- Fare doppio click sulla transizione del ramo di ritorno dell'anello. Fare scorrere la lista dei simboli fino a visualizzare il simbolo **switch2**. Fare doppio click su questo simbolo per copiarlo nel campo Transition logic. Cliccare sul pulsante **OK** per chiudere la finestra di dialogo e accettare le modifiche.

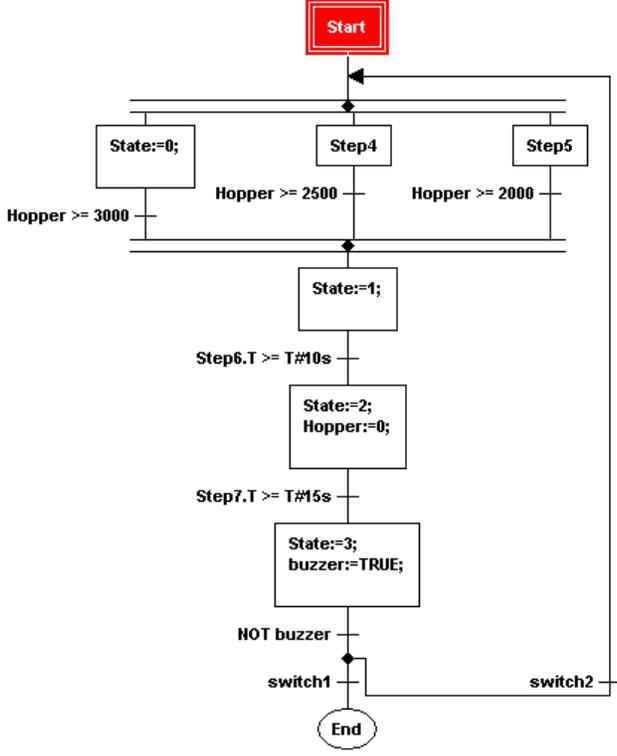
Dopo la specifica dei dettagli delle transizioni booleane, l'aspetto del programma SFC dovrebbe essere il seguente:



Esercizio 6: Introduzione dei comandi in testo strutturato

- Fare doppio click su **Step3** per aprire la finestra di dialogo **Edit Step**.
- Selezionare **Structured Text** come tipo dei comandi. Selezionare il campo Command List e digitare la seguente frase di assegnazione:
State := 0;
- Cliccare sul pulsante **OK** per chiudere il dialogo e accettare le modifiche.
- Fare doppio click su **Step6** per aprire al finestra di dialogo **Edit Step** e introdurre:
State := 1;
- Fare doppio click su **Step7** per aprire al finestra di dialogo **Edit Step** e introdurre:
State := 2;
Hopper := 0;
- Fare doppio click su **Step8** per aprire al finestra di dialogo **Edit Step** e introdurre:
State := 3;
buzzer := TRUE;
- Per vedere tutti i comandi dei passi, cliccare sul menu a discesa **View** e selezionare **All Steps/As Commands**.

Ora il programma dovrebbe presentarsi così:



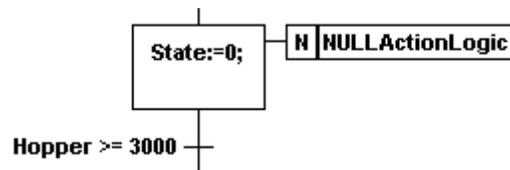
Esercizio 7: Aggiunta ed editazione dei blocchi delle azioni

Editazione dei passi 3,4 e 5

- Selezionare lo strumento azione nella barra degli strumenti SFC.



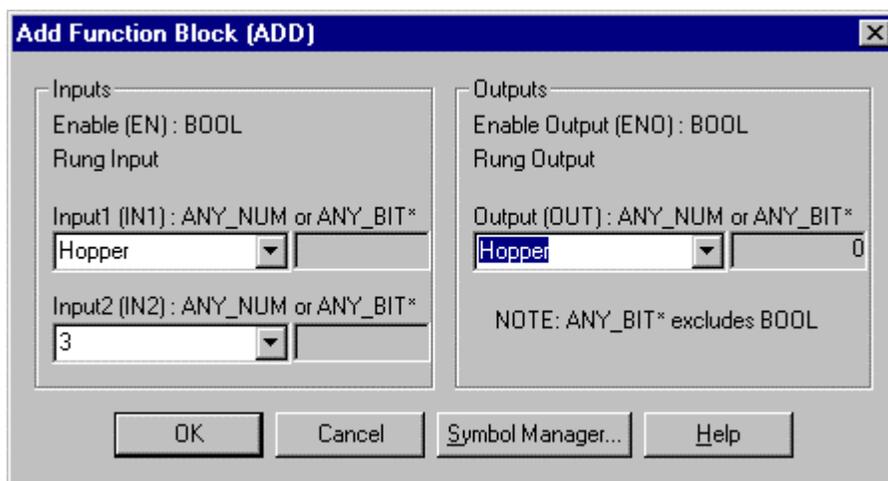
- Portare il cursore sul passo **Step3**. Cliccare per associare un'azione a questo passo. L'aspetto del passo sarà ora il seguente:



- Fare doppio click sul nome dell'azione per aprire la finestra di dialogo **Edit Action Association**. Selezionare il campo Action Name e digitare il nome dell'azione. Per questo esempio, digitare "**Flour**" e cliccare sul pulsante **OK**. Dato che questa azione non esiste ancora, compare una finestra di dialogo che chiede se la si vuole creare. Cliccare sul pulsante **OK** per creare la nuova azione. Si aprirà una finestra contenente un programma RLL vuoto per la nuova azione.
- Usando la lista a discesa di Function Block Palette, selezionare la palette **Math**.



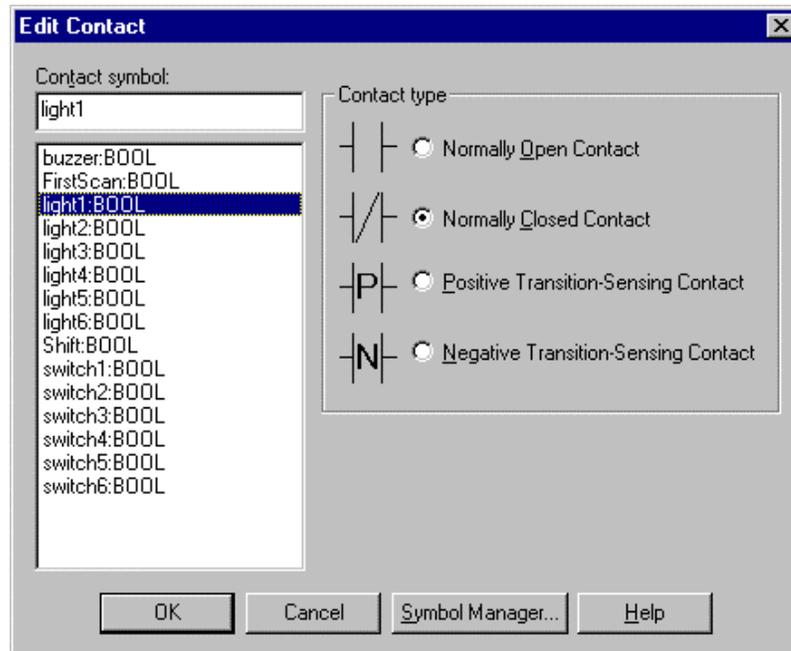
- Cliccare sul pulsante del blocco funzionale **ADD**. Il cursore cambierà forma per indicare il blocco funzionale. Posizionare il cursore sul rung della finestra RLL e cliccare con il pulsante sinistro del mouse per inserire il blocco funzionale ADD.
- Comparirà la finestra di dialogo Add Function Block (ADD). Nel campo **Input1**, digitare **Hopper**. Nel campo **Input2**, digitare **3**. Nel campo **Output**, digitare **Hopper**. Questo sommerà "3" a Hopper ad ogni scansione degli I/O, finché l'azione resterà attiva.



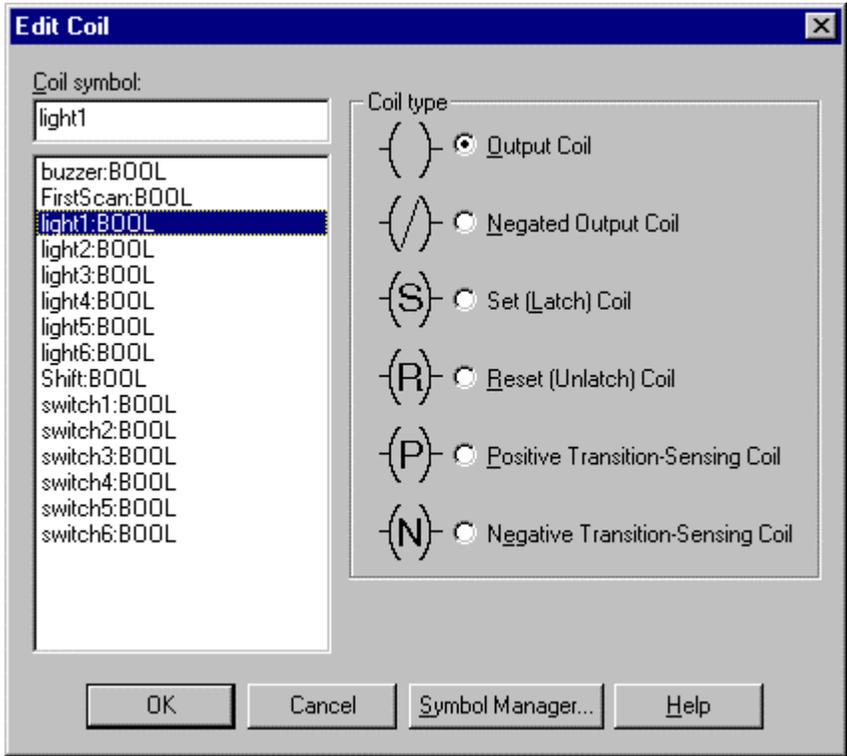
- Cliccare sul pulsante **OK** per chiudere la finestra di dialogo e accettare le modifiche.
- Cliccare sul pulsante **Close Window** per chiudere il blocco RLL dell'azione.
- Poi dare un nome alle azioni dei passi 4 e 5: **Sugar** e **Chips** rispettivamente. Inserire nell'azione Sugar un blocco funzionale **ADD** che sommi 4 ad **Hopper** e inserire nell'azione Chips un blocco funzionale **ADD** che sommi 5 ad **Hopper**.

Editazione del passo 6

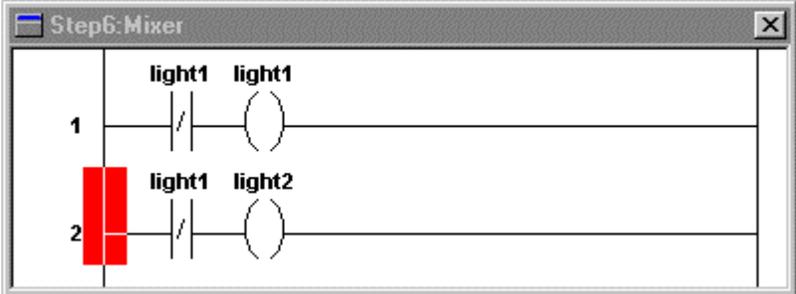
- Chiamare **Mixer** l'azione associata a **Step6**.
- Selezionare lo strumento **Rung**  nella barra degli utensili. Il cursore cambierà forma per indicare il rung. Posizionare il cursore immediatamente sotto al rung esistente e cliccare con il pulsante sinistro del mouse per inserire un nuovo rung.
- Selezionare lo strumento **Contatto**  nella barra degli strumenti RLL. Il cursore cambierà forma per indicare il Contatto. Posizionare il cursore sul primo rung e cliccare con il pulsante sinistro del mouse per inserire un contatto.
- Comparirà la finestra di dialogo Edit Contact. Nella lista dei simboli selezionare **light1**. Per il tipo di contatto, selezionare **Normally Closed Contact**. Cliccare sul pulsante **OK** per chiudere la finestra di dialogo e accettare le modifiche.



- Selezionare lo strumento **Bobina**  nella barra degli strumenti RLL.. Il cursore cambierà forma per indicare la bobina. Posizionare il cursore sul primo rung, a destra del contatto, e cliccare con il pulsante sinistro del mouse per inserire la bobina.
- Comparirà la finestra di dialogo Edit Coil. Selezionare **light1** nella lista dei simboli. Per il tipo di bobina, selezionare **Output Coil**. Cliccare sul pulsante **OK** per chiudere la finestra di dialogo e accettare le modifiche.

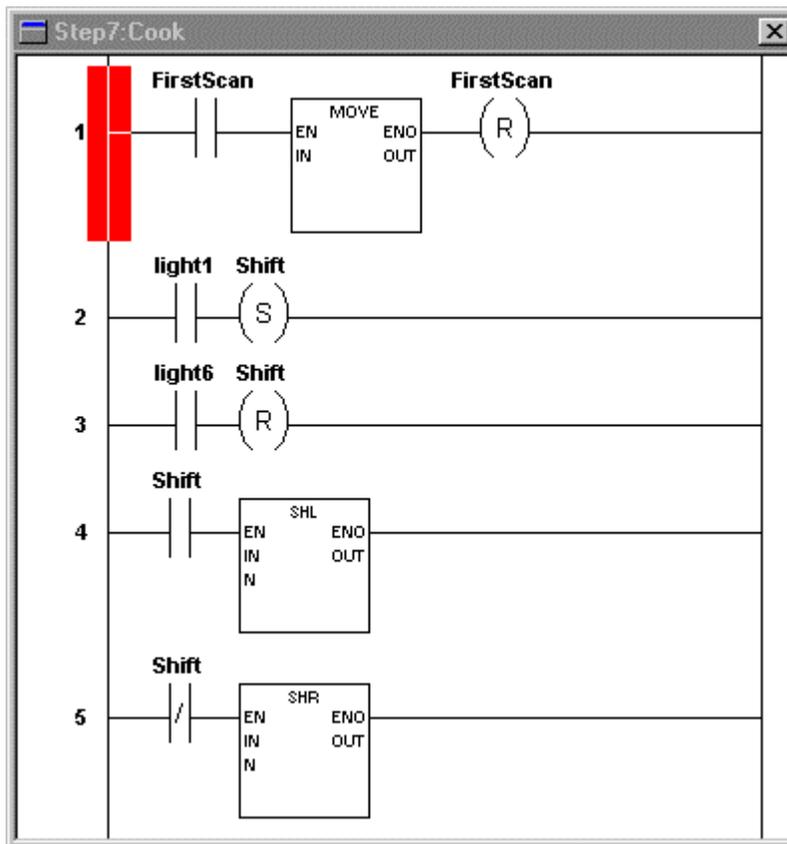


- Aggiungere al secondo rung un altro contatto **light1**, normalmente chiuso, e una bobina di uscita per **light2**.
- Alla fine il programma RLL dovrebbe presentarsi così:

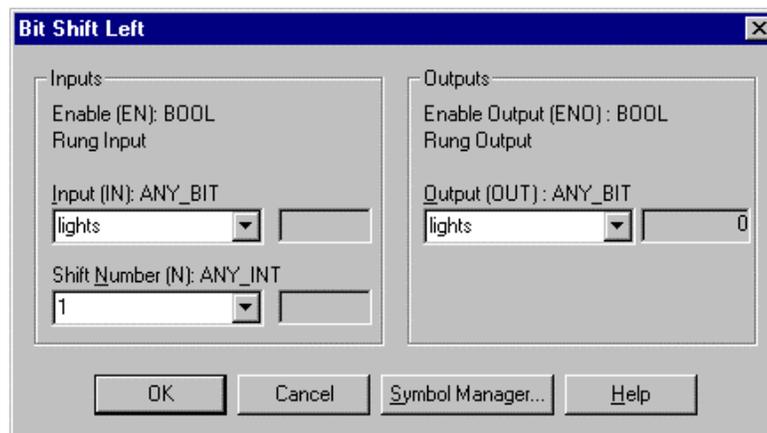


Passo 7

- Chiamare **Cook** l'azione associata al passo **Step7**.
- Aggiungere i rung, i contatti, le bobine e i blocchi funzionali opportuni per ottenere una figura simile a questa:



- Il blocco funzionale **MOVE**, sul rung 1, muove un 1 nel simbolo **lights**.
- **Bit Shift Left(BSL)** e **Bit Shift Right(BSR)** sui rung 4 e 5 (i blocchi funzionali di questo tipo si trovano nella paletta dei blocchi funzionali **Bitwise**) shiftano il simbolo **lights** di **1**. Ciascuna delle finestre di dialogo Bit Shift dovrebbe essere configurata come segue:

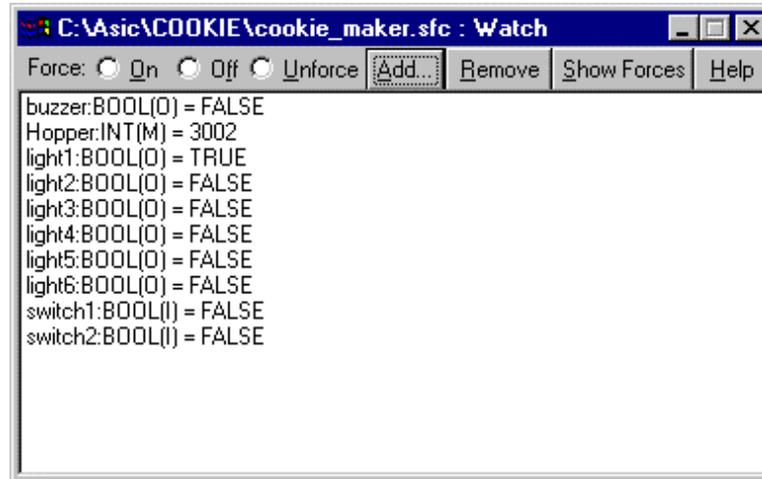


Esercizio 8: Esecuzione del programma dell'esempio

- Per eseguire cookie_maker.SFC, usare il comando di menu Execute\Run. Se i sottosistemi di runtime di PC CONTROL non sono attivi, compare la richiesta di avviarli. Quando il programma comincia a girare, la sua visualizzazione viene evidenziata. I passi e le transizioni attivi sono evidenziati in verde.
- Aprire la **Watch Window** cliccando sull'icona Watch Window nella barra degli strumenti del Program Editor.



- Cliccare sul pulsante **Add**. Evidenziare i seguenti nomi simbolici cliccando su di essi con il pulsante sinistro del mouse. Cliccare sul pulsante **OK** per chiudere la finestra di dialogo Add e aggiungere i simboli.. La Watch Window (finestra di controllo) dovrebbe essere simile alla seguente:



- Forzare i simboli così come è necessario per verificare il funzionamento del programma.

Esercizio 9: Creazione dello schermo di interfaccia con l'operatore

Per creare un nuovo schermo operatore, selezionare il comando Edit/New Screen nel menu a discesa. Comparirà la finestra di dialogo Select New Screen Name. Digitare "**CookieMaker**" e premere il pulsante **OK**. Verrà creato uno schermo operatore in bianco con il nome visualizzato nella barra del titolo.



Quando si crea un file Interfaccia Operatore, viene generato e selezionato come "schermo iniziale" uno schermo standard del PC Control. Dopo aver creato un nuovo schermo, lo schermo standard può essere cancellato, con il comando Edit/Delete Screen del menu a discesa e può essere selezionato un nuovo schermo iniziale.

Aggiunta dei controlli per l'operatore

Indicatori

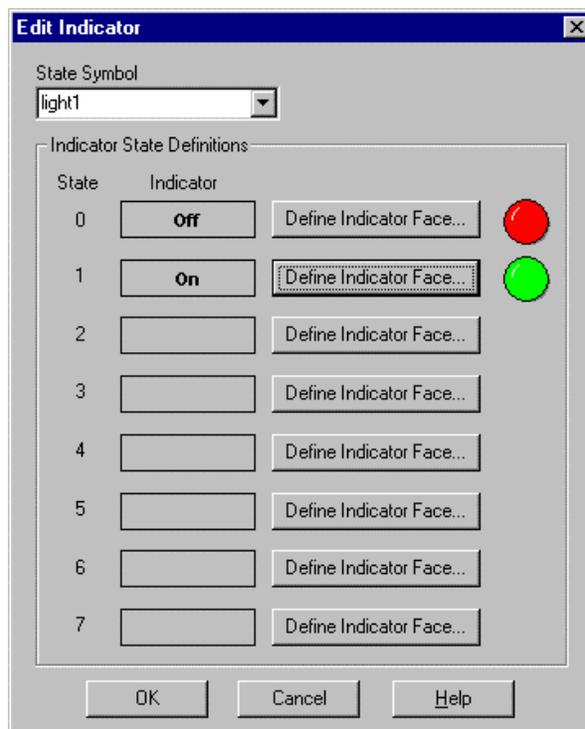
Aggiungere alcuni indicatori per poter osservare lo stato della macchina per la produzione di biscotti (cookie). Ne serviranno sei per riflettere lo stato corrente delle spie e uno per sapere in quale programma si è.



- Selezionare lo strumento indicatore nella barra degli strumenti dell'interfaccia con l'operatore. Il cursore cambierà forma per indicare

l'indicatore. Portare il cursore nella posizione desiderata dello schermo di interfaccia con l'operatore e cliccare con il pulsante sinistro del mouse per inserire l'indicatore.

- Fare doppio click sul nuovo indicatore per aprire la finestra di **Edit Indicator**.
- Nella campo **State Symbol**, digitare, o selezionare nella lista a discesa il simbolo "**light1**". Questo farà dipendere l'aspetto dell'indicatore dal valore corrente di light1.
- Cliccare sul pulsante **Define Indicator Face** per lo stato 0, allo scopo di aprire la finestra di dialogo Define Indicator.
- Nel campo "**Text:**" digitare: "**Off**". Poi cliccare sul pulsante "**Select Light Color**" e selezionare l'icona della spia rossa.
- Cliccare sul pulsante **OK** per accettare le modifiche e chiudere la finestra di dialogo.
- Ripetere il processo per lo stato 1, inserendo però il testo "**On**" e selezionando la spia verde. Alla fine, la finestra di dialogo dovrebbe presentarsi così:



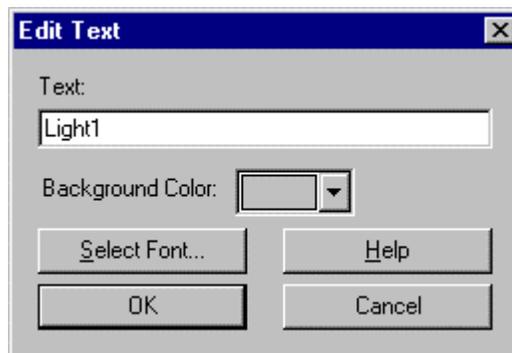
- Usando i comandi **Copy/Paste** (copia/incolla) creare un totale di sei indicatori. Assegnare allo **State Symbol** dei cinque nuovi indicatori i simboli **light2** a **light6**.
- Usando la barra degli strumenti, inserire ancora un indicatore. Assegnare allo **State Symbol** del nuovo indicatore il simbolo **State**.
- Per gli stati da 0 a 3 assegnare il testo:
stato 0 **Filling Hopper** (riempimento tramoggia)
stato 1 **Mixing Batter** (miscelatura impasto)
stato 2 **Cooking Cookies** (cottura biscotti)
stato 3 **Done** (fatto)

Visualizzazione di testi

Per non confondere tra loro gli indicatori, è opportuno aggiungere un testo sopra a ciascuno di essi per identificarli.



- Selezionare lo strumento testo nella barra degli strumenti dell'interfaccia operatore. Il cursore cambierà forma per indicare il testo. Portare il cursore sopra all'indicatore della spia 1 e cliccare con il pulsante sinistro del mouse per inserire una casella di testo.
- Fare doppio click sulla casella di testo per aprire la finestra di dialogo **Edit Text**.
- Inserire **Light1** nel campo **Text**.
- Cliccare sul pulsante **OK** per chiudere la finestra di dialogo e accettare le modifiche.
- Creare altre cinque caselle di testo per identificare gli indicatori delle spie da 2 a 6.



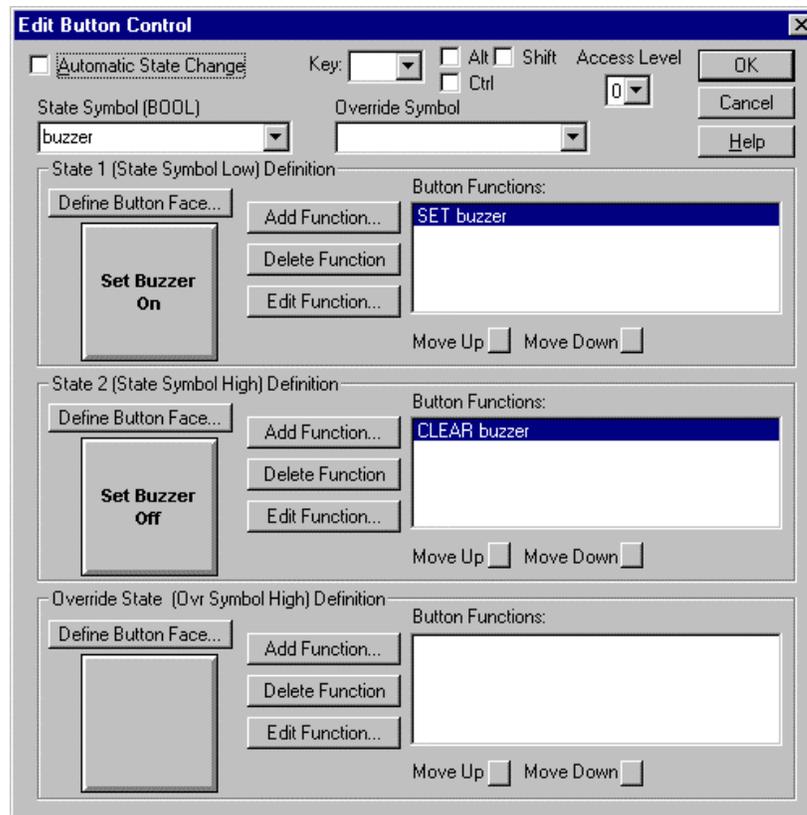
Pulsante multistato (Click button)

Per controllare il cicalino di fine lavoro della macchina per la produzione di biscotti, è necessario aggiungere un pulsante multistato.



- Selezionare il pulsante multistato nella barra degli strumenti dell'interfaccia operatore. Il cursore cambierà forma per indicare il pulsante. Portare il cursore nella posizione desiderata dello schermo di interfaccia e cliccare con il pulsante sinistro del mouse per inserire il pulsante.
- Fare doppio click sul nuovo pulsante per aprire la finestra di dialogo **Edit Button Control**.
- Nel campo **State Symbol**, introdurre il nome simbolico **buzzer**.
- Nella porzione **State 1 Definition** della finestra di dialogo, cliccare sul pulsante **Add Function**.
- Nella lista a discesa che compare, selezionare la funzione **Set**.
- Nella lista a discesa che compare successivamente, selezionare il simbolo **buzzer**.
- Nella porzione **State 2 Definition** della finestra di dialogo, cliccare sul pulsante **Add Function**.
- Nella lista a discesa che compare, selezionare la funzione **Clear**.
- Nella lista a discesa che compare successivamente, selezionare il simbolo **buzzer**.

- Mediante i pulsanti **Define Button Face** per gli stati 1 e 2, editare il testo del pulsante in modo che sia, rispettivamente, "**Set Buzzer On**" e "**Set Buzzer Off**". Alla fine, la finestra di dialogo dovrebbe presentarsi così:



- Cliccare sul pulsante **OK** per chiudere la finestra di dialogo e accettare le modifiche.

Se non si dispone di un simulatore di I/O, sarà necessario aggiungere altri due pulsanti, per controllare i simboli **switch1** e **switch2**.

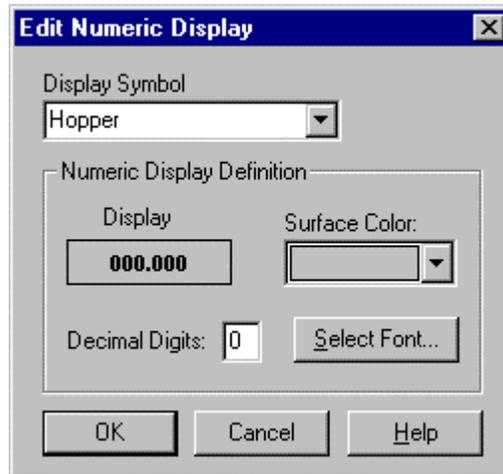
Indicatore numerico

Allo scopo di monitorare il peso della tramoggia, aggiungere un elemento per la visualizzazione di valori numerici.



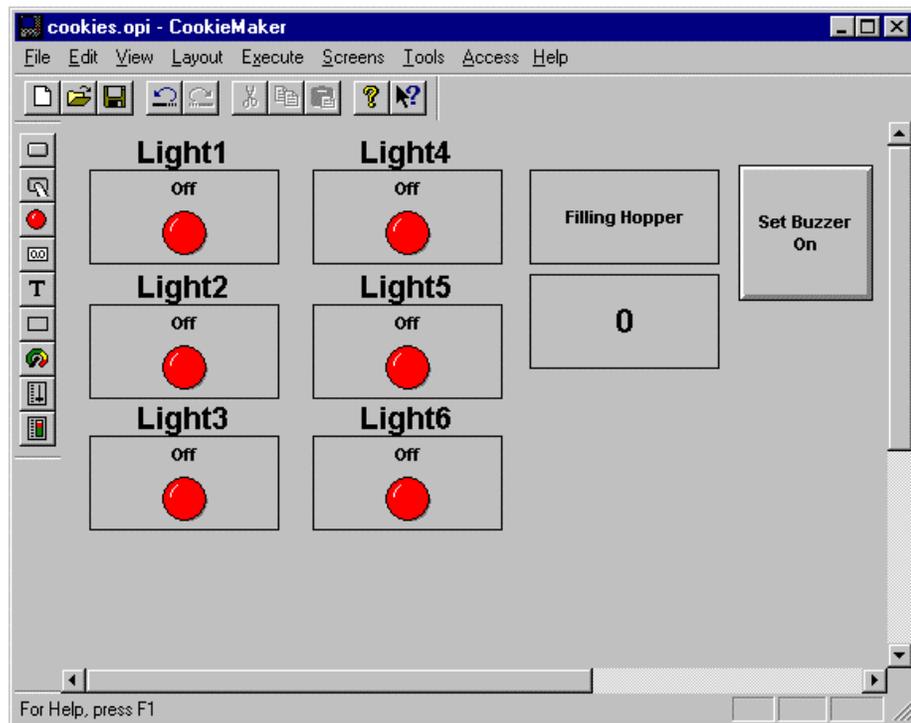
- Selezionare l'indicatore numerico nella barra degli strumenti dell'interfaccia operatore. Il cursore cambierà forma per indicare l'indicatore numerico. Portare il cursore nella posizione desiderata sullo schermo dell'operatore e cliccare con il pulsante sinistro del mouse per inserire il nuovo elemento.
- Fare doppio click sul nuovo indicatore numerico per aprire la finestra di dialogo **Edit Numeric Display**.
- Nel campo **Display Symbol**, inserire il nome simbolico **Hopper**.
- Specificare **0** nel campo **Decimal Digits**.

- Cliccare sul pulsante **OK** per chiudere la finestra di dialogo e accettare le modifiche.



Attivazione dello schermo operatore

Alla fine, l'aspetto dello schermo operatore dovrebbe essere il seguente:



Per vedere lo schermo in azione, accertarsi che "Cookie_maker.SFC" sia in esecuzione e selezionare il comando di menu **Execute\Activate Screens**.

Il linguaggio di programmazione Motion Control è un linguaggio testuale conforme allo standard RS-274D per la programmazione di funzioni di controllo dei movimenti. Il linguaggio permette di programmare movimenti bi- e tridimensionali mediante parametri quali:

- posizioni in un sistema di coordinate
- velocità di avanzamento
- movimenti da una posizione ad un'altra con accelerazione e decelerazione controllate.

Un programma Motion Control consiste in una serie di comandi di una lettera, seguiti dai valori numerici (parametri) di questi comandi. I comandi sono organizzati in singole righe di testo, dette blocchi. I blocchi formano unità di esecuzione, eseguite in sequenza. L'esecuzione del programma si arresta ad ogni blocco finché tutte le funzioni programmate in quel blocco non sono completate; poi viene eseguito il blocco successivo.

E' possibile aggiungere istruzioni del linguaggio Motion Control a un SFC e includere frasi in Testo strutturato in un codice Motion Control. Per ulteriori informazioni, fare riferimento al paragrafo "Uso delle frasi Motion Control", a pagina E-4.

Configurazione per Motion Control

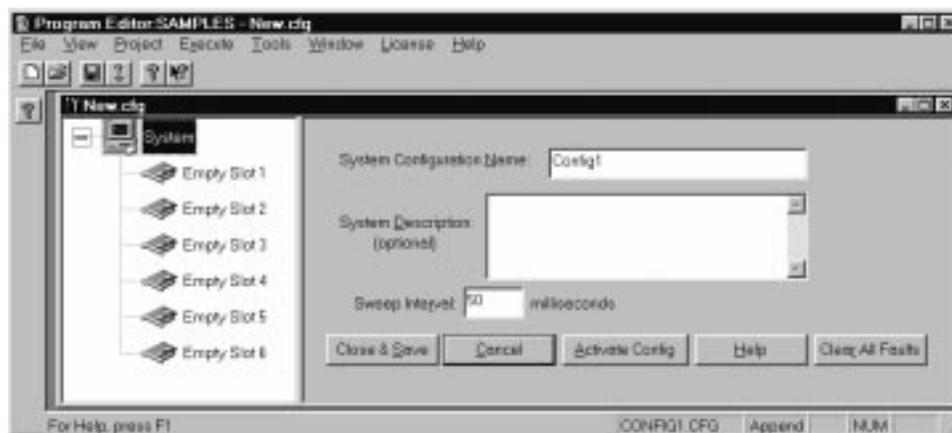
PC Control può essere usato con le seguenti schede di interfaccia, che supportano Motion Control:

- Delta Tau PMAC-PC
- Delta Tau PMAC Direct
- Delta Tau PMAC2
- Motion Engineering PCDSP
- Motion Options Compumotor

Nota

In un sistema è possibile installare una sola scheda Motion.

- A. **Dal menu** Program Editor File del PC Control, cliccare su Open Config (o New Config). Comparirà la seguente finestra di configurazione:

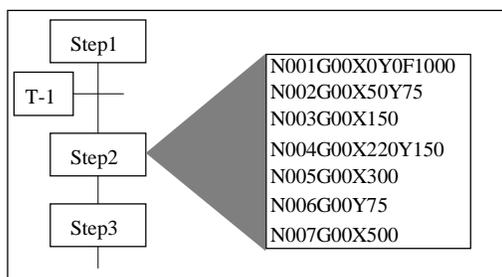


- B. Nell'albero visualizzato nella parte sinistra della finestra, selezionare o slot che si vuole configurare.
- C. Nel campo Board Type, selezionare la scheda Motion installata. Comparirà il dialogo di configurazione della scheda selezionata.
- D. Cliccare sul pulsante Define Board. Per i dettagli necessari al completamento della configurazione, fare riferimento all'aiuto in linea della scheda Motion Control.

Programmazione Motion Control

Aggiunta di codice Motion Control a un SFC

Nella creazione del codice applicativo per un passo SFC, è possibile scegliere di usare codice Motion Control, come si vede qui sotto. Quando viene eseguito l'SFC, il codice Motion Control incorporato in un passo viene eseguito quando il passo diventa attivo. Il codice Motion Control può essere scritto direttamente nel passo, oppure è possibile collegare al passo un file contenente il codice Motion Control durante la configurazione del passo. Le informazioni contenute nel file devono avere lo stesso formato di quelle inserite direttamente nel passo.



Per le informazioni relative ai miglioramenti apportati alle specifiche RS-274D dal software del sistema di comando, fare riferimento a “Miglioramenti del software PC CONTROL allo standard RS-274D” a pagina E-3.

Miglioramenti del software PC CONTROL allo standard RS-274D

Il software del sistema di comando apporta i seguenti miglioramenti alle specifiche RS-274D.

- E' possibile includere assegnazioni ed espressioni in Testo strutturato nel codice Motion Control. Fare riferimento a “Inclusione di Testo strutturato nel codice Motion Control ” a pagina E-27.
- E' possibile usare le azioni dei passi per sincronizzare le operazioni di I/O con l'esecuzione del codice Motion Control. E' anche possibile usare qualificatori di movimento per sincronizzare la logica delle azioni con i movimenti.
- Il software del sistema di comando fornisce simboli motion mappati sugli I/O. Ad esempio, asse.CMDPOS è un simbolo di tipo REAL che contiene la posizione comandata di un asse.
- Ai codici M del linguaggio Motion Control sono associati indicatori M, messi on o off in base ai codici M utilizzati. Gli indicatori M possono essere usati da altri programmi, determinando l'inizio di altre azioni dopo che nell'esecuzione del programma viene incontrato un codice M.
- E' possibile migliorare il proprio controllo sul flusso del programma usando i comandi WHILE e IF-GOTO insieme ai comandi Motion Control.
- E' possibile richiamare una macro (un sottoprogramma) mediante il comando preparatorio G65.

Uso delle frasi Motion Control

Uso dei comandi Motion Control

Il software del sistema di comando converte i comandi RS-274D in comandi appropriati generati specificamente per scheda assi selezionata.

I comandi Motion devono essere strutturati in uno specifico formato (Fare riferimento a “Formato di un blocco Motion Control” a pagina E-5). I comandi Motion sono:

Comando	Funzione
A	Comando di movimento rotativo X in unità predefinite.
B	Comando di movimento rotativo Y in unità predefinite.
C	Comando di movimento rotativo Z in unità predefinite.
D	Funzione utensile per la selezione del correttore.
F	Funzione avanzamento o comando di velocità in unità predefinite.
G	Funzione preparatoria. Vedere “Uso dei codici G” a pagina E-6.
I	Parametro dell’interpolazione o passo filettatura parallela all’asse X.
J	Parametro dell’interpolazione o passo filettatura parallela all’asse Y.
K	Parametro dell’interpolazione o passo filettatura parallela all’asse Z.
M	Funzione miscellanea. Vedere “Uso dei codici M” a pagina E-8.
N	Numero blocco (facoltativo).
S	Funzione velocità mandrino.
T	Funzione utensile. Campo da 01 a 32,
X	Comando di movimento lineare X in unità predefinite.
Y	Comando di movimento lineare Y in unità predefinite.
Z	Comando di movimento lineare Z in unità predefinite.

Formato di un blocco Motion Control

Il formato di un blocco Motion Control block (una singola riga di codice) è il seguente:

- N** Se si usa il numero di sequenza del blocco, esso deve essere il primo comando del blocco. Per il sistema di comando è facoltativo.
- G** La o le funzioni preparatorie G devono seguire N.
- X** Le parole dimensionali lineari seguono le G. Specificare l'asse X per primo.
- Y** Le parole dimensionali lineari seguono le G. Specificare l'asse Y per secondo.
- Z** Le parole dimensionali lineari seguono le G. Specificare l'asse Z per terzo.
- A** Le parole dimensionali rotative seguono le G. Specificare l'asse X per primo.
- B** Le parole dimensionali rotative seguono le G. Specificare l'asse Y per secondo.
- C** Le parole dimensionali rotative seguono le G. Specificare l'asse Z per terzo.
- I** Le parole dell'interpolazione seguono le parole dimensionali. Specificare l'asse X per primo.
- J** Le parole dell'interpolazione seguono le parole dimensionali. Specificare l'asse Y per secondo.
- K** Le parole dell'interpolazione seguono le parole dimensionali. Specificare l'asse Y per secondo.
- D** La selezione del correttore utensile deve seguire K
- F** Se è specificata una velocità di avanzamento valida per più di un asse, F deve seguire l'ultima parola dimensionale (e di interpolazione) alla quale si applica.
- T** La funzione di selezione utensile segue S.
- M** Le eventuali funzioni miscelanee devono essere specificate per ultime in un blocco, immediatamente prima del carattere di fine blocco.

Fine blocco Indicare la fine del blocco con i caratteri ritorno carrello/avanzamento riga. Tra l'ultimo comando e il carattere di fine blocco non devono esserci spazi in bianco, tabulazioni o altri caratteri.

Esempi di blocchi Motion Control

Esempio 1

N009G01X-3.0Y-7.0Z+1.0F95

N009 Nono blocco del programma.

G01 Posiziona l'utensile sul punto successivo lungo un percorso rettilineo.

Specifica la posizione dell'asse X.

Specifica la posizione dell'asse Y.

Specifica la posizione dell'asse Z.

F95 Specifica la velocità di avanzamento (95 unità).

Esempio 2

N011G02X+0.5Y+1.0I0.75J0.0

N011 Undicesimo blocco del programma.

G02 Posiziona l'utensile con un movimento circolare.

Specifica la posizione dell'asse X.

Specifica la posizione dell'asse Y.

I0.75 Parametro X dell'arco.

J0.0 Parametro Y dell'arco.

Esempio 3

N003G70X+1.3Y-7.0Z+2.1M08

N003 Terzo blocco del programma.

G70 Programmazione in pollici.

Specifica la posizione dell'asse X.

Specifica la posizione dell'asse Y.

Specifica la posizione dell'asse Z.

M08 Attiva il refrigerante numero 1.

Uso dei codici G

Il software del sistema di comando supporta i codici G definiti dallo standard RS-274D G per controllare molte operazioni di movimento. I codici G supportati dipendono dal driver. Non tutti i codici G sono supportati da tutti i driver. Nella tabella dei codici G, un puntino (•) nella colonna di un driver indica che il codice G è supportato da quel driver.

I codici G disponibili sono i seguenti:

Codice	Descrizione	Note	PMAC	PCDSP	Compu- motor
G00	Movimento rapido punto a punto	Gli assi possono arrivare sul punto finale non contemporaneamente.	•	•	•
G01	Movimento lineare coordinato	La velocità del movimento è controllata e tutti gli assi si muovono in modo coordinato.	•	•	
G02	Movimento circolare orario	Gli assi eseguono un movimento circolare coordinato in senso orario, visto dall'alto.	•	Solo i primi due assi	
G03	Movimento circolare antiorario	Gli assi eseguono un movimento circolare coordinato in senso antiorario, visto dall'alto.	•	Solo i primi due assi	
G04	Sosta	L'esecuzione del programma è sospesa per il periodo di tempo specificato. La durata si specifica con una parola F nello stesso blocco. Es.: G04F2.5 = ritardo di 2.5 secondi.	•	•	•
G17	Selezione del piano Xp - Yp	Xp: asse X o asse parallelo a X	•		
G18	Selezione del piano Zp - Xp	Yp: asse Y o asse parallelo a Y.	•		
G19	Selezione del piano Yp - Zp	Zp: asse Z o asse parallelo a Z	•		
G40	Cancella compensazione raggio		•		
G41	Compensazione diametro utensile (sinistra)	Compensazione tridimensionale.	•		
G42	Compensazione diametro utensile (destra)	Compensazione tridimensionale.	•		
G43	Compensazione lunghezza utensile (Più)		•		
G44	Compensazione lunghezza utensile (Meno)		•		
G45	Incremento correzione utensile		•		
G46	Decremento correzione utensile		•		
G47	Doppio incremento correzione utensile		•		
G48	Doppio decremento correzione utensile		•		
G52	Origine coordinate locali	Dipende dall'applicazione	•		
G53	Movimento nel sistema di coordinate della macchina		•	•	
G54	Sistema di coordinate del pezzo 1		•		
G55	Sistema di coordinate del pezzo 2		•		
G56	Sistema di coordinate del pezzo 3		•		
G57	Sistema di coordinate del pezzo 4		•		
G58	Sistema di coordinate del pezzo 5		•		
G59	Sistema di coordinate del pezzo 6		•		
G61	Modo arresto esatto		•		
G64	Modo taglio		•		
G65	Richiamo macro		•	•	•

Codice	Descrizione	Note	PMAC	PCDSP	Compu- motor
G66	Funzione speciale	Dipende dal driver	•	•	
G70	Modo programmazione in pollici	Definito durante la configurazione, non disponibile al runtime.	Non in runtime	Non in runtime	
G71	Modo programmazione metrica	Definito durante la configurazione, non disponibile al runtime.	Non in Runtime	Non in Runtime	
G90	Modo programmazione assoluta		•	•	
G91	Modo programmazione incrementale		•	•	
G92	Preselezione della posizione	I valori di posizione degli assi diventano quelli specificati nel blocco	•	•	
G93	Modo programmazione avanzamento con l'inverso del tempo		•		
G94	Modo avanzamento al minuto		•	Solo avanzamento al minuto	

Uso dei codici M

I codici M sono operazioni definite dall'utente supportate in RS-274D e SFC. I codici M validi vanno da M0 a M99. I codici M sono eseguiti in RS-274D e sono segnalati alla logica di supporto nei programmi SFC o RLL. A ciascun codice M corrisponde un simbolo del sistema di comando, da Mflag0 a Mflag98. I codici M pari mettono a off il simbolo Mflag corrispondente, i codici M dispari lo mettono a on. Ad esempio:

Comando RS-274D	Risultato
M10	Mflag10 off (FALSE)
M11	Mflag11 on (TRUE)
M96	Mflag96 off (FALSE)
M97	Mflag97 on (TRUE)

Note:

I codici M che hanno una funzione speciale devono essere in un blocco da soli. Questi includono: M00, M01, M03, M04, M05, M58, e M59.

I codici M programmati in un blocco con movimento, sono eseguiti prima del movimento.

Codici M predefiniti

Il software del sistema di comando predefinisce vari codici M usati per operazioni interne; questi codici M non possono essere usati per l'on/off degli Mflag .

Codice	Descrizione
M0	Arresto programma
M1	Arresto opzionale programma
M2	Fine programma
M3	Mandrino positivo
M4	Mandrino negativo
M5	Arresto mandrino
M30	Fine programma e riavvolgimento
M99	Fine funzione macro

Codici M con attesa

Il software del sistema di comando ha la possibilità di mettersi in attesa quando incontra un codice M RS-274D. Terminato il trattamento del codice M, il software del sistema di comando può mettere il programma applicativo in attesa che la logica predisposta dall'utente comunichi al sistema che l'operazione comandata dal codice M è stata completata e che l'esecuzione del programma può riprendere dal successivo blocco RS-274D. Per abilitare questa prestazione, occorre selezionare la casella di controllo Wait on All M Codes nella pagina Motion Options Configuration.

I simboli del sistema di comando elencati qui sotto riguardano i codici M con attesa:

				Resettato (R) da/ Settato (S) da	
Nome della variabile	Tipo	Uso	Letture(L) e Scrittura (S)	S/W sistema	Utente
GruppoDiAssi.MWAIT	BOOL	Attesa completamento codice M	L	S/R	---
GruppoDiAssi.MCODE	INT	Valore del codice M	L	S/R	---
GruppoDiAssi.MCONT	BOOL	Codice M completato – riprendere l'esecuzione	L/S	R	S

Definizione dei simboli Mflag

Se si desidera che il software del sistema di comando generi 46 simboli globali di tipo BOOL, selezionare la casella dell'opzione "Define M Flag Symbols". I simboli sono generati quando questa opzione è selezionata e la configurazione viene salvata e attivata. Questi simboli compaiono nel Symbol Manager e i loro nomi vanno da Mflag6 a Mflag 96, contando per numeri pari.

Uso dei codici M con attesa

Se si desidera che il programma applicativo sospenda l'esecuzione del codice RS-274D finché la logica predisposta per un codice M non indica che l'azione corrispondente è stata completata, selezionare la casella dell'opzione "Wait on All M codes". Quando viene eseguito un codice M, il programma RS-274D è sospeso e la variabile booleana GruppoDiAssi.MWAIT è TRUE, finché il programma applicativo non assegna TRUE alla variabile booleana GruppoDiAssi.MCONT; questo provoca la ripartenza del programma RS-274D e il reset delle variabili GruppoDiAssi.MWAIT e GruppoDiAssi.MCONT da parte del software di sistema. E' responsabilità del programmatore applicativo predisporre la logica che setta la variabile GruppoDiAssi.MCONT dopo l'attivazione di GruppoDiAssi.MWAIT.

La variabile GruppoDiAssi.MCODE è un intero contenente il valore del codice M. "GruppoDiAssi" sta per il nome del gruppo di assi al quale è associato il codice M. I nomi dei gruppi di assi si definiscono nel file della configurazione del progetto.

Codici M escludibili

Il software del sistema di controllo supporta questa funzione speciale dei codici M, come è descritto nella specifica RS-274D. Se viene selezionata la casella dell'opzione "Do Not Process M Codes", il programma applicativo ignora i seguenti codici M:

- M3 (mandrino positivo)
- M4 (mandrino negativo)
- M5 (arresto mandrino)

Simboli Motion Control predefiniti

Il software del sistema di controllo fornisce vari simboli predefiniti che possono essere usati per comandare e controllare alcune operazioni Motion Control dell'applicazione. La connessione dei simboli con le funzioni appropriate si esegue durante la configurazione degli I/O.

I simboli predefiniti sono raggruppati come segue:

- “Simboli delle uscite asse”
- “Simboli delle uscite dei gruppi di assi” a pagina E-12
- “Simboli delle uscite mandrino” a pagina E-13
- “Simboli degli ingressi asse” a pagina E-13
- “Simboli degli ingressi dei gruppi di assi” a pagina E-16

Nota: Non tutti i simboli predefiniti relativi agli assi sono supportati da tutti i driver. Per individuare i simboli supportati, ricorrere all'aiuto del driver.

Simboli delle uscite asse

Simbolo	Tipo di dato	Funzione
asse.ACTPOS	reale	Contiene la posizione reale dell'asse.
asse.ACTVEL	reale	Contiene la velocità reale dell'asse.
asse.CMDPOS	reale	Contiene la posizione comandata dell'asse.
asse.TPOS	reale	Contiene la posizione finale programmata dell'asse.
asse.AXSFE	reale	Contiene l'errore di inseguimento dell'asse.
asse.A	booleano	Indica se l'asse è abilitato.
asse.IP	booleano	Indica se l'asse è in posizione.
asse.MC	booleano	Indica se il movimento dell'asse è completato.
asse.ESTPO	booleano	Indica se è attivato l'ESTOP dell'asse.
asse.RELPOS	reale	Contiene la posizione relativa dell'asse.
asse.HOME	booleano	Indica se l'asse è nella posizione HOME (zero).
asse.STATUS	reale	Indica lo stato dell'asse – la definizione dei bit dipende dal driver. Riferirsi all'aiuto del driver.
asse.CMDSPD	reale	Contiene la velocità comandata dell'asse.

Simboli delle uscite dei gruppi di assi

Simbolo	Tipo di dato	Funzione
GruppoDiAssi.ESTPO	booleano	Indica se è attivo l'arresto in emergenza (ESTOP).
GruppoDiAssi.TOOL	intero	Contiene il correttore utensile attivo.
GruppoDiAssi.INTPL	intero	Contiene il modo di interpolazione attivo per il gruppo. 0=punto a punto 1=lineare 2=circolare orario 3=circolare antiorario.
GruppoDiAssi.CIR	booleano	Indica se è attiva l'interpolazione circolare (oraria o antioraria).
GruppoDiAssi.PTP	booleano	Indica se è attivo il modo punto a punto.
GruppoDiAssi.LIN	booleano	Indica se è attiva l'interpolazione lineare.
GruppoDiAssi.CIRCW	booleano	Indica se è attiva l'interpolazione circolare oraria.
GruppoDiAssi.CIRCCW	booleano	Indica se è attiva l'interpolazione circolare antioraria.
GruppoDiAssi.DWL	booleano	Indica se è attiva la sosta.
GruppoDiAssi.DWLTIM	intero	Contiene il tempo si sosta in millisecondi..
GruppoDiAssi.ENG	booleano	Indica se sono attive le unità inglesi (pollici).
GruppoDiAssi.METRIC	booleano	Indica se sono attive le unità metriche.
GruppoDiAssi.ABSDIM	booleano	Indica se è attiva la programmazione assoluta.
GruppoDiAssi.INCDIM	booleano	Indica se è attiva la programmazione incrementale.
GruppoDiAssi.WAIT	booleano	Indica che il programma è sospeso per il cambio utensile..
GruppoDiAssi.CONT	booleano	Indica se il programma è ripartito dopo il cambio utensile..
GruppoDiAssi.MWAIT	booleano	Indica l'attesa del completamento della funzione M.
GruppoDiAssi.MCODE	intero	Contiene il valore del codice M.
GruppoDiAssi.PLANE	intero	Indica il piano attivo per l'interpolazione circolare: 0=FIXOFF[0] (G54) 1=FIXOFF[1] (G55) 2=FIXOFF[2] (G56) 3=FIXOFF[3] (G57) 4=FIXOFF[40] (G58) 5=FIXOFF[5] (G59)
GruppoDiAssi.CUTMOD	intero	Modo di lavorazione attivo. 0=Modo taglio (G64). 1=Modo arresto esatto (G61).
GruppoDiAssi.FEDMOD	intero	Modo attivo della velocità di avanzamento. 0=Avanzamento al minuto (G94). 1=Inverso del tempo (G93).

Simboli delle uscite mandrino

Simbolo	Tipo di dato	Funzione
S.WAIT	booleano	Indica che il programma è sospeso in seguito a un nuovo comando mandrino.
S.CONT	booleano	Indica che il programma è ripartito dopo un comando mandrino.
S.DIR	booleano	Indica il senso di rotazione programmato del mandrino. 0=positivo (orario/M3) 1=negativo (antiorario/M4)
S.CMDSPD	reale	Contiene la velocità programmata del mandrino: la velocità specificata dall'ultimo comando S.

Lo stato del gruppo di assi per i comandi Motion Control è scritto nel gruppo contenente l'asse specificato. Se non è specificato alcun asse, lo stato del gruppo assi è scritto nel gruppo di assi 1 (axisGroup#1), che è il gruppo di assi assunto per difetto.

Simboli degli ingressi asse

Simbolo	Tipo di dato	Funzione
asse.JP	booleano	Causa un movimento manuale in direzione +. "Uso dei simboli degli ingressi asse .JM e .JP" a pagina E-15.
asse.JM	booleano	Causa un movimento manuale in direzione -. Vedere "Uso dei simboli degli ingressi asse .JM e .JP" a pagina E-15.
asse.HLD	booleano	Causa l'arresto del movimento asse finché il bit non è resettato.
asse.STP	booleano	Causa l'arresto del movimento asse.
asse.JSPD	reale	Specifica la velocità dell'asse in manuale.
asse.JINCR	reale	Specifica l'incremento per il movimento manuale dell'asse
asse.JTYPE	intero	Seleziona il tipo di movimento manuale dell'asse (0=azzeramento, 1=continuo, 2=incrementale).
asse.SOVR	reale	Specifica la regolazione della velocità di avanzamento dell'asse (100.0=100%).
asse.TOOLOFF[x]	reale	Un array di correzioni utensile per l'asse. x = 0-9. Vedere "Uso dei simboli degli ingressi asse .TOOLOFF" a pagina E-14. (Non per MEL.)
asse.FIXOFF[x]	reale	Un array di compensazioni dell'attrezzo di fissaggio per l'asse. x = 0-5. Vedere "Uso dei simboli degli ingressi asse .FIXOFF" a pagina E-14. (Non per MEL.)
asse.RAPID	reale	Velocità G00.

Uso dei simboli degli ingressi asse .FIXOFF

I comandi preparatori da G-55 a G-59 selezionano le compensazioni dell'attrezzo di fissaggio del pezzo. G-54 cancella le compensazioni. I simboli asse.FIXOFF contengono i valori di compensazione usati da questi comandi. La relazione fra i comandi e i simboli contenenti la loro rispettiva compensazione è indicata nella sottostante tabella.

Comando G	Simbolo
G-54	asse.FIXOFF[0]
G-55	asse.FIXOFF[1]
G-56	asse.FIXOFF[2]
G-57	asse.FIXOFF[3]
G-58	asse.FIXOFF[4]
G-59	asse.FIXOFF[5]

Assegnare i valori appropriati agli elementi dell'array nel codice in testo strutturato di un passo precedente al passo Motion Control contenente i comandi G.

Uso dei simboli degli ingressi asse .TOOLOFF

I comandi preparatori da G-45 a G-48 selezionano le correzioni utensile. Questi quattro comandi richiedono che nello stesso blocco sia specificato il comando D. Il comando D specifica l'elemento di asse.TOOLOFF[] da usare. I simboli asse.TOOLOFF contengono i valori di correzione usati dai comandi D. La relazione fra i comandi D e i simboli contenenti il loro rispettivo valore è la seguente:

Comando D	Simbolo
D-0	asse.TOOLOFF[0]
D-1	asse.TOOLOFF[1]
.	.
D-8	asse.TOOLOFF[8]
D-9	asse.TOOLOFF[9]

Assegnare i valori delle correzioni utensile agli elementi dell'array nel codice in testo strutturato di un passo precedente al passo Motion Control contenente i comandi G e D.

Uso dei simboli degli ingressi asse .JM e .JP

I simboli asse.JP e asse.JM sono simboli booleani che attivano il movimento manuale dell'asse in direzione più e in direzione meno, rispettivamente. Questi simboli hanno i seguenti requisiti:

- I simboli asse.JP e asse.JM non sono abilitati finché non viene assegnato un valore appropriato ai simboli asse.JSPD e asse.JTYPE, che specificano la velocità e il tipo del movimento manuale.
- L'asse cui si riferiscono i simboli asse.JP e asse.JM deve essere in un gruppo che non contiene altri assi, oppure, se il gruppo contiene altri assi, questi non possono essere sotto il controllo di un altro programma Motion Control quando viene dato un comando di movimento manuale per l'asse corrispondente ai simboli asse.JP e asse.JM.

Per azzerare un asse

1. Porre .JTYPE = 0.
2. Porre .JSPD = velocità di azzeramento.
3. Attivare .JP o .JM.
4. Attendere finché non è .HOME = TRUE per più di un secondo.
5. Resettare .JP o .JM.

Simboli degli ingressi dei gruppi di assi

Simbolo	Tipo di dato	Funzione
GruppoDiAssi.ESTP	booleano	Attiva l'arresto in emergenza (ESTOP). Il simbolo asse.ESTP è un booleano che causa l'arresto di tutti gli assi definiti. Notare che questo include tutti gli assi di tutti i gruppi, indipendentemente dal gruppo di assi al quale si riferisce il simbolo.
GruppoDiAssi.HLD	booleano	Causa l'arresto del gruppo di assi fino al reset del bit.
GruppoDiAssi.STP	booleano	Causa l'arresto del gruppo di assi.
GruppoDiAssi.SOVR	reale	Regolazione della velocità per il gruppo di assi (100.0=100%).
GruppoDiAssi.MCONT	booleano	Il trattamento del codice M è terminato.
GruppoDiAssi.TOOLRAD[x]	reale	Valore usato per la compensazione raggio utensile (G41 e G42). x = 0-31. Per ulteriori informazioni, vedere "Uso dei simboli di ingresso per gruppi di assi .TOOLRAD" a pagina E-16
GruppoDiAssi.TOOLLEN[x]	reale	Valore usato per la compensazione lunghezza utensile (G43 e G44). x = 0-31. Per ulteriori informazioni, vedere "Uso dei simboli di ingresso per gruppi di assi .TOOLLEN" a pagina E-17.
GruppoDiAssi.RESTP	booleano	Resetta ESTOP finché non viene a sua volta resettato. NOTA: questo ingresso deve essere resettato subito dopo essere stato attivato, se viene lasciato in on, il sistema non risponderà agli altri ingressi.
GruppoDiAssi.OPSTP	booleano	Attiva l'arresto opzionale (M01).
GruppoDiAssi.TSTRUN	booleano	Attiva il modo prova a vuoto.

Uso dei simboli di ingresso per gruppi di assi .TOOLRAD

I comandi preparatori G-41 e G-42 applicano la compensazione raggio utensile al percorso programmato. Entrambi i comandi richiedono la specifica del comando D nello stesso blocco. Il comando D specifica l'elemento di GruppoDiAssi.TOOLRAD[] da selezionare. Il simbolo GruppoDiAssi.TOOLRAD contiene i valori di compensazione. La relazione fra i comandi D e i simboli contenenti i valori di compensazione è data nella tabella che segue. I comandi preparatori sono codici G. Per ulteriori informazioni, fare riferimento a "Uso dei codici G", a pagina E-7.

Relazione fra i comandi D e i simboli

Comando D	Simbolo
D-0	GruppoDiAssi.TOOLRAD[0]
D-1	GruppoDiAssi.TOOLRAD[1]
D-2	GruppoDiAssi.TOOLRAD[2]
.	.
.	.
.	.
D-29	GruppoDiAssi.TOOLRAD[29]
D-30	GruppoDiAssi.TOOLRAD[30]
D-31	GruppoDiAssi.TOOLRAD[31]

Assegnare i valori appropriati agli elementi dell'array nel codice in testo strutturato di un passo precedente al passo Motion Control contenente i comandi G e D.

Il codice G40 cancella la compensazione.

Uso dei simboli di ingresso per gruppi di assi .TOOLLEN

I comandi G43H- e G44H- applicano la compensazione lunghezza utensile al percorso programmato. H specifica l'elemento di GruppoDiAssi.TOOLLEN da selezionare. Il simbolo GruppoDiAssi.TOOLLEN contiene i valori di compensazione. La relazione fra H e i simboli contenenti i valori di compensazione è data nella tabella che segue. I comandi preparatori sono codici G. Per ulteriori informazioni, fare riferimento a "Uso dei codici G", a pagina E-7.

Comando H	Simbolo
H0	GruppoDiAssi.TOOLLEN[0]
H1	GruppoDiAssi.TOOLLEN[1]
H2	GruppoDiAssi.TOOLLEN[2]
.	.
.	.
.	.
H29	GruppoDiAssi.TOOLLEN[29]
H30	GruppoDiAssi.TOOLLEN[30]
H31	GruppoDiAssi.TOOLLEN[31]

Assegnare i valori appropriati agli elementi dell'array nel codice in testo strutturato di un passo precedente al passo Motion Control contenente i comandi G e H.

Il codice G40 cancella la compensazione.

Configurazione delle opzioni di Motion Control

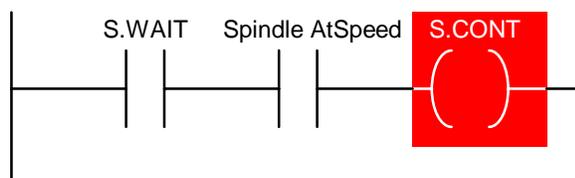
Il sistema di controllo contiene opzioni che sono di aiuto nella configurazione di Motion Control:

- “Sospensione del movimento al comando mandrino” a pagina E-18
- “Sospensione dei movimenti al cambio utensile” a pagina E-19
- “Definizione dei simboli Mflag “ a pagina E-10
- “Codici M con attesa ” a pagina E-10
- “ Codici M escludibili” a pagina E-10

Sospensione del movimento al comando mandrino

Se si desidera che un programma RS-274D sospenda l'esecuzione dei movimenti quando viene eseguito un comando del mandrino (S, M3, M4 or M5) selezionare la casella “Suspend on Spindle Commands”. Questa opzione permette al programma applicativo di sospendere l'esecuzione del programma RS-274D finché la logica dell'applicazione non indica che il comando mandrino è stato completato. Poi può essere eseguito il resto del programma. Quando viene eseguito un comando mandrino, il programma Motion Control è sospeso e il simbolo booleano S.WAIT è attivato finché il programma applicativo non attiva S.CONT; questo causerà la ripartenza del programma Motion Control e il reset di S.WAIT e S.CONT da parte del sistema. E' responsabilità del programmatore dell'applicazione predisporre la logica che attiva S.CONT dopo l'attivazione di S.WAIT e il completamento del comando mandrino.

Esempio:



Sospensione dei movimenti al cambio utensile

Se si desidera che un programma RS-274D sospenda l'esecuzione dei movimenti quando viene eseguito un comando utensile (T), selezionare la casella "Suspend on Tool Changes". Questa opzione permette al programma applicativo di sospendere l'esecuzione del programma RS-274D finché la logica dell'applicazione non indica che il comando utensile è stato completato. Poi può essere eseguito il resto del programma. Quando viene eseguito un comando utensile, il programma Motion Control è sospeso e il simbolo booleano GruppoDiAssi.WAIT è attivato finché il programma applicativo non attiva GruppoDiAssi.CONT; questo causerà la ripartenza del programma Motion Control e il reset di GruppoDiAssi.WAIT e GruppoDiAssi.CONT da parte del sistema. E' responsabilità del programmatore dell'applicazione predisporre la logica che attiva GruppoDiAssi.CONT dopo l'attivazione di GruppoDiAssi.WAIT e il completamento del comando utensile.

GruppoDiAssi è il nome del gruppo di assi al quale è associato l'utensile. I nomi dei gruppi di assi sono definiti nel file di configurazione del progetto.

Controllo del flusso del programma in applicazioni Motion Control

Come miglioramento delle specifiche RS-274D, il software del sistema di comando permette di usare i comandi WHILE e IF-GOTO insieme ad altri comandi Motion Control per ottenere un migliore controllo del flusso del programma. Fare riferimento ai paragrafi:

- "Uso del comando WHILE" a pagina E-19
- "Uso del comando IF-GOTO" a pagina E-21

Uso del comando WHILE

I comandi all'interno di un ciclo WHILE sono eseguiti ripetutamente, finché il risultato di una <espressione booleana> non è FALSE. Poi, il flusso del programma continua con il comando RS-274D che segue immediatamente il comando END_WHILE.

Formato del comando WHILE

```
WHILE (<espressione booleana >) DO
    <comandi RS-274>
END_WHILE
```

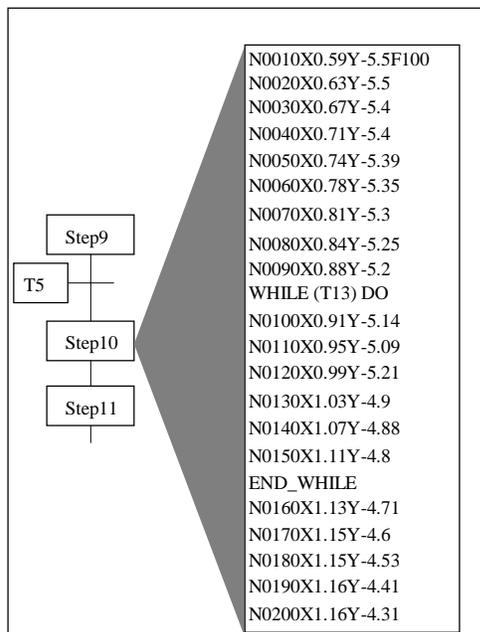
Dove:

< espressione booleana > : espressione in testo strutturato che fornisce come risultato un valore booleano;

< comandi RS-274 > : una serie di normali comandi RS-274D.

Esempio di comando WHILE

La sottostante figura illustra un esempio di comando WHILE.



Quando il passo step10 è attivo e T13 è TRUE, l'esecuzione del programma Motion Control entra nel ciclo WHILE. I blocchi da N0100 a N0150 sono eseguiti continuamente, finché T13 non diventa FALSE. A quel punto, viene eseguito il blocco N0160.

Uso del richiamo macro G65 con Motion Control

Come miglioramento delle specifiche RS-274D, il software del sistema di controllo permette di richiamare delle macro, talvolta dette sottoprogrammi, usando il comando G65 in un programma RS-274D. La programmazione di un sottoprogramma G65 consiste in due semplici passi:

1. Predisporre la macro; vedere “Scrittura di una macro.”
2. Richiamare la macro per l’esecuzione; vedere “Richiamo di macroistruzioni.”

Scrittura di una macro

Una macro può essere scritta in un punto qualsiasi del programma. Il comando G65, che richiama la macro specificata, può comparire prima o dopo la macro.

La sintassi delle macroistruzioni è la seguente:

```
O< x >  
< comandi RS-274 >  
M99
```

Dove:

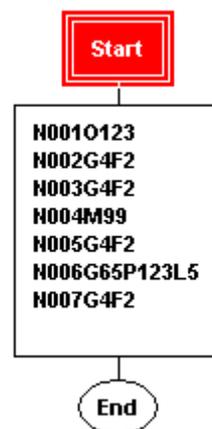
< x > è un intero che identifica la macro

< comandi RS-274 > è il codice della macro, che è costituito da normali comandi RS-274D

M99 è un blocco M che dichiara la fine della macro

I Comandi RS-274D compresi fra l’identificatore $O< x >$ e il blocco M99 sono eseguiti solo se vengono richiamati da un comando G65. In altre parole, l’esecuzione salta il codice compreso fra il blocco con la parola O e il blocco M99, a meno che questo codice non venga richiamato esplicitamente con un comando G65.

Questo è un esempio di programma SFC, tratto dal software del sistema di comando, che dimostra la funzionalità G65.



Richiamo di macroistruzioni

Una macro si richiama con un blocco G65. La sintassi del comando è la seguente:

G65P< etichetta >L< loop >

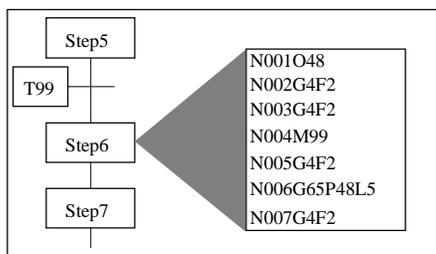
Dove:

< etichetta > è l'intero che identifica la macro. Questo intero deve coincidere con l'identificatore specificato nella parola O.

< loop > è un intero che indica il numero di ripetizioni della macro.

Il parametro < etichetta > deve coincidere con l'identificatore specificato nella parola O che dà inizio alla macro. I parametri < etichetta > e < loop > sono ambedue obbligatori. Ad esempio se la macro 22 deve essere chiamata una volta, la sintassi è: G65P22L1.

Qui sotto viene esemplificato il richiamo di macroistruzioni.



Quando è attivo il passo step6, il flusso del programma salta i primi quattro blocchi, che costituiscono la macro, e l'esecuzione inizia dal blocco N005. Poi viene eseguito il blocco N006, che richiama la macro (blocchi da N001 a N004) per un totale di cinque volte. Dopo la quinta esecuzione della macro, il flusso del programma continua con il blocco N007.

Monitoraggio ed esecuzione di programmi applicativi Motion Control

Questa sezione spiega come usare il pannello jog, il pannello di stato monoasse o multiasse, e come monitorare il plottaggio asse.

Nota: Questi pannelli visualizzano solo i primi tre caratteri di un nome asse e sono limitati ai primi otto assi.

Uso del pannello Jog

Il pannello Jog si usa per azzerare gli assi configurati e per muoverli manualmente. Questo pannello contiene vari pulsanti che permettono di selezionare:

- l'asse
- il tipo di movimento manuale (azzeramento, continuo, incrementale)
- la velocità del movimento manuale
- l'entità di un incremento

L'asse selezionato è visualizzato nella parte superiore del pannello Jog e viene mosso o azzerato (portato nella posizione HOME) nella direzione desiderata premendo e tenendo premuto il pulsante jog+ o il pulsante jog-. Se il pulsante (jog+ o jog-) viene rilasciato, il movimento viene cancellato. Per il movimento manuale incrementale, l'asse si ferma alla fine di un incremento. Per eseguire un altro incremento, rilasciare e premere di nuovo il pulsante jog. Il pannello Jog viene predisposto automaticamente per gli assi configurati.

Il pannello Jog visualizza:

- la posizione assoluta
- la posizione comandata
- l'errore di inseguimento e la velocità dell'asse selezionato

Uno specifico indicatore segnala un eventuale errore dell'asse selezionato. Quando si verifica un errore asse, l'indicatore diventa rosso e il corrispondente pulsante è abilitato. Se non ci sono errori asse, l'indicatore è verde.

Per visualizzare i dettagli dell'errore premere il pulsante di errore. Quando l'errore viene cancellato l'indicatore ritorna ad essere verde.

Plottaggio dell'asse

Il comando Axis Plot del menu View permette di visualizzare e plottare rispetto al tempo i dati asse. Questo comando apre una finestra di stato nella quale vengono plottate rispetto al tempo le informazioni di stato selezionate dell'asse prescelto. L'asse selezionato e i dati di stato sono visualizzati nella parte superiore della finestra di stato.

Per cambiare l'asse o i dati di stato selezionati, premere il pulsante relativo all'asse o al dato di stato desiderati. Usare i pulsanti Zoom In, Zoom Out, Shift Up, Shift Down, Speed Up e Slow Down per posizionare il plottaggio dei dati come si desidera. E' possibile attivare contemporaneamente più copie dello plottaggio dello stato dell'asse.

Pannello di stato monoasse

Lo stato degli assi può essere visualizzato asse per asse selezionando View/Single/Axis Status. Il pannello Single Axis Status visualizza lo stato di un asse. Questo pannello viene predisposto automaticamente per tutti gli assi configurati.

Per visualizzare lo stato di un asse, selezionare l'asse premendo il pulsante del pannello ad esso corrispondente. Il nome dell'asse selezionato compare nella parte superiore del pannello.

Per l'asse selezionato sono visualizzati i seguenti dati:

- posizione assoluta
- posizione comandata
- errore d'inseguimento
- velocità

Uno specifico indicatore segnala un eventuale errore dell'asse selezionato. Quando si verifica un errore asse, l'indicatore diventa rosso e il corrispondente pulsante è abilitato. Se non ci sono errori asse, l'indicatore è verde.

Per visualizzare i dettagli dell'errore premere il pulsante di errore. Quando l'errore viene cancellato l'indicatore ritorna ad essere verde.

E' possibile attivare più copie di questo pannello per visualizzare lo stato di più di un asse contemporaneamente.

Pannello di stato multiasse

Per visualizzare contemporaneamente lo stato di più assi, selezionare View\Multi-Axis Status. Il pannello Multi-Axis Status visualizza contemporaneamente lo stato specificato di tutti gli assi configurati. Il dato di stato da visualizzare è selezionabile fra i seguenti:

- posizione assoluta (POS)
- posizione comandata (CMD)
- errore d'inseguimento (FE)
- velocità (VEL)

Selezionare il dato desiderato premendo il corrispondente pulsante del pannello. Il nome del dato selezionato compare nella parte superiore del pannello.

Per ciascun asse, uno specifico indicatore segnala un eventuale errore di quell'asse. Se non ci sono errori, l'indicatore è verde. Quando si verifica un errore asse, l'indicatore dell'asse in errore diventa rosso e il corrispondente pulsante è abilitato.

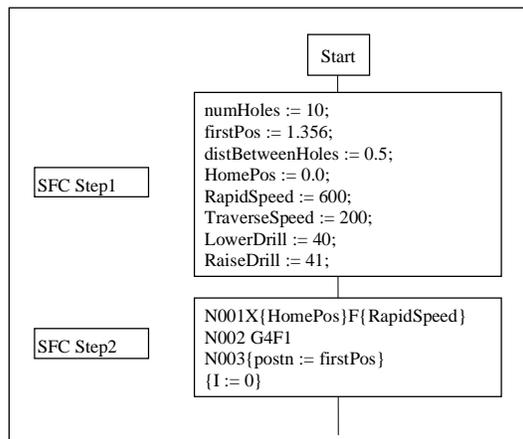
Per visualizzare i dettagli dell'errore premere il pulsante di errore. Quando l'errore viene cancellato l'indicatore ritorna ad essere verde.

Il pannello di stato multiasse è predisposto automaticamente per tutti gli assi configurati.

E' possibile attivare contemporaneamente più copie del pannello di stato multiasse.

Inclusione di Testo strutturato nel codice Motion Control

Come miglioramento delle specifiche IEC-1131-3, il software del sistema di comando permette di usare assegnazioni ed espressioni del linguaggio ST nel codice Motion Control. Ad esempio, nella sottostante figura, il blocco 1 del passo Step2 comanda un movimento dell'asse X alla posizione *HomePos* alla velocità *RapidSpeed*. I valori dei simboli *HomePos* e *RapidSpeed* sono determinati da assegnazioni ST programmate nel passo Step1. I blocchi 3 e 4 del passo Step2 presentano esempi di frasi ST incluse nel codice Motion Control.



Per usare testo strutturato in un passo Motion control, seguire queste linee guida:

- Racchiudere il codice ST tra parentesi graffe { }, come si vede nei blocchi 1, 3 e 4 del passo Step2 della figura qui sopra..
- Per un blocco contenente testo strutturato, il comando N è facoltativo.
- Anche il punto e virgola, che si usa normalmente per chiudere le righe di testo strutturato, è facoltativo.
- I parametri di tutti i comandi Motion Control, salvo N e G, possono essere scritti in testo strutturato.
- Ad esempio, N016 G4F{tempo_di_sosta} è un blocco valido e tempo_di_sosta è un simbolo ST contenente la durata della sosta specificata dal comando F.
- N100 G{comando_prep} non è valido. Il simbolo ST comando_prep non può essere usato come parametro del comando G.
- Nelle espressioni ST è possibile usare simboli locali o globali. I simboli possono essere simboli semplici o elementi di un array. I valori di questi simboli possono essere loro assegnati all'interno del programma (passo Step1 nella figura) o scaricati mediante DDE (Dynamic Data Exchange = scambio dinamico dei dati).

- E' possibile usare frasi di assegnazione in blocchi separati (passo Step2, blocchi 3 e 4, nella figura).
- Sono disponibili tre blocchi funzionali ST per la specifica di comandi di movimento all'interno di un passo SFC codificato in testo strutturato: AXSJOG, MOVEAXS, e STOPJOG (pagina E-30).

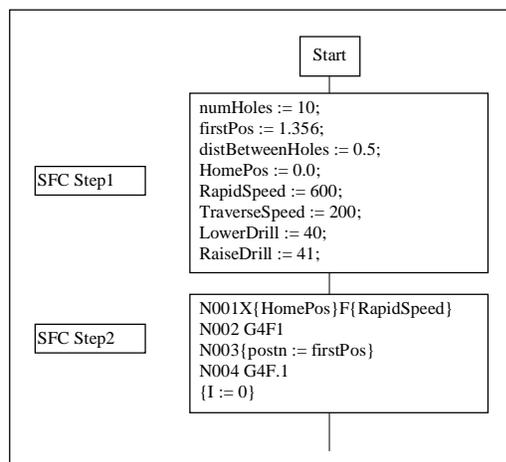
Valutazione del testo strutturato incluso nel codice Motion Control

Quando il flusso del programma incontra un blocco contenente testo strutturato, l'esecuzione del programma e tutte le attività Motion sono momentaneamente sospese; poi il flusso del programma continua con il blocco che segue immediatamente il testo strutturato.

Le espressioni sono valutate al momento dell'esecuzione del blocco e possono rappresentare il risultato di ingressi in tempo reale, quali i dati dei sensori. Essi sono valutati come interi o reali, a seconda di come è appropriato per la parola Motion Control.

Poiché il software del sistema di comando ha una funzione "leggi-avanti", le frasi di assegnazione ST sono lette ed eseguite prima che il flusso del programma incontri il blocco che le contiene. E' possibile controllare l'esecuzione di una assegnazione ST inserendo un comando G04 nel blocco che precede quello contenente l'assegnazione. La funzione "leggi-avanti" non si estende oltre il comando G04.

Ad esempio, nella sottostante figura la frase di assegnazione contenuta nel quinto blocco del passo Step2 non è eseguita finché non viene incontrata dal flusso del programma. Senza il comando G04 del quarto blocco, il simbolo I potrebbe essere messo a zero prima dell'esecuzione del secondo e del terzo blocco.



Funzioni Motion Control del linguaggio Testo strutturato

Il linguaggio ST dispone delle seguenti funzioni Motion Control: AXSJOG, MOVEAXS e STOPJOG.

AXSJOG

La funzione AXSJOG avvia un comando jog per l'asse specificato.

Formato:

AXSJOG(<asse>, JOGDIR:= <direzione> , JOGTYPE:= <tipo>, JOGSPD:= <velocità>, JOGDIST:= <distanza>);

dove:

<asse> è il nome dell'asse (X, Y, Z, ecc.)

<direzione> è la direzione del movimento in jog (JOGPLUS o JOGMINUS)

<tipo> è il tipo di movimento in jog (JOGCONT = continuo, JOGINCR = incrementale, JOGHOME = ritorno a zero)

<velocità> e <distanza> sono numeri reali, simboli, o espressioni il cui risultato è un numero reale, che specificano la velocità e l'entità del movimento.

Esempio:

```
AXSJOG (X, JOGDIR:=JOGMINUS, JOGTYPE:=JOGCONT,
JOGSPD:=10.0);
```

MOVEAXS

La funzione MOVEAXS avvia un comando di movimento dell'asse specificato.

Formato:

AXSJOG (<asse>, POSTN:= <posizione> , VEL:= <velocità>, ACCEL:= <accelerazione>);

Dove:

<asse> è il nome dell'asse (X, Y, Z, ecc.)

<posizione>, <velocità>, e <accelerazione> sono numeri reali, simboli, o espressioni il cui risultato è un numero reale, che specificano la posizione finale, la velocità e l'accelerazione del movimento

Esempio:

```
MOVEAXS (X, POSTN:= positionA, VEL:= velocityA,
ACCEL:=10.0);
```

STOPJOG

La funzione STOPJOG ferma il comando jog dell'asse specificato.

Formato:

STOPJOG(<asse>);

Dove:

<asse> è il nome dell'asse (x, y, z, ecc.).

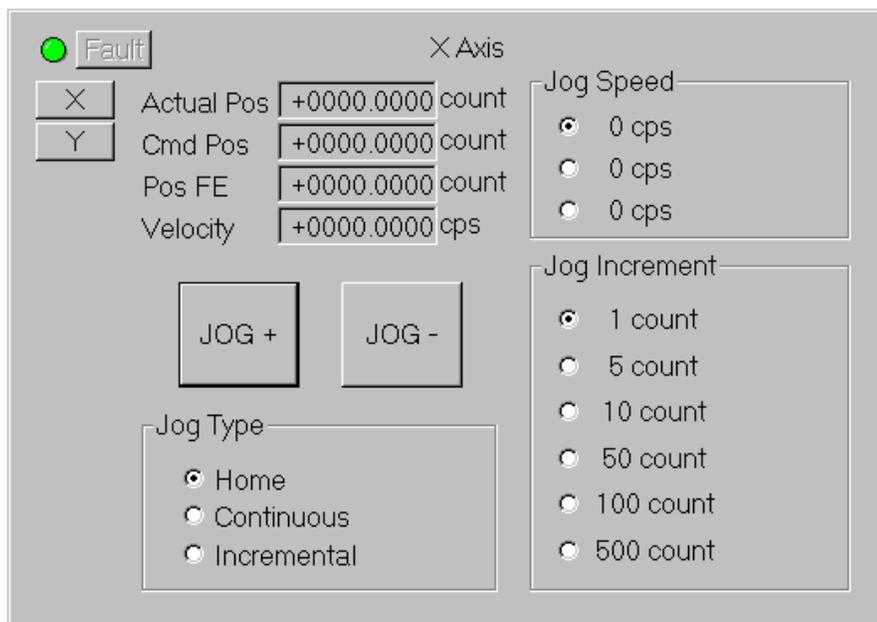
Esempio:

STOPJOG (Z);

Controlli Motion dell'interfaccia con l'operatore

Pannello Jog

Il pannello Jog fornisce le funzioni necessarie per l'azzeramento e il movimento manuale degli assi configurati. Questo pannello contiene i pulsanti per la selezione dell'asse, del tipo di movimento (azzeramento, avanzamento continuo, avanzamento incrementale) la velocità e l'incremento. L'asse selezionato è visualizzato nella parte superiore del pannello. Il pannello contiene anche la posizione assoluta, la posizione comandata, l'errore di inseguimento e la velocità dell'asse selezionato. Un indicatore segnala un eventuale errore dell'asse selezionato. Se non ci sono errori, l'indicatore è verde. Se si verifica un errore, l'indicatore diventa rosso e viene abilitato il pulsante di errore. Per visualizzare i dettagli dell'errore verificatosi, premere questo pulsante. Quando l'errore asse viene cancellato, l'indicatore ritorna verde. L'asse selezionato viene mosso o azzerato nella direzione desiderata premendo uno dei pulsanti Jog+ e Jog-. Il movimento o il ritorno a zero terminano quando il pulsante viene rilasciato. Nel caso del movimento incrementale, l'asse si ferma al termine dell'incremento. Per eseguire un altro incremento, rilasciare e premere di nuovo il pulsante.

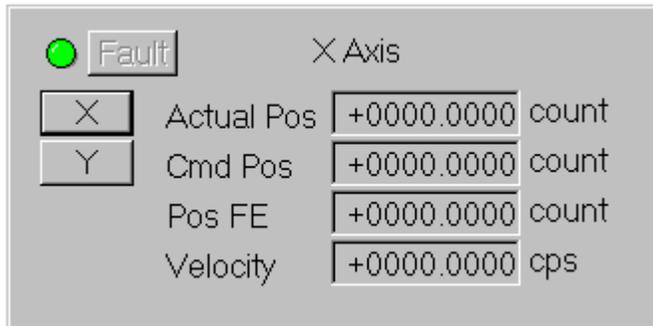


Il pannello Jog è impostato automaticamente per tutti gli assi configurati. Non esistono opzioni di configurazione del pannello Jog.

Pannello di stato monoasse

Il pannello Single Axis Status visualizza lo stato di un asse. Per visualizzare lo stato di un asse, premere il corrispondente pulsante asse del pannello. L'asse del quale è visualizzato lo stato si seleziona premendo il pulsante asse. Il nome dell'asse selezionato compare nella parte superiore del pannello. I dati di stato visualizzati sono: la posizione assoluta, la posizione comandata, l'errore d'inseguimento e la velocità. Un indicatore segnala un eventuale errore dell'asse selezionato. Se non ci sono errori, l'indicatore è verde. Se si verifica un errore, l'indicatore diventa rosso e viene abilitato il pulsante di errore. Per visualizzare i dettagli dell'errore verificatosi, premere questo pulsante. Quando l'errore asse viene cancellato, l'indicatore ritorna verde.

Il pannello monoasse è impostato automaticamente per tutti gli assi configurati. Non esistono opzioni di configurazione per questo pannello.



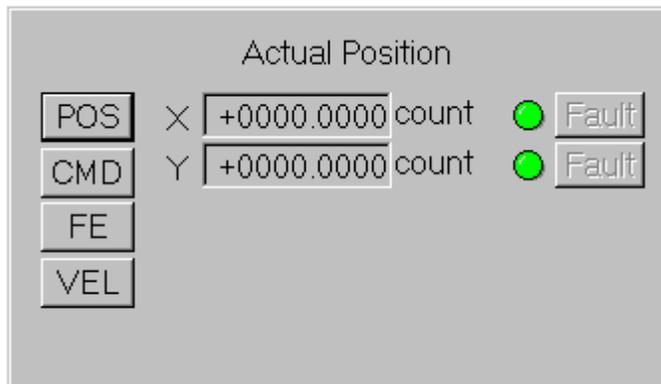
Uso del pannello di stato monoasse

L'asse selezionato è visualizzato nella parte superiore del pannello. Per cambiare l'asse selezionato, premere il pulsante corrispondente all'asse che si desidera visualizzare. E' possibile attivare più copie di questo pannello per vedere lo stato completo di più di un asse per volta.

Pannello di stato multiasse

Il pannello Multi Axis Status visualizza lo stato specificato di tutti gli assi configurati. Lo stato da visualizzare si seleziona tra i seguenti: posizione assoluta (POS), posizione comandata (CMD), errore d'inseguimento (FE) e velocità (VEL). La selezione dello stato da visualizzare si opera premendo il corrispondente pulsante del pannello. Il nome dello stato selezionato compare nella parte superiore del pannello. Per ciascun asse esiste un indicatore che segnala un eventuale errore su quell'asse. Se non ci sono errori, gli indicatori sono verdi. Se si verifica un errore asse, l'indicatore dell'asse in errore diventa rosso e viene abilitato il pulsante di errore per quell'asse. Per visualizzare i dettagli dell'errore verificatosi, premere questo pulsante. Quando l'errore asse viene cancellato, l'indicatore ritorna verde.

Il pannello multiasse è impostato automaticamente per tutti gli assi configurati. Non esistono opzioni di configurazione per questo pannello.



Uso del pannello di stato multiasse

Il nome del dato di stato visualizzato compare nella parte superiore del pannello. Per cambiare il dato visualizzato, premere il pulsante corrispondente al dato desiderato. E' possibile attivare contemporaneamente più copie di questo pannello.

Finestra dei blocchi RS274

La finestra dei blocchi RS274 permette di visualizzare i blocchi RS274 attivi sullo schermo dell'interfaccia operatore. Sono visualizzati tutti i blocchi contenuti nel passo SFC attivo. Il blocco attivo è evidenziato in verde ed è indicato da un puntatore (>).

Editazione di un blocco RS274

Per editare un blocco RS274, fare doppio click sul blocco stesso o selezionare il blocco e premere il tasto invio. Comparirà la finestra di dialogo Edit RS274 Block Display, che permette all'utente di cambiare i dati definiti nel blocco.

La casella combinata (combo box) RS274 Program Symbol serve per definire il simbolo di tipo STRING che contiene il percorso del file del programma SFC del quale saranno visualizzati i blocchi RS274 attivi. Selezionare il nome del simbolo nella lista a discesa della casella o digitarlo direttamente.

Questa funzione usa la tecnologia OLE per creare il collegamento con il Program Editor. Il Program Editor visualizza i blocchi RS274 contenuti nella finestra RS274 Block Display dello schermo operatore. Quando cambia il simbolo che definisce il programma, la finestra RS274 Block Display si collega al nuovo programma SFC.

Sono visualizzati solo i blocchi RS274 contenuti nei programmi SFC di livello più alto. Se i blocchi RS274 si trovano in un SFC figlio, essi non sono visualizzati.

Considerazioni sulle prestazioni

Per collegare la finestra RS274 Block Display al Program Editor, il sottosistema OLE deve avviare l'applicazione Program Editor. Su alcuni sistemi, il tempo richiesto da questa operazione può causare un ritardo eccessivo nell'interfaccia operatore. Per eliminare questo ritardo, si suggerisce di:

- 1) Avviare il Program Editor minimizzato in background (se lo si desidera con il solo accesso in lettura) prima di avviare l'interfaccia operatore.
- 2) Inizializzare tutti i simboli RS274 Block Display con i percorsi dei file SFC desiderati. Dato che tutti i collegamenti OLE sono realizzati quando viene attivata l'interfaccia operatore, il ritardo dovuto al collegamento si verifica all'avvio del sistema.

Diffida

**I puntatori devono essere usati esclusivamente da esperti.
Un uso errato dei puntatori può dare luogo a operazioni
imprevedibili e a grandi difficoltà di messa a punto del
programma.**

Il linguaggio Testo strutturato dispone di due operatori riservati ai puntatori: l'operatore indirizzo & e l'operatore di indirizzamento *. Questi operatori si usano nelle operazioni di *indirizzamento indiretto*.

Questa appendice fornisce le seguenti informazioni:

- Indirizzamento
- Operatori riservati ai puntatori
- Definizione dei simboli di tipo puntatore
- Puntatori ad array

Indirizzamento

Nell'*indirizzamento diretto*, a un dato viene attribuito un nome simbolico, che si usa per accedere direttamente a quel dato. Nell'esempio che segue, a X è assegnato il valore del dato Y.

```
X := Y;
```

Nell'*indirizzamento indiretto*, un nome simbolico, comunemente detto puntatore, rimanda alla posizione di memoria nella quale è memorizzato il dato. Per leggere il valore di un dato con l'indirizzamento indiretto, si utilizza un puntatore per ottenere l'indirizzo del dato poi si legge la posizione di memoria corrispondente a tale indirizzo.

Nel seguente esempio, pVar1 è un puntatore..

```
pVar1 := & X; a pVar1 viene assegnato l'indirizzo di memoria del dato X.
```

```
Y := * pVar1; dato che ora pVar1 contiene l'indirizzo di X, a Y è assegnato il valore di X.
```

```
* pVar1 := Y; a X è assegnato il valore di Y.
```

Operatori riservati ai puntatori

Assumiamo che VarInt1 e VarInt2 siano simboli interi e che plnt sia definito come puntatore ad interi.

Per assegnare un indirizzo ad un puntatore

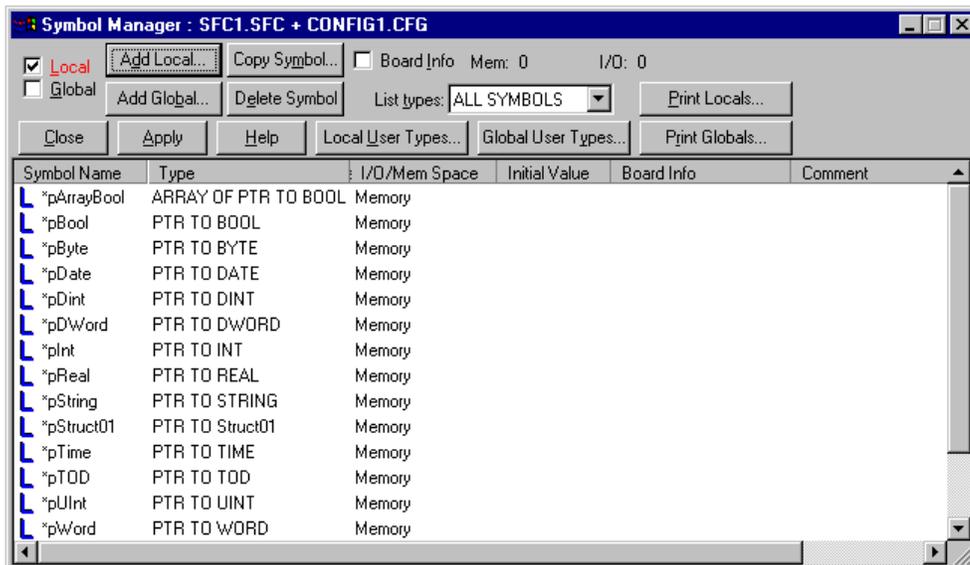
```
plnt := & VarInt1; a plnt viene assegnato l'indirizzo di VarInt1. "&"  
significa "ottieni l'indirizzo di".
```

Per ottenere il dato indirizzato dal un puntatore

```
VarInt2 := * plnt; a VarInt2 viene assegnato il valore VarInt1. "**"  
significa "ottieni il valore all'indirizzo".
```

Definizione dei simboli di tipo puntatore

Come per gli altri simboli, i puntatori si definiscono nel Symbol Manager. La sottostante figura presenta la definizione di vari tipi di puntatori.



Note sui puntatori

1. Quando viene definito un puntatore, esso è definito come puntatore ad un simbolo di uno specifico tipo di dati (REAL, INT, STRING, ecc.). Ad esempio, al puntatore `pVar1` può essere assegnato qualsiasi simbolo del tipo specificato nella sua definizione.
2. Possono essere definiti puntatori a tipi di dati standard, strutture definite dall'utente e array. *Non possono* essere definiti puntatori a blocchi funzionali e oggetti del sistema.
3. Un puntatore può essere sostituito a un simbolo del suo tipo base, antepoendo al puntatore l'operatore di indirizzamento *. (`*pVarInt1 = VarInt1`, purché sia stata eseguita l'assegnazione `pVarInt1:=&VarInt1`.)

4. Un puntatore ad una struttura può essere usato direttamente al posto del nome della struttura. Ad esempio:

Assumiamo che sia stata definita la struttura utente `UserStruct1` e che `pStruct1` sia stato definito come puntatore a tale struttura.

Il nome di una struttura definita dall'utente (in questo caso `UserStruct1`) è un puntatore ai dati della struttura. Tuttavia, al nome della struttura non può essere assegnato un indirizzo diverso, punterà sempre a quella struttura.

```
pStruct1 := UserStruct1;      a pStruct1 viene assegnato
                              l'indirizzo di UserStruct1.
```

```
pStruct1.intMember1 := VarInt1; Al membro intMember1 di
                                UserStruct1 viene assegnato il
                                valore di VarInt1.
```

5. Un puntatore ad un array (ad esempio, `pInt:= &intArray[5];`) può essere usato con l'operatore di indirizzamento nell'array (`pInt[Index]`) per fare riferimento ai singoli elementi dell'array. Questo indirizzo parte da 0 (l'elemento dell'array puntato dal puntatore) e continua fino alla fine dell'array. Ad esempio:

`intArray` è definito come un array di 10 interi (`ARRAY[1..10]`)

`pInt` è definito come un puntatore ad interi

Allora:

```
pInt:= &intArray[5];
```

```
pInt[0]:= 0;          (*intArray[5]*)
```

```
pInt[5]:= 5;          (*intArray[10]*)
```

Altri esempi di utilizzo di puntatori ad array si trovano a pagina E-5.

6. A un puntatore può essere assegnato un valore nullo:

```
pInt := NULL;    a pInt è assegnato il valore NULL. Un puntatore può
                 ricevere e può essere confrontato (uguale e non
                 uguale a) con NULL. NULL è una parola chiave del
                 sistema.
```

7. Un puntatore può essere inizializzato in un blocco funzionale MOVE. In questo caso l'uscita del blocco funzionale MOVE è il puntatore da inizializzare e l'ingresso è un altro puntatore o un simbolo diretto preceduto dall'operatore indirizzo `&`.
8. I puntatori *non possono* essere passati alle funzioni FILE, alle funzioni che operano su array di bit (`SHL`, `AND_BITS`, ecc.), e alle funzioni `STRING_TO_ARRAY`.

Puntatori ad array

Per usare un puntatore ad array

Assumiamo che `IntArray` sia un array di dieci interi (`IntArray: ARRAY [1..10] OF INT`) e che `pArray` sia definito come un puntatore ad interi.

Per i simboli di tipo array, il nome dell'array (in questo caso `IntArray`) è simile ad un puntatore ai dati dell'array. Tuttavia, al nome dell'array non potrà mai essere assegnato un indirizzo diverso, esso punterà sempre all'array.

`pArray := &IntArray;` a `pArray` viene assegnato l'indirizzo di `IntArray`. `pArray` punta al primo elemento dell'array.

`pArray[indice] := VarInt1;` a `IntArray[indice]` viene assegnato il valore di `VarInt1`.

`pArray := & IntArray[indice];` `pArray` punta all'elemento dell'array specificato dall'indice.

Per usare un puntatore ad un array di strutture definite dall'utente

Assumiamo che `structArray` sia un array di strutture utente (`structArray : ARRAY [1..10] OF USER_STRUCT1`) e che `pStructArray` sia definito come un puntatore alla stessa struttura utente (`pStructArray : PTR TO USER_STRUCT1`).

`pStructArray := &structArray;` a `pStructArray` viene assegnato l'indirizzo di `structArray`.

`pStructArray[5].intMember1 := VarInt1;` al membro `intMember1` di `structArray[5]` viene assegnato il valore di `VarInt1`.

Per usare un array di puntatori che puntano ad un INT

Assumiamo che sia definito l'array di puntatori `ArrayPtr` (`ARRAY [1..10] OF PTR TO INT`).

`ArrayPtr[index] := &VarInt1;` a `ArrayPtr[index]` viene assegnato l'indirizzo di `VarInt1`.

`*ArrayPtr[index] := VarInt2;` a `VarInt1` viene assegnato il valore di `VarInt2`.

Per usare un array di puntatori che puntano ad un array di INT

Assumiamo che siano definiti l'array di interi `IntArray` (ARRAY [1..10] OF INT) e l'array di puntatori `ArrayPtr` (ARRAY [1..10] OF PTR TO INT).

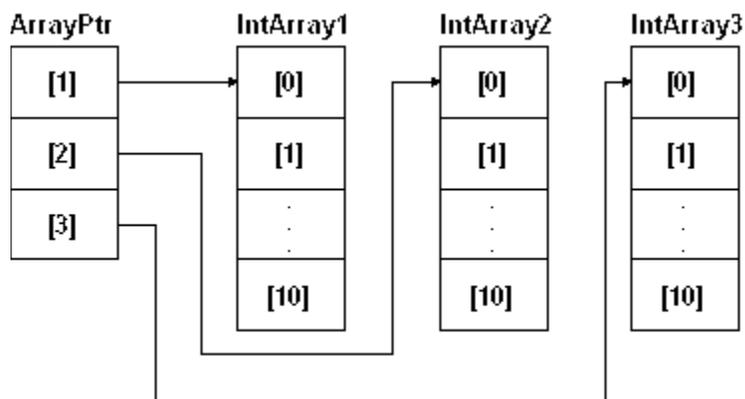
<code>ArrayPtr[indice] := &IntArray;</code>	ad <code>ArrayPtr[indice]</code> viene assegnato l'indirizzo del primo elemento di <code>IntArray</code> .
<code>*ArrayPtr[indice] := VarInt1;</code>	a <code>IntArray[1]</code> viene assegnato il valore di <code>VarInt1</code> .
<code>ArrayPtr[index][1] := VarInt1;</code>	questa assegnazione è uguale alla precedente.

Per usare un array di puntatori ad array di INT

Assumiamo che siano definiti gli array di interi `IntArray1`, `IntArray2`, `IntArray3` (ARRAY [0..10] OF INT) e l'array di puntatori `ArrayPtr` (ARRAY [1..3] OF PTR TO INT).

```
ArrayPtr[1] := &IntArray1[0];  
ArrayPtr[2] := &IntArray2[0];  
ArrayPtr[3] := &IntArray3[0];
```

Fare riferimento alla sottostante figura.



Può essere usata la seguente sintassi:

ArrayPtr[1] [0] := VarInt1; (*IntArray1[0] = VarInt1*)

ArrayPtr[1] [1] := VarInt2; (*IntArray1[1] = VarInt2*)

ArrayPtr[1] [2] := VarInt3; (*IntArray1[2] = VarInt3*)

ArrayPtr[2] [0] := VarInt1; (*IntArray2[0] = VarInt1*)

ArrayPtr[2] [1] := VarInt2; (*IntArray2[1] = VarInt2*)

ArrayPtr[2] [2] := VarInt3; (*IntArray2[2] = VarInt3*)

ArrayPtr[3] [0] := VarInt1; (*IntArray3[0] = VarInt1*)

ArrayPtr[3] [1] := VarInt2; (*IntArray3[1] = VarInt2*)

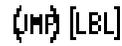
ArrayPtr[3] [2] := VarInt3; (*IntArray3[2] = VarInt3*)

Questa appendice elenca sinteticamente le istruzioni dei linguaggi RLL, SFC, Testo strutturato e Lista di istruzioni. Per le spiegazioni dettagliate si rinvia all'aiuto in linea.

Set di istruzioni del linguaggio RLL

Contatti e bobine

Tipo	Pulsante della barra degli strumenti	Descrizione
Contatto normalmente aperto		Il contatto lascia passare il flusso di corrente se il punto che rappresenta è TRUE.
Contatto normalmente chiuso		Il contatto lascia passare il flusso di corrente se il punto che rappresenta è FALSE.
Contatto rilevatore di transizione positiva		Il contatto lascia passare il flusso di corrente se il punto che rappresenta passa da FALSE a TRUE immediatamente prima della sua valutazione.
Contatto rilevatore di transizione negativa		Il contatto lascia passare il flusso di corrente se il punto che rappresenta passa da TRUE a FALSE immediatamente prima della sua valutazione.
Bobina di uscita		Il punto rappresentato dalla bobina è TRUE quando la bobina riceve corrente.
Bobina di uscita negata		Il punto rappresentato dalla bobina è TRUE quando la bobina non riceve corrente.
Bobina SET (Latch)		Il punto rappresentato dalla bobina diventa TRUE quando la bobina riceve corrente. Può essere messo a FALSE solo da una bobina RESET.
Bobina RESET (Unlatch)		Il punto rappresentato dalla bobina diventa FALSE quando la bobina riceve corrente. Il punto rimane FALSE anche dopo che cessa il flusso di corrente alla bobina e può essere messo a TRUE solo da una bobina SET.
Bobina rilevatrice di transizione positiva		Quando il flusso di corrente che perviene alla bobina passa da FALSE a TRUE, il punto che rappresenta diventa TRUE e rimane TRUE per un ciclo di scansione, a meno che non sia messo a FALSE altrove.

Tipo	Pulsante della barra degli strumenti	Descrizione
Bobina rilevatrice di transizione negativa		Quando il flusso di corrente che perviene alla bobina passa da TRUE a FALSE, il punto che rappresenta diventa TRUE e rimane TRUE per un ciclo di scansione, a meno che non sia messo a FALSE altrove.
Bobina di salto/etichetta		La coppia salto/etichetta si usa per disabilitare temporaneamente una sezione del programma. I due elementi devono sempre essere usati insieme. Un'etichetta senza salto causa un errore al runtime.
Bobina di transizione SFC		La bobina di transizione SFC è un elemento RLL che in certe condizioni (all'interno di un'azione SFC) può essere usato in un programma SFC.

Stringhe di Bit

Tipo	Abbr.	Descrizione
AND logico	AND	Calcola l'AND a livello di bit di due numeri.
NOT logico	NOT	Calcola il complemento a livello di bit di un numero..
OR logico	OR	Calcola l'OR a livello di bit di due numeri.
Rotazione a sinistra	ROL	Ruota verso sinistra l'ingresso per il numero specificato di bit.
Rotazione a destra	ROR	Ruota verso destra l'ingresso per il numero specificato di bit.
Spostamento a sinistra	SHL	Sposta verso sinistra l'ingresso per il numero specificato di bit.
Spostamento a destra	SHR	Sposta verso destra l'ingresso per il numero specificato di bit.
OR logico esclusivo	XOR	Calcola l'OR esclusivo a livello di bit di due numeri.

Stringhe di caratteri

Tipo	Abbr.	Descrizione
Concatenazione	CAT	Concatena la seconda stringa in ingresso alla fine della prima.
Cancellazione	DEL	Cancella uno o più caratteri dalla stringa in ingresso.
Ricerca	FIND	Cerca una stringa in ingresso all'interno di un'altra.
Inserimento	INS	Inserisce una stringa in ingresso in un'altra.
Copia a sinistra	LEFT	Copia i caratteri di sinistra della stringa in ingresso.
Lunghezza	LEN	Memorizza la lunghezza della stringa.
Copia all'interno	MID	Copia uno o più caratteri dall'interno di una stringa.
Copia a destra	RGHT	Copia i caratteri di destra della stringa in ingresso.
Sostituzione	RPLC	Sostituisce uno o più caratteri della stringa in ingresso con un'altra stringa in ingresso
Da array di byte a stringa	BATOS	Memorizza i byte dell'array in ingresso come caratteri in una stringa.
Da intero ad ASCII	ITOA	Converte l'intero in ingresso in una stringa ASCII
Visualizzazione di un messaggio sullo schermo operatore	MSGB	Visualizza un messaggio sullo schermo di interfaccia con l'operatore.
Visualizzazione di un messaggio nella Output Window	MSGW	Visualizza un messaggio nella Output Window..
Da reale ad ASCII	RTOA	Converte il numero reale in ingresso in una stringa ACSII.
Da stringa ad array di byte	STOBA	Memorizza i caratteri della stringa in ingresso in un array di byte.

Confronto (String+)

Tipo	Abbr.	Descrizione
Uguale	EQ	Verifica se due ingressi sono tra loro uguali.
Maggiore o uguale	GE	Verifica se il primo ingresso è maggiore o uguale al secondo.
Maggiore	GT	Verifica se il primo ingresso è maggiore del secondo.
Minore o uguale	LE	Verifica se il primo ingresso è minore o uguale al secondo.
Minore	LT	Verifica se il primo ingresso è minore del secondo.
Non uguale	NE	Verifica se due ingressi sono tra loro diversi.

Conversione

Tipo	Abbr.	Descrizione
Da array di byte a stringa	BATOS	Converte un array di byte in un valore STRING.
Da data a stringa	DateToString	Converte un valore DATE in un valore STRING.
Da intero a stringa	ITOA	Converte un valore INT in un valore STRING.
Da reale a stringa	RTOA	Converte un valore REAL in un valore STRING.
Da RGB a DWORD	RGB_TO_DWORD	Converte una tripletta di valori di rosso, verde e blu in un valore DWORD.
Da stringa ad array di byte	STOBA	Converte un valore STRING in un array di byte.
Da stringa a data	StringToDate	Converte un valore STRING in un valore DATE.
Da stringa a intero	ATOI	Converte una stringa di caratteri numerici ASCII in un valore INT.
Da stringa a reale	ATOR	Converte una stringa di caratteri numerici ASCII in un valore REAL.
Da stringa a ora del giorno	StringToTOD	Converte un valore STRING in un valore TOD.
Da ora del giorno a stringa	TODToString	Converte un valore TOD in un valore STRING.

Contatori e temporizzatori

Tipo	Abbr.	Descrizione
Contatore a decremento	CTD	Conta gli eventi decrementandosi di uno.
Contatore a incremento	CTU	Conta gli eventi incrementandosi di uno.
Contatore a incremento/decremento	CTUD	Conta gli eventi incrementandosi o decrementandosi.
Temporizzatore di ritardo alla disattivazione	TOF	Disattiva un'uscita dopo il ritardo presettato.
Temporizzatore di ritardo all'attivazione	TON	Attiva un'uscita dopo il ritardo presettato.
Temporizzatore a impulso	TP	E' attivato da un impulso, poi funziona come il temporizzatore TOF.

Rilevazione fronti

Tipo	Abbr.	Descrizione
Attivazione al fronte di discesa	FTRG	Attiva un'uscita quando rileva un fronte di discesa.
Attivazione al fronte di salita	RTRG	Attiva un'uscita quando rileva un fronte di salita.

Temporizzatori estesi

Temporizzatore esteso di ritardo alla disattivazione	XTOF	Conta il trascorrere del tempo se non è alimentato.
Temporizzatore esteso di ritardo all'attivazione	XTON	Conta il trascorrere del tempo se è alimentato.
Temporizzatore esteso a impulso	XTP	E' attivato da un impulso, poi conta il trascorrere del tempo se non è alimentato.

File

Tipo	Abbr.	Descrizione
Chiudi file	FCLOS	Chiude un file.
Copia	FCOPY	Copia un file.
Cancella File	FDEL	Cancella un file.
Crea File	FNEW	Crea un nuovo file.
Apri File	FOPEN	Apri un file esistente.
Leggi File	FREAD	Legge i dati di un file.
Riavvolgi File	FRWN D	Riporta un file all'inizio.
Scrivi File	FWRIT	Scrivi i dati in un file.

Matematica

Tipo	Abbr.	Descrizione
Valore assoluto	ABS	Calcola il valore assoluto di un numero.
Addizione	ADD	Somma due numeri
Divisione	DIV	Divide un numero per un altro.
Potenza	EXPT	Eleva il primo numero alla potenza specificata dal secondo numero.
Modulo	MOD	Divide un numero per un altro e memorizza il resto.
Trasferimento	MOVE	Copia un dato da una posizione ad un'altra.
Moltiplicazione	MUL	Moltiplica due numeri.
Negazione	NEG	Nega (inverte) gli ingressi.
Radice quadrata	SQRT	Calcola la radice quadrata di un numero.
Sottrazione	SUB	Sottrae un numero da un altro.

Miscellanee

Tipo	Abbr.	Descrizione
Termina tutti i programmi	ABTAL	Termina tutti i programmi in esecuzione nei sottosistemi di runtime.
Visualizza messaggio	MSGB	Visualizza un messaggio in una finestra di messaggio Windows
Messaggio	MSGW	Visualizza un messaggio nella Output Window del Program Editor.
Cambia schermo MMI	Change MMIScreen	Visualizza lo schermo HMI specificato.

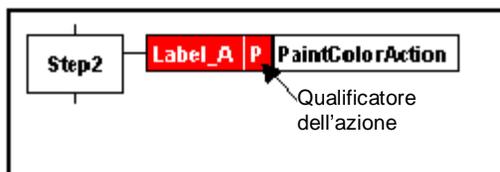
Trigonometriche e Logaritmiche

Tipo	Abbr.	Descrizione
Arcocoseno	ACOS	Calcola l'arcocoseno di un numero.
Arcoseno	ASIN	Calcola l'arcoseno di un numero.
Arcotangente	ATAN	Calcola l'arcotangente di un numero.
Coseno	COS	Calcola il coseno di un numero.
Esponenziale	EXP	Calcola l'esponenziale naturale di un numero.
Logaritmo naturale	LN	Calcola il logaritmo naturale di un numero.
Logaritmo	LOG	Calcola il logaritmo in base 10 di un numero.
Seno	SIN	Calcola il seno di un numero.
Tangente	TAN	Calcola la tangente di un numero.

Set di istruzioni del linguaggio SFC

Istruzione	Descrizione
Passo	Un passo rappresenta una situazione nella quale il comportamento del processo è definito dalle azioni associate al passo.
Transizione RLL	Quando il flusso di corrente che percorre il rung RLL raggiunge la bobina di uscita, attivandola, la transizione diventa TRUE, permettendo al programma di continuare con il passo successivo.
Transizione booleana	Quando il risultato dell'espressione booleana è TRUE, la transizione diventa TRUE, permettendo al programma di continuare con il passo successivo.
Passo macro	Il passo macro dà modo di richiamare un SFC da un passo di un altro SFC.
Azione	Un'azione è una porzione di codice che viene eseguita quando il passo al quale l'azione è associata diventa attivo. I qualificatori di azione specificano le condizioni di esecuzione del codice RLL contenuto nell'azione.
Funzione salto/etichetta	Trasferisce il flusso del programma all'etichetta specificata.
Anello	Permette all'esecuzione di un programma SFC di ritornare ad un punto precedente allo scopo di ripetere una serie di passi.
Divergenza selettiva	Una divergenza selettiva permette di scegliere il percorso attivo fra due o più percorsi alternativi.
Divergenza simultanea	Una divergenza simultanea permette di attivare simultaneamente più percorsi in parallelo.
Blocco di controllo del programma (PRGCB)	Permette ad un programma applicativo SFC di compilare e di controllare l'esecuzione di altri programmi applicativi SFC o RLL.

I qualificatori compaiono in un'azione come segue:



Qualificatori di azione

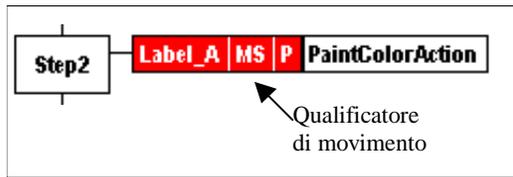
Se dopo l'attivazione del passo si vuole che	Usare il qualificatore
L'esecuzione dell'RLL inizi e continui finché il passo è attivo.	Non Stored (N)
L'esecuzione dell'RLL inizi e continui finché non viene resettata dal qualificatore Reset.	Stored (S)
L'RLL venga eseguito una sola volta.	Pulsed (P)
L'esecuzione dell'RLL inizi dopo un ritardo* e continui finché il passo è attivo.	Time Delayed (D)
L'esecuzione dell'RLL inizi e continui finché non scade il limite di tempo o il passo non diventa inattivo.	Time Limited (L)
L'esecuzione dell'RLL inizi dopo un ritardo* e continui finché non viene resettata dal qualificatore Reset. Se un altro qualificatore di azione resetta l'RLL durante il ritardo, il reset non ha effetto perché l'RLL non è ancora stata memorizzata. Se il passo diventa inattivo durante il ritardo, l'RLL non viene memorizzata né eseguita.	Delayed and Stored (DS)
L'esecuzione dell'RLL inizi dopo un ritardo* e continui finché non viene resettata dal qualificatore Reset. Se un'azione viene resettata durante il ritardo, l'RLL non è eseguita.	Stored and Time Delayed (SD)
L'esecuzione dell'RLL inizi e continui per il tempo specificato*. Per poter essere eseguita di nuovo, l'RLL deve essere resettata da un qualificatore Reset. Se il passo diventa inattivo, l'esecuzione dell'RLL continua finché non scade il limite di tempo. Non è richiesto un qualificatore Reset, ma se ne può usare uno per fermare l'esecuzione dell'RLL.	Stored and Time Limited (SL)
L'esecuzione dell'RLL inizi e continui per il tempo specificato*. Se il passo diventa inattivo, l'esecuzione dell'RLL continua finché non scade il limite di tempo. Non è richiesto un qualificatore Reset, ma se ne può usare uno per fermare l'esecuzione dell'RLL.	Pulse Width (PW)
L'RLL iniziata da un qualificatore Stored (memorizzata) è terminata dal qualificatore Reset (R). Il qualificatore Reset può essere usato in un'altra azione associata allo stesso passo o in un'azione associata ad un altro passo. L'RLL continua a girare fra i passi. Se l'azione è associata ad un altro passo, essa deve avere lo stesso nome dell'azione da resettare.	Reset (R)

*Per i dettagli, fare riferimento al paragrafo "Durata" del capitolo 4.

Qualificatori di movimento

Nota: Non tutti i qualificatori di movimento sono supportati da tutte le schede di controllo dei movimenti. Per sapere quali qualificatori sono supportati, fare riferimento alla documentazione della scheda..

I qualificatori di movimento specificano vincoli di movimento che devono essere soddisfatti prima che inizi l'esecuzione del codice RLL. Se è specificata un'etichetta di programma, il vincolo si applica ai blocchi di movimento successivi all'etichetta. Questi qualificatori compaiono nell'azione come segue:



Scegliere tra i seguenti qualificatori. Lasciare in bianco se non deve essere specificato un qualificatore di movimento..

Se si vuole che l'esecuzione del codice RLL non inizi prima che:	Usare questo qualificatore:
inizi il movimento	Motion Started (MS)
sia completato il profilo di accelerazione	Acceleration Complete (AC)
il movimento raggiunga la velocità specificata	At Speed (AS)
inizi il profilo di decelerazione	Deceleration Started (DS)
sia finito il movimento	Motion Complete (MC)
sia finito il movimento e tutti gli assi associati al movimento siano entro la tolleranza di posizione rispetto al punto finale programmato	In Position (IP)
sia completata l'esecuzione del comando	End of Block (EB)

Set di istruzioni del linguaggio Testo strutturato

Fraasi

Istruzione	Descrizione
Assegnazione	Sostituisce il valore di un oggetto con il risultato di un'espressione.
CASE	Esegue un gruppo di frasi selezionandolo fra altri gruppi possibili in base al valore di una variabile.
Commento	Si usa per incorporare utili annotazioni nel codice del programma.
EXIT	Termina un processo iterativo come FOR o WHILE.
FOR	Si usa per eseguire ripetutamente una serie di frasi; il numero di ripetizioni dipende dal valore di una variabile di controllo intera.
Richiamo di funzione	Esegue un algoritmo predefinito.

Stringhe di Bit

Istruzione	Descrizione
AND	Restituisce l'AND booleano o logico a livello di bit dei valori in ingresso.
NOT	Restituisce l'OR booleano o logico a livello di bit dei valori in ingresso.
OR esclusivo (XOR)	Restituisce l'OR esclusivo booleano o logico a livello di bit dei valori in ingresso.
Rotazione a sinistra (ROL)	Restituisce un valore calcolato spostando circolarmente verso sinistra i bit del valore in ingresso, per il numero specificato di posizioni. I valori spostati dal bit più significativo (MSB = Most Significant Bit) sono ruotati nel bit meno significativo (LSB = Least Significant Bit).
Rotazione a destra (ROR)	Restituisce un valore calcolato spostando circolarmente verso destra i bit del valore in ingresso, per il numero specificato di posizioni. I valori spostati dal bit meno significativo (LSB = Least Significant Bit) sono ruotati nel bit più significativo (MSB = Most Significant Bit).
Spostamento a sinistra (SHL)	Restituisce un valore calcolato spostando verso sinistra i bit del valore in ingresso, per il numero specificato di posizioni. I valori spostati dal bit più significativo sono scartati e i bit meno significativi sono riempiti di zeri.
Spostamento a destra (SHR)	Restituisce un valore calcolato spostando verso destra i bit del valore in ingresso, per il numero specificato di posizioni. I valori spostati dal bit meno significativo sono scartati e i bit più significativi sono riempiti di zeri.

Stringhe di caratteri

Istruzione	Descrizione
ARRAY_TO_STRING	Converte i byte dell'array in ingresso in caratteri ASCII e li memorizza nella stringa in uscita.
CONCAT	Restituisce il risultato della concatenazione di due stringhe (appendendo una stringa alla fine dell'altra).
DELETE	Restituisce il risultato ottenuto cancellando dalla stringa in ingresso il numero di caratteri specificato, a partire dalla posizione specificata
DSPMSG	Visualizza un messaggio in una finestra aperta sullo schermo operatore.
Uguale (EQ)	Restituisce un valore booleano TRUE se gli ingressi sono uguali, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
FIND	Restituisce la posizione iniziale di una stringa in una seconda stringa. Se non trova la prima stringa restituisce 0.
Maggiore o uguale (GE)	Restituisce un valore booleano TRUE se il primo ingresso è maggiore o uguale al secondo, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
Maggiore di (GT)	Restituisce un valore booleano TRUE se il primo ingresso è maggiore del secondo, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
INT_TO_STRING	Converte un intero in una stringa di caratteri ASCII.
INSERT	Restituisce la stringa ottenuta inserendo una stringa di caratteri in un'altra, a partire dalla posizione specificata.
LEFT	Restituisce il numero specificato dei caratteri più a sinistra della stringa in ingresso.
Lunghezza (LEN)	Restituisce la lunghezza di una stringa di caratteri.
Minore o uguale (LE)	Restituisce un valore booleano TRUE se il primo ingresso è minore o uguale al secondo, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
Minore di (LT)	Restituisce un valore booleano TRUE se il primo ingresso è minore del secondo, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
MID	Restituisce il numero specificato di caratteri della stringa in ingresso, a partire dalla posizione specificata.
MSGWND	Visualizza un messaggio nella Output Window.
Non uguale (NE)	Restituisce un valore booleano TRUE se gli ingressi sono diversi, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
REAL_TO_STRING	Converte un numero reale in una stringa di caratteri ASCII.
REPLACE	Restituisce la stringa ottenuta sostituendo, a partire dalla posizione specificata, i caratteri di una stringa con il numero specificato di caratteri di un'altra stringa.
RIGHT	Restituisce il numero specificato dei caratteri più a destra della stringa in ingresso.

Conversione

Istruzione	Descrizione
BCD_TO_INT	Converte il valore BCD (Binary-Coded Decimal) in ingresso in un intero.
Da array di byte a stringa (BATOS)	Converte i byte dell'array in ingresso in caratteri ASCII e li memorizza nella stringa in uscita. Il byte finale deve contenere zero.
Da data a stringa (DateToString)	Converte un valore DATE in un valore STRING. Per puntare alla stringa è utilizzato un puntatore.
Da intero a stringa (ITOA)	Converte un intero nella rappresentazione come stringa di caratteri ASCII del suo valore e restituisce il risultato.
Da reale a stringa (RTOA)	Converte un numero reale nella rappresentazione come stringa di caratteri ASCII del suo valore e restituisce il risultato.
RGB to DWORD	Converte una tripletta di valori di rosso, verde e blu in un valore DWORD. E' utilizzata per i controlli ActiveX dell'interfaccia con l'operatore.
STRING_TO_ARRAY	Memorizza i caratteri della stringa ASCII in ingresso in un array di byte.
Da stringa ad array di byte (STOBA)	Converte ciascun carattere della stringa in ingresso nel suo valore decimale e memorizza i risultati in un array di byte. Nell'ultimo byte dell'array viene scritto uno zero.
Da stringa a data (StringToDate)	Converte un valore STRING in un valore DATE e restituisce il risultato.
Da stringa a intero (ATOI)	Converte una stringa numerica ASCII nell'intero ad essa equivalente.
Da stringa a reale (ATOR)	Converte una stringa numerica ASCII (con il punto decimale) nel numero reale ad essa equivalente.
Da stringa a ora del giorno (StringToTOD)	Converte un valore STRING in un valore TOD e restituisce il risultato.
Da ora del giorno a stringa (TODToString)	Converte un valore TOD in un valore STRING e pone il risultato nel simbolo puntato da pString.

Contatori e temporizzatori

Istruzione	Descrizione
Contatore a decremento (CTD)	Conta dal valore predefinito a zero. L'uscita (Q) diventa TRUE quando il conteggio arriva a zero. Può essere usato, ad esempio, per contare eventi ricorrenti.
Contatore a incremento (CTU)	Conta da zero al valore predefinito. L'uscita (Q) diventa TRUE quando il conteggio arriva al valore predefinito. Può essere usato, ad esempio, per contare eventi ricorrenti.
Contatore a incremento/decremento (CTUD)	Conta in ambedue i sensi, settando un'uscita quando il conteggio è maggiore o uguale al valore predefinito e un'altra uscita quando il conteggio è minore o uguale a zero.
Temporizzatore di ritardo alla disattivazione (TOF)	Disattiva un'uscita dopo il ritardo predefinito.
Temporizzatore di ritardo all'attivazione (TON)	Attiva un'uscita dopo il ritardo predefinito.
Temporizzatore a impulso (TP)	Il blocco funzionale TP temporizza la durata di un evento. Dopo che il suo ingresso ha ricevuto un impulso, è passato da off ad on, TP attende per l'intervallo di tempo predefinito e poi disattiva un'uscita (FALSE); questo fa del blocco funzionale TP un temporizzatore di ritardo alla disattivazione.

Rilevazione fronti

Istruzione	Descrizione
Attivazione al fronte di discesa (F_TRIG)	Il blocco funzionale F_TRIG mette un'uscita a TRUE per una scansione quando l'ingresso del blocco funzionale passa da TRUE a FALSE.
Attivazione al fronte di salita (R_TRIG)	Il blocco funzionale RTRIG mette a TRUE un'uscita per una scansione quando l'ingresso del blocco funzionale passa da FALSE a TRUE.

Temporizzatori estesi

Istruzione	Descrizione
Temporizzatore esteso di ritardo alla disattivazione (XTOF)	Disattiva un'uscita dopo il ritardo predefinito. E' come TOF, salvo che ha due ingressi in più (LoadTime e NewTime).
Temporizzatore esteso di ritardo all'attivazione (XTON)	Attiva un'uscita dopo il ritardo predefinito. E' come TON, salvo che ha due ingressi in più (LoadTime e NewTime).
Temporizzatore esteso a impulso (XTP)	Il blocco funzionale XTP temporizza la durata di un evento. Dopo che il suo ingresso ha ricevuto un impulso, è passato da off ad on, XTP attende per l'intervallo di tempo predefinito e poi disattiva un'uscita (FALSE); questo fa del blocco funzionale XTP un temporizzatore di ritardo alla disattivazione E' come TP, salvo che ha due ingressi in più (LoadTime e NewTime).

File

Istruzione	Descrizione
APPENDFILE	Scrive dati alla fine del file specificato dal blocco di controllo file.
BREAK	Causa la sospensione dell'esecuzione di un programma se è abilitato il debug.
CLOSEFILE	Chiude un file aperto dalla funzione OPENFILE.
COPYFILE	Copia un file esistente.
DELETEFILE	Cancella un file esistente.
NEWFILE	Crea un nuovo file.
OPENFILE	Apre un file per varie operazioni, come lettura e scrittura.
READFILE	Legge dati da un file.
REWINDFILE	Posiziona il puntatore interno all'inizio del file.
WRITEFILE	Scrive dati in un file.

Spostamento e rotazione di bit

Istruzione	Descrizione
ROL	Ruota verso sinistra i singoli bit di un valore per il numero specificato di posizioni.
ROR	Ruota verso destra i singoli bit di un valore per il numero specificato di posizioni.
SHL	Sposta verso sinistra i singoli bit di un valore per il numero specificato di posizioni.
SHR	Sposta verso destra i singoli bit di un valore per il numero specificato di posizioni.
IF	Permette di eseguire un gruppo di frasi solo quando una variabile booleana è TRUE.
INCLUDE	Chiama un file esterno ed esegue le frasi contenute in quel file.
REPEAT	Si usa per ripetere l'esecuzione di un gruppo di frasi finché una condizione booleana non diventa TRUE.
SCAN	Sospende l'esecuzione delle frasi ST mentre ha luogo una scansione degli I/O.
WHILE	Si usa per ripetere l'esecuzione di un gruppo di frasi finché una condizione booleana non diventa FALSE.

Matematica

Istruzione	Descrizione
Addizione (ADD)	Restituisce il risultato della somma degli ingressi.
ABS	Calcola il valore assoluto dell'ingresso.
Divisione (DIV)	Restituisce il risultato della divisione del primo ingresso per il secondo.
Potenza (EXPT)	Restituisce il risultato dell'elevazione del primo valore alla potenza specificata dal secondo valore.
MAX	Determina il massimo fra due valori.
MIN	Determina il minimo fra due valori.
Modulo (MOD)	Restituisce il resto della divisione del primo ingresso per il secondo.
MOVE	Restituisce il risultato della conversione del valore in ingresso nello stesso tipo dell'uscita.
Moltiplicazione (MUL)	Restituisce il risultato della moltiplicazione fra i valori in ingresso.
Negazione (NEG)	Restituisce il risultato del cambiamento di segno del valore in ingresso.
Radice quadrata (SQRT)	Restituisce la radice quadrata del valore in ingresso.
Sottrazione (SUB)	Restituisce il risultato della sottrazione del secondo valore in ingresso dal primo.

Miscellanee

Istruzione	Descrizione
Abort All	Mette fine a tutti i programmi che girano nel sottosistema di runtime.
Display Message	Visualizza un messaggio in una finestra messaggi di Windows. Durante la visualizzazione della finestra l'esecuzione del programma continua ininterrotta. L'operatore può cliccare su <i>OK</i> per cancellare il messaggio.
Initialize Array	Initializza tutti gli elementi di un array con il valore specificato.
Message Window	Visualizza un messaggio nella <i>Output Window</i> del Program Editor.
Change MMI Screen	Visualizza lo schermo HMI specificato.

Selezione

Istruzione	Descrizione
Massimo (MAX)	Determina il valore massimo degli ingressi e restituisce tale valore come uscita. Accetta fino a 16 ingressi.
Minimo (MIN)	Determina il valore minimo degli ingressi e restituisce tale valore come uscita. Accetta fino a 16 ingressi.

Oggetti di sistema

Istruzione	Descrizione
PID	Un PID è un'istruzione che permette la regolazione automatica in anello chiuso di anelli di controllo per processi continui. Per ciascun anello l'istruzione esegue il controllo proporzionale, e, facoltativamente, il controllo integrale, il controllo derivativo o entrambi:
PRGCB (blocco di controllo del programma)	Permette ad un programma applicativo SFC di compilare e di controllare l'esecuzione di altri programmi applicativi SFC o RLL.
TMR (temporizzatore)	Implementa un temporizzatore nel linguaggio ST.

Trigonometriche e Logaritmiche

Istruzione	Descrizione
Arcocoseno (ACOS)	Calcola l'arcocoseno dell'ingresso. Il risultato è in radianti.
Arcoseno (ASIN)	Calcola l'arcoseno dell'ingresso. Il risultato è in radianti.
Arcotangente (ATAN)	Calcola l'arcotangente dell'ingresso. Il risultato è in radianti.
Coseno (COS)	Calcola il coseno dell'ingresso, che deve essere in radianti.
Esponenziale (EXP)	Calcola l'esponenziale naturale dell'ingresso (eleva e alla potenza dell'ingresso).
EXPT	Eleva un valore alla potenza specificata da un secondo valore.
Logaritmo naturale (LN)	Calcola logaritmo naturale di un valore.
Logaritmo (LOG)	Calcola il logaritmo in base 10 di un valore.
Seno (SIN)	Calcola il seno dell'ingresso, che deve essere in radianti.
Tangente (TAN)	Calcola la tangente dell'ingresso, che deve essere in radianti.

Funzioni Motion Control

AXSJOG	Avvia un comando jog per l'asse specificato.
MOVEAXS	Avvia un comando di movimento per l'asse specificato.
STOPJOG	Ferma un comando jog per l'asse specificato.

Set di istruzioni del linguaggio Lista di istruzioni

Stringhe di bit

Istruzione	Descrizione
AND	Restituisce l'AND booleano o logico a livello di bit dei valori in ingresso.
NOT	Restituisce l'OR booleano o logico a livello di bit dei valori in ingresso.
OR esclusivo (XOR)	Restituisce l'OR esclusivo booleano o logico a livello di bit dei valori in ingresso.
Rotazione a sinistra (ROL)	Restituisce un valore calcolato spostando circolarmente verso sinistra i bit del valore in ingresso, per il numero specificato di posizioni. I valori spostati dal bit più significativo (MSB = Most Significant Bit) sono ruotati nel bit meno significativo (LSB = Least Significant Bit).
Rotazione a destra (ROR)	Restituisce un valore calcolato spostando circolarmente verso destra i bit del valore in ingresso, per il numero specificato di posizioni. I valori spostati dal bit meno significativo (LSB = Least Significant Bit) sono ruotati nel bit più significativo (MSB = Most Significant Bit).
Spostamento a sinistra (SHL)	Restituisce un valore calcolato spostando verso sinistra i bit del valore in ingresso, per il numero specificato di posizioni. I valori spostati dal bit più significativo sono scartati e i bit meno significativi sono riempiti di zeri.
Spostamento a destra (SHR)	Restituisce un valore calcolato spostando verso destra i bit del valore in ingresso, per il numero specificato di posizioni. I valori spostati dal bit meno significativo sono scartati e i bit più significativi sono riempiti di zeri.

Stringhe di caratteri

Istruzione	Descrizione
Concatenate	Restituisce il risultato della concatenazione di due stringhe (appendendo una stringa alla fine dell'altra).
Delete	Restituisce il risultato ottenuto cancellando dalla stringa in ingresso il numero di caratteri specificato, a partire dalla posizione specificata.
Find	Restituisce la posizione iniziale di una stringa in una seconda stringa. Se non trova la prima stringa restituisce 0.
Insert	Restituisce la stringa ottenuta inserendo una stringa di caratteri in un'altra, a partire dalla posizione specificata.
Left	Restituisce il numero specificato dei caratteri più a sinistra della stringa in ingresso.
Length	Restituisce la lunghezza di una stringa di caratteri.
Middle	Restituisce il numero specificato di caratteri della stringa in ingresso, a partire dalla posizione specificata.
Replace	Restituisce la stringa ottenuta sostituendo, a partire dalla posizione specificata, i caratteri di una stringa con il numero specificato di caratteri di un'altra stringa.
Right	Restituisce il numero specificato dei caratteri più a destra della stringa in ingresso.

Istruzione	Descrizione
Uguale (EQ)	Restituisce un valore booleano TRUE se gli ingressi sono uguali, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
Maggiore o uguale (GE)	Restituisce un valore booleano TRUE se il primo ingresso è maggiore o uguale al secondo, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
Maggiore di (GT)	Restituisce un valore booleano TRUE se il primo ingresso è maggiore del secondo, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
Minore o uguale (LE)	Restituisce un valore booleano TRUE se il primo ingresso è minore o uguale al secondo, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
Minore di (LT)	Restituisce un valore booleano TRUE se il primo ingresso è minore del secondo, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.
Non uguale (NE)	Restituisce un valore booleano TRUE se gli ingressi sono diversi, altrimenti restituisce FALSE. Setta di conseguenza l'uscita del rung RLL.

Conversione

Istruzione	Descrizione
Da array di byte a stringa (BATOS)	Converte i byte dell'array in ingresso in caratteri ASCII e li memorizza nella stringa in uscita. Il byte finale deve contenere zero.
Da data a stringa (DateToString)	Converte un valore DATE in un valore STRING. Per puntare alla stringa è utilizzato un puntatore.
Da intero a stringa (ITOA)	Converte un intero nella rappresentazione come stringa di caratteri ASCII del suo valore e restituisce il risultato.
Da reale a stringa (RTOA)	Converte un numero reale nella rappresentazione come stringa di caratteri ASCII del suo valore e restituisce il risultato.
RGB to DWORD	Converte una tripletta di valori di rosso, verde e blu in un valore DWORD. E' utilizzata per i controlli ActiveX dell'interfaccia con l'operatore.
Da stringa ad array di byte (STOBA)	Converte ciascun carattere della stringa in ingresso nel suo valore decimale e memorizza i risultati in un array di byte. Nell'ultimo byte dell'array viene scritto uno zero.
Da stringa a data (StringToDate)	Converte un valore STRING in un valore DATE e restituisce il risultato.
Da stringa a intero (ATOI)	Converte una stringa numerica ASCII nell'intero ad essa equivalente.
Da stringa a reale (ATOR)	Converte una stringa numerica ASCII (con il punto decimale) nel numero reale ad essa equivalente.
Da stringa a ora del giorno (StringToTOD)	Converte un valore STRING in un valore TOD e restituisce il risultato.
Da ora del giorno a stringa (TODToString)	Converte un valore TOD in un valore STRING e pone il risultato nel simbolo puntato da pString.

Matematica

Istruzione	Descrizione
Valore assoluto (ABS)	Calcola il valore assoluto dell'ingresso.
Addizione (ADD)	Restituisce il risultato della somma degli ingressi.
Divisione (DIV)	Restituisce il risultato della divisione del primo ingresso per il secondo.
Potenza (EXPT)	Restituisce il risultato dell'elevazione del primo valore alla potenza specificata dal secondo valore.
Modulo (MOD)	Restituisce il resto della divisione del primo ingresso per il secondo.
MOVE	Restituisce il risultato della conversione del valore in ingresso nello stesso tipo dell'uscita.
Moltiplicazione (MUL)	Restituisce il risultato della moltiplicazione fra i valori in ingresso.
Negazione (NEG)	Restituisce il risultato del cambiamento di segno del valore in ingresso.
Radice quadrata (SQRT)	Restituisce la radice quadrata del valore in ingresso.
Sottrazione (SUB)	Restituisce il risultato della sottrazione del secondo valore in ingresso dal primo.

Miscellanee

Istruzione	Descrizione
Abort All	Mette fine a tutti i programmi che girano nel sottosistema di runtime.
Change MMI Screen	Visualizza lo schermo HMI specificato.

Trigonometriche e Logaritmiche

Istruzione	Descrizione
Arcocoseno (ACOS)	Calcola l'arcocoseno dell'ingresso. Il risultato è in radianti.
Arcoseno (ASIN)	Calcola l'arcoseno dell'ingresso. Il risultato è in radianti.
Arcotangente (ATAN)	Calcola l'arcotangente dell'ingresso. Il risultato è in radianti.
Coseno (COS)	Calcola il coseno dell'ingresso, che deve essere in radianti.
Esponenziale (EXP)	Calcola l'esponenziale naturale dell'ingresso (eleva e alla potenza dell'ingresso).
Logaritmo naturale (LN)	Calcola il logaritmo naturale di un valore.
Logaritmo (LOG)	Calcola il logaritmo in base 10 di un valore.
Seno (SIN)	Calcola il seno dell'ingresso, che deve essere in radianti.
Tangente (TAN)	Calcola la tangente dell'ingresso, che deve essere in radianti.

.CFG

Un file con l'estensione .CFG è un file di dati che contiene informazioni sui parametri di configurazione dell'hardware del sistema e informazioni simboliche che si riferiscono ai componenti fisici e ai segnali I/O.

Anello

Elemento di programmazione SFC grazie al quale l'esecuzione del programma può continuare normalmente verso il basso o ritornare a un punto precedente del percorso di controllo. L'elemento anello contiene due condizioni di transizione, una per la continuazione verso il basso e una per il ritorno all'inizio dell'anello. E' possibile annidare uno nell'altro più anelli, ma essi non possono incrociarsi né entrare in una divergenza selettiva o in una divergenza parallela.

ANY

Tipo generico di dati che può rappresentare uno qualsiasi (any) dei tipi di dati supportati.

ANY_BIT

Tipo generico di dati che può rappresentare i tipi DWORD, WORD, BYTE, BOOL, incluso un singolo bit di uno di questi tipi di dati.

ANY_INT

Tipo generico di dati che può rappresentare i tipi INT, UINT o DINT.

ANY_NUM

Tipo generico di dati che può rappresentare i tipi ANY_REAL e ANY_INT.

ANY_REAL

Tipo generico di dati che può rappresentare il tipo REAL.

Azione

Collezione di operazioni dotata di un nome e associata ad uno o più passi in un SFC.

Blocco funzionale

Algoritmo RLL predefinito che esegue una singola operazione, estendendo le capacità del programma oltre la semplice logica implicita in contatti e bobine.

Bobina

Elemento grafico della programmazione RLL che rappresenta un simbolo booleano in uscita.

Bobina di transizione SFC

Elemento grafico di programmazione SFC che può essere associato a un passo o a un passo macro per controllare il flusso del programma. Associata ad un passo, la bobina di transizione SFC può mettere fine all'esecuzione del passo, trasferendo il flusso del programma e un altro passo. Associata ad un passo macro, questa bobina può mettere fine all'esecuzione dell'SFC figlio, trasferendo il flusso del programma ad un altro passo dell'SFC padre.

BOOL

Membro del gruppo di tipi di dati ANY_BIT. I dati di tipo BOOL sono validi nelle istruzioni e nei blocchi funzionali che accettano simboli di tipo ANY, ANY_BIT, o BOOL. Un BOOL è lungo un bit e può assumere uno di questi due valori: TRUE (1, o on) o FALSE (0, o off).

Booleano

Elemento logico o espressione che ha come risultato TRUE o FALSE.

BYTE

Membro del gruppo di tipi di dati ANY_BIT. I dati di tipo BYTE sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY, ANY_BIT, o BYTE. Un BYTE è un intero senza segno composto da una o più cifre (0-9). (Un dato di tipo BYTE non può contenere il punto decimale.) Un BYTE è lungo 8 bit e il campo dei suoi valori va da 0 a 255.

Condizione di transizione

Una condizione di transizione è un'espressione logica, che fornisce un risultato di tipo BOOL, associata a una transizione SFC. Il risultato booleano dell'espressione determina il momento in cui il passo attivo viene disattivato e viene attivato il passo successivo alla transizione.

Connettore

Gruppo di porte I/O facenti normalmente capo a un connettore fisico o a una scheda.

Contatto

Elemento grafico di programmazione RLL che rappresenta un simbolo booleano in ingresso.

Cursore

Il cursore è l'oggetto che segue il movimento del dispositivo di puntamento (tastiera, mouse, trackball o touch screen) e si usa per selezionare gli elementi dell'interfaccia operatore sullo schermo.

DATE

Membro del gruppo di tipi di dati ANY_DATE. I dati di tipo DATE sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY, ANY_DATE, o DATE.

DDE

Il DDE (Dynamic Data Exchange = scambio dinamico dei dati) è il passaggio di dati fra applicazioni diverse realizzato senza il coinvolgimento dell'operatore. In ambiente Windows, il DDE è realizzato mediante un gruppo di tipi di messaggi, un gruppo di procedure (protocolli) per il trattamento di questi messaggi e vari tipi specifici di dati. Seguendo i protocolli, applicazioni scritte indipendentemente l'una dall'altra possono scambiarsi i dati senza coinvolgimento da parte dell'operatore.

Diagramma Ladder

Il diagramma ladder è un linguaggio grafico di programmazione definito da IEC 1131, che usa i concetti della logica a relè (RLL).

Diagramma funzionale sequenziale

Il Diagramma funzionale sequenziale (Sequential Function Chart) o SFC è un linguaggio grafico che permette di schematizzare la logica sequenziale, ricorrendo a passi, transizioni e azioni.

DINT

Membro del gruppo di tipi di dati ANY_NUM. I dati di tipo DINT sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY, ANY_NUM, ANY_INT, o DINT. Un DINT è un intero con segno composto da una o più cifre (0-9). (Un dato di tipo DINT non può contenere il punto decimale.) Un DINT è lungo 32 bit e il suo campo di valori va da -2147483648 a +2147483647.

Divergenza parallela

Elemento di un programma SFC che suddivide il percorso di controllo in due o più rami paralleli. Quando l'esecuzione del programma raggiunge una divergenza parallela, tutti i percorsi di controllo successivi diventano attivi in parallelo. I percorsi interni alla divergenza parallela continuano ad essere attivi finché non raggiungono tutti il punto di convergenza. A questo punto tutti i percorsi della divergenza parallela sono disattivati e diventa attivo il percorso sottostante al punto di convergenza.

Divergenza selettiva

Elemento grafico della programmazione SFC che suddivide un percorso di controllo in due o più rami. Il ramo o percorso di controllo selezionato per l'esecuzione è determinato dalle condizioni di transizione posizionate all'inizio di ciascun ramo.

Divergenza simultanea

Elemento di un programma SFC che suddivide il percorso di controllo in due o più rami paralleli. Quando l'esecuzione del programma raggiunge una divergenza simultanea, tutti i percorsi di controllo successivi diventano attivi in parallelo. I percorsi interni alla divergenza simultanea continuano ad essere attivi finché non raggiungono tutti il punto di convergenza. A questo punto tutti i percorsi della divergenza simultanea sono disattivati e diventa attivo il percorso sottostante al punto di convergenza.

DWORD

Membro del gruppo di tipi di dati ANY_BIT. I dati di tipo DWORD sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY, ANY_BIT, o DWORD. Un è un intero senza segno composto da una o più cifre (0-9). (Un dato di tipo DWORD non può contenere il punto decimale.) Un DWORD è lungo 32 bit e il suo campo di valori va da 0 a 4,294,967,295.

Etichetta

Elemento di programmazione grafica SFC o RLL, che identifica il punto dal quale deve riprendere il flusso del programma dopo il salto corrispondente.

FCB

File Control Block (blocco controllo file). Un dato interno usato per controllare le operazioni di trattamento dei file nei programmi ST.

File attivo

File di programma contenuto nella finestra in primo piano del Program Editor e il cui titolo è evidenziato. I comandi di menu e i pulsanti della barra degli strumenti agiscono sul file attivo.

File di configurazione

Un file di configurazione contiene dati e parametri che descrivono la configurazione hardware del sistema. Questo file contiene anche gli identificatori simbolici usati nei programmi applicativi per fare riferimento alle unità hardware fisiche e ai segnali I/O.

Funzione

Algoritmo SFC predefinito che esegue una singola operazione, come radice quadrata, rotazione a sinistra, riavvolgimento file, ecc.

HMI

HMI sta per "Human/Machine Interface" e si riferisce all'insieme di funzioni che permettono all'operatore di interagire con la macchina o il controllore del processo.

I/O

I/O (Input/Output) sta per ingressi e uscite del controllore.

I/O Scanner

Lo I/O scanner è il sottosistema che acquisisce tutti gli ingressi e trasmette tutte le uscite all'interfaccia I/O. Lo I/O scanner aggiorna le tabelle interne di I/O usate da altri sottosistemi.

Icona

Un'icona è una figura che rappresenta un altro oggetto. In SFC+ può essere usata per rappresentare passi o passi macro. Le icone possono contenere funzionalità predefinite e si possono usare per creare librerie di funzioni o routine fisse.

Icona applicativa

Un'icona applicativa è un modello di passo al quale è associata un'icona. Quando in un programma SFC viene inserita un'icona applicativa, viene creato un nuovo passo nel quale sono copiate le informazioni contenute in tale icona.

IEC 1131-3

Lo IEC 1131-3 è uno standard internazionale della International Electrotechnical Commission che definisce i linguaggi di programmazione per controllori programmabili (Norma italiana: CEI EN 61131-3).

Instruction List

Instruction List (Lista di istruzioni) è un linguaggio testuale definito dallo standard IEC 1131-3.

INT

Membro del gruppo di tipi di dati ANY_NUM. I dati di tipo INT sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY, ANY_NUM, ANY_INT, o INT. Un INT è un intero con segno composto da una o più cifre (0-9). (Un dato di tipo INT non può contenere il punto decimale.) Migliorando rispetto alle specifiche IEC 1131-3, un INT è lungo 32 bit e il suo campo di valori va da -2147483648 a +2147483647.

Interfaccia operatore

L'interfaccia operatore è un insieme di schermi, messaggi e finestre che permettono all'operatore di inviare comandi a una macchina o a un processo e di monitorarne il funzionamento.

Intero

Il tipo di dati intero, o INT, è composto da 32 bit, ha il segno e può rappresentare valori compresi tra -2,147,483,648 e +2,147,483,648.

Lista di Istruzioni

Lista di istruzioni (Instruction List o IL) è un linguaggio testuale definito dallo standard IEC 1131-3.

Logica della transizione

La logica della transizione può essere rappresentata da una singola espressione booleana o da un singolo rung RLL.

Motore di runtime

E' il modulo responsabile della schedulazione e dell'esecuzione della logica associata al codice sorgente del progetto (SFC, RLL, ecc.). Questo modulo funziona anche come server per i client DDE.

Nome di passo

Il nome di passo è un identificatore che individua un passo in un diagramma funzionale sequenziale. Il nome attribuito ad un passo può essere cambiato facendo doppio click sul rettangolo che rappresenta il passo, per aprire la finestra di dialogo Edit Step, facendo doppio click sull'identificatore che compare nel campo Step Name e inserendo un nuovo identificatore.

Passo

Elemento grafico con un nome che, in un SFC, rappresenta uno stato, o un periodo di tempo, nel corso dell'esecuzione del programma durante il quale sono eseguite le azioni e le funzioni associate al passo.

Passo macro

Elemento grafico con un nome in un SFC, che rappresenta l'inclusione di un altro SFC completo in un singolo passo. L'esecuzione dell'SFC incluso inizia dal suo passo iniziale (Start) quando il macro passo che lo richiama diventa attivo. L'esecuzione del passo macro è completata quando l'SFC incluso raggiunge il suo passo finale (End). I passi macro possono essere usati per controllare la complessità o per presentare una vista di più alto livello di un SFC molto grande, frazionandolo in una collezione di SFC più piccoli e più semplici. A un passo macro possono essere associate azioni, esattamente come ai passi normali.

Percorso di controllo

Il percorso di controllo è una linea fra i passi di un SFC indicante il percorso del flusso del programma.

Porta

Una porta I/O è normalmente costituita da otto bit. Un programma può accedere a una porta come se fosse un intero.

Posizione del carattere

Posizione di un carattere in una stringa. La posizione del primo carattere di una stringa è 1.

PRGCB

Il Program Control Block permette a un programma SFC di creare e di controllare l'esecuzione di altri programmi applicativi.

Prima scansione

La logica di prima scansione (First Scan Logic) serve per eseguire determinati comandi solo una volta, durante la prima scansione logica del programma. Un esempio può essere fornito dall'inizializzazione di contatori e temporizzatori.

Progetto

Organizza il gruppo di programmi applicativi e di file di configurazione di un'applicazione in una subdirectory separata.

Program Editor

Il Program Editor è un'utilità usata per editare i programmi applicativi. Include gli editor per i programmi SFC e RLL.

Qualificatore di azione

Elemento della programmazione SFC associato a ciascuna azione che controlla l'esecuzione della logica dell'azione nel periodo durante il quale il passo cui è associata l'azione è attivo.

REAL

Membro del gruppo di tipi di dati ANY_NUM. I dati di tipo REAL sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY, ANY_NUM, ANY_REAL, o REAL. Un dato REAL è composto da una o più cifre (0-9), ha il segno, e contiene il punto decimale. Il campo dei valori dei dati REAL va da -3.402823 E38 a -1.401298 E-45 (negativi), e da +1.401298E-45 a +3.402823 E38 (positivi).

Relay Ladder Logic

RLL è il linguaggio grafico di programmazione usato per descrivere la logica del programma applicativo mediante contatti e bobine.

RLL

RLL sta per Relay Ladder Logic. Un linguaggio grafico di programmazione che utilizza contatti e bobine.

Rung

Elemento della programmazione RLL che rappresenta una funzione RLL.

Scheda

Vedere "Scheda I/O".

Scheda I/O

Si inserisce in uno degli slot di espansione dell'unità di sistema ed è usata per la connessione con i rack o i moduli di I/O periferici. E' detta anche semplicemente scheda.

Sequential Function Chart

Il Sequential Function Chart (Diagramma funzionale sequenziale) o SFC è un linguaggio grafico che permette di schematizzare la logica sequenziale, ricorrendo a passi, transizioni e azioni.

SFC

SFC sta per Sequential Function Chart (Diagramma funzionale sequenziale), un linguaggio grafico che permette di schematizzare la logica sequenziale, ricorrendo a passi, transizioni e azioni. Lo standard IEC 1131 ha definito una specifica implementazione di SFC per la programmazione di PLC.

SFC+

SFC+ è una versione ampliata del linguaggio di programmazione Sequential Function Chart definito nello standard IEC 1131, usata dal software PC Control.

Simbolo

Posizione della memoria interna contenente informazioni. La natura delle informazioni è definita dal tipo di dato e in base a questo il contenuto del simbolo può essere un numero reale, un intero una stringa di caratteri, ecc. Per definire un simbolo assegnandogli un nome e un tipo di dato si usa il Symbol Manager.

Slot

Uno spazio in un I/O o in un computer nel quale può essere inserita una scheda.

STRING

Membro del gruppo di tipi di dati ANY. I dati di tipo STRING sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY o STRING. Il formato di un dato di tipo STRING è una stringa di un massimo di 250 caratteri ASCII tra virgolette singole.

Structured Text

Lo Structured Text (Testo strutturato) o ST è un linguaggio testuale di programmazione definito nello standard IEC 1131.

Strumento anello

Lo strumento anello è un pulsante della barra degli strumenti SFC, che inserisce un elemento di controllo nel diagramma SFC.

Strumento contatto

Uno strumento della barra degli strumenti RLL usato per inserire un contatto in un programma RLL.

Testo strutturato

Il Testo strutturato (Structured Text o ST) è un linguaggio testuale di programmazione definito nello standard IEC 1131.

TIME

Membro del gruppo di tipi di dati ANY. I dati di tipo TIME sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY o TIME. Il formato di un dato di tipo TIME consiste nella stringa T#, o t#, seguita da uno o più numeri e specificatori delle unità di tempo. Esempi:

T#1D2h = 1 giorno e 2 ore

t#26H = 26 ore

t#5m45s = 5 minuti e 45 secondi

t#26S200MS = 26 secondi e 200 millisecondi

T#900ms = 900 millisecondi

Time of Day

Il tipo di dati Time of Day (ora del giorno), o TOD, si usa per rappresentare una specifica ora del giorno. Per i dati di tipo TOD, lo standard IEC 1131-3 definisce il formato TOD#HH:MM:SS.ms. Esempio: TOD#23:59:59.999.

TMR

Il tipo di dati timer, o TMR si usa per implementare un temporizzatore in linguaggio strutturato.

TOD

Membro del gruppo di tipi di dati ANY_DATE. I dati di tipo TOD (Time of Day = Ora del giorno) sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY, ANY_DATE, o TOD.

Transition Mode

Il pulsante Transition Mode della barra degli strumenti SFC determina il tipo della transizione inserita con lo strumento Inserimento transizione. Se il pulsante Transition Mode non è premuto, viene inserita una transizione RLL. Se questo pulsante è premuto, viene inserita una transizione booleana.

Transizione

Elemento grafico del linguaggio SFC, che fornisce un risultato booleano. Questo risultato determina il passaggio del flusso del programma dal passo o dai passi precedenti al passo o ai passi successivi alla transizione.

Transizione booleana

Transizione SFC rappresentata da un'espressione costituita da simboli e operandi il cui risultato è di tipo BOOL. La condizione di transizione è soddisfatta quando il risultato dell'espressione è TRUE.

Transizione RLL

Transizione programmata in un SFC usando la logica a contatti. La transizione RLL è costituita da un singolo rung con una bobina di uscita avente lo stesso nome della transizione. La condizione di transizione è soddisfatta quando questa bobina riceve corrente.

UINT

Membro del gruppo di tipi di dati ANY_INT. I dati di tipo UINT sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY, ANY_INT, o UINT. Un UINT è un intero senza segno composto da una o più cifre (0-9). (Un dato di tipo UINT non può contenere il punto decimale.) Un UINT è lungo 16 bit e il suo campo di valori va da 0 a 65535.

WORD

Membro del gruppo di tipi di dati ANY_BIT. I dati di tipo WORD sono validi nelle istruzioni e nei blocchi funzionali che accettano dati di tipo ANY, ANY_BIT, o WORD. Un WORD è un intero senza segno composto da una o più cifre (0-9). (Un dato di tipo WORD non può contenere il punto decimale.) Un WORD è lungo 16 bit e il suo campo di valori va da 0 a 65535.

A

Aggiornamento della versione precedente, 1-4

Aggiunta dei controlli di flusso del programma SFC, 4-71

Aggiunta della logica delle transizioni, D-11

Aggiunta di codice Motion Control a un SFC, E-3

Aggiunta di controlli, 6-9

Aggiunta ed editazione dei blocchi delle azioni, D-14

Aiuto in linea, 1-7

Allineamento di controlli, 6-12

Apertura del Symbol Manager, 4-12

Apertura di un documento IL, 4-105

Apertura di un documento in testo strutturato, 4-82

Apertura di un file interfaccia operatore, 6-6

Apertura di un progetto, 2-20

Apertura di un programma, 2-22

Array, 4-12

Attivazione di configurazioni, 6-13

Attivazione di un file di configurazione, 3-4

Attivazione di una configurazione, 2-21

Attivazione di una configurazione, 2-32

Attivazione/disattivazione della visualizzazione dei commenti, 2-25

Avvio dei programmi con un file batch, 5-4

Avvio del software Operator Interface, 6-2

Avvio di programmi con il comando Run With Restart, 5-5

B

Bar, 6-16

Barre degli strumenti e barra di stato, 2-16

Bitmap, 6-18

BOOL (Booleano), 4-7

Box, 6-19

BYTE, 4-7

C

Cancellazione dell'esecuzione di un programma, 5-3

Cancellazione di controlli, 6-12

Cancellazione di un simbolo, 4-16

Cancellazione di uno schermo, 6-7

Caratteristiche elettriche del PCIM , A-15

Che cosa è l'editazione in linea, 4-115

Che cosa sono i passi, 4-41

Che cosa sono i simboli, 4-2

Che cosa sono le azioni, 4-44

Che cosa sono le divergenze, 4-51

Che cosa sono le transizioni, 4-50

Chiusura dei programmi applicativi, 2-24

Click Button, 6-20

Come iniziare, 2-1

Come sono valutate le transizioni, 4-57

Come vengono risolti gli SFC, 4-56

Come vengono risolti i programmi applicativi RLL, 4-27

Come viene risolta la logica RLL quando si usano i blocchi funzionali, 4-28

Indice alfabetico

- Come viene risolta la semplice logica a relè, 4-27
- Commenti, 4-88
- Commutazione dei modi del software
Operator Interface, 6-4
- Compatibilità, C-2
- Comunicazioni DDE con il software Microsoft Excel, 3-38
- Configurazione degli I/O, 3-1
- Configurazione degli I/O della Serie 90-30, 3-16
- Configurazione degli I/O DeviceNet, 3-20
- Configurazione degli I/O Profibus, 3-22
- Configurazione dei programmi per l'esecuzione automatica, 5-4
- Configurazione dei simboli, 4-2
- Configurazione dell'UPS, 2-30
- Configurazione della scheda PCIF-30, B-4
- Configurazione delle opzioni di Motion Control, E-18
- Configurazione di altri bus di campo, 3-20
- Configurazione di GENIUS I/O, 3-7
- Configurazione per Motion Control, E-2
- Conflitti hardware, 3-2
- Connessione del PCIM al bus, A-12
- Connettori, C-4
- Continuous Button, 6-23
- Controlli ActiveX, 6-33
- Controlli ActiveX e controlli standard, 6-34
- Controlli Motion dell'interfaccia con l'operatore, E-31
- Controlli standard, 6-14
- Controllo del flusso dei programmi applicativi RLL e ST, 4-100
- Controllo del flusso del programma in applicazioni Motion, E-19
- Controllo di programmi RLL mediante l'interfaccia con l'operatore, 6-5
- Copia, taglia e incolla, 6-11
- Copia di un progetto, 2-21
- Copia di un simbolo, 4-16
- Copia di uno schermo, 6-7
- Costanti letterali, 4-4
- Costanti letterali che rappresentano ora e data, 4-6
- Costanti letterali di durata, 4-5
- Costanti letterali numeriche, 4-4
- Creazione dei simboli, D-9
- Creazione dello schermo di interfaccia con l'operatore, D-20
- Creazione di applicazioni di interfaccia con l'operatore, 6-1, 6-14
- Creazione di diagrammi funzionali sequenziali, 4-60
- Creazione di programmi applicativi, 4-1
- Creazione di un nuovo file interfaccia operatore, 6-6
- Creazione di un nuovo progetto, 2-20
- Creazione di un nuovo progetto, D-1
- Creazione di un nuovo programma, 2-22
- Creazione di un nuovo schermo di interfaccia con l'operatore, 6-7
- Creazione di un programma RLL, 4-28
- Creazione di un programma SFC, 4-60

Creazione di un SFC, D-5
Creazione di un simbolo,
4-14
Creazione di una nuova
configurazione, 3-4
Creazione di una nuova
configurazione del
sistema, D-2

D

DATE, 4-8
Definizione degli indirizzi
sul bus Genius, 3-11
Definizione dei simboli di
tipo puntatore, F-3
Definizione delle
connessioni delle
porte analogiche,
3-19
Definizione delle
connessioni delle
porte digitali, 3-19
Denominazione di un bit
in un simbolo, 4-16
Descrizione, A-1
Descrizione del
linguaggio, 4-84
Descrizione del
linguaggio, 4-107
Descrizione del Software,
2-2
Descrizione dell'hardware,
C-2
Descrizione generale dei
diagrammi
funzionali
sequenziali, 4-41
Descrizione generale dei
diagrammi RLL
(Relay Ladder
Logic), 4-26
Descrizione generale del
funzionamento del
programma, 2-32
Descrizione generale
dell'Operator
Interface Editor, 6-2
Dimensionamento di
controlli, 6-11
Dimensionamento in scala
di un programma per
adeguarlo alla
finestra, 2-25
DINT (doppio intero), 4-8

Documentazione di un
programma SFC,
4-81
DWORD (doppia parola),
4-8

E

Editazione dei tipi di dati
definiti dall'utente,
4-17
Editazione del testo
strutturato, 4-83
Editazione delle istruzioni,
4-106
Editazione di controlli, 6-9
Editazione di controlli
ActiveX, 6-37
Editazione di simboli, 6-13
Editazione di testo
strutturato in un
passo SFC, 4-82
Editazione di un file di
configurazione
esistente, 3-4
Editazione di un simbolo,
4-14
Editazione in linea, 4-115
Elementi di una
configurazione, 3-4
Enumerazioni, 4-19
Esecuzione dei programmi
applicativi, 5-1
Esecuzione del programma
attivo, 5-2
Esecuzione del programma
dell'esempio, D-19
Esecuzione di un singolo
programma, 5-2
Esecuzione e
autorizzazione del
software PC Control,
1-5
Esempi di applicazioni,
D-1
Esempi di
programmazione,
4-112
Esempi di uso degli
operatori, 4-113
Esempio di richiamo di
blocco funzionale,
4-114

Indice alfabetico

Esportazione di simboli
per CIMPLICITY
HMI, 4-25
Espressioni, 4-84
Estensioni dello standard
IEC 1131-3, 4-59
ETICHETTA, 4-93

F

Finestra dei blocchi
RS274, E-34
Finestra di dialogo
Discrete Point
Information, 3-13
Finestra di dialogo Genius
Bus Address
Definitions, 3-12
Finestra di dialogo Global
Data Setup, 3-11
Finestra di dialogo Global
Data: Device, 3-11
Finestra di dialogo Global
Data: Device, Word,
3-11
Finestra di dialogo
Module, 3-18
Finestra di dialogo PCIF
Board Definition,
3-16
Finestra di dialogo Rack
Definition, 3-18
Fornitori di controlli
ActiveX, 6-34
Frase BREAK, 4-87
Frase CASE, 4-87
Frase di assegnazione,
4-86
Frase EXIT, 4-89
Frase FOR, 4-91
Frase IF, 4-90
Frase INCLUDE, 4-91
Frase REPEAT, 4-94
Frase SCAN, 4-95
Frase WHILE, 4-95
Funzionamento normale,
2-33
Funzioni del software
Operator Interface,
6-6
Funzioni e blocchi
funzionali, 4-110
Funzioni Motion Control
del linguaggio Testo
strutturato, E-29

G

Gauge, 6-25
Gestione dei programmi
applicativi, 2-24
Gestione progetti, 2-20
Gestore delle azioni, 4-50
Glossario dei termini, H-1
Guida all'avviamento
rapido, C-7

I

I file di configurazione, 3-4
I/O, 4-116
Identificatori, 4-3
Il modulo interfaccia PC
(PCIM = PC
Interface Module),
3-7
Importazione/esportazione
di configurazioni,
3-41
Impostazione delle opzioni
del sistema, 2-26
Inclusione di Testo
strutturato nel codice
Motion Control,
E-27
Indicator, 6-27
Indirizzamento, F-2
Indirizzo di memoria, 3-3
Inserimento di controlli
ActiveX, 6-36
Installazione, 1-2
Installazione del software
PC Control, 1-4
Installazione della scheda
PCIF-30, B-1
Installazione e
configurazione, A-3
Installazione e
configurazione delle
unità sulla rete
DeviceNet, 3-20
INT (intero), 4-8
Integrazione di testo
strutturato in un
SFC, 4-80
Interfaccia con personal
computer per I/O
della Serie 90-30
(PCIF), B-1

Interfaccia con personal computer per I/O della Serie 90-30 (PCIF2), C-1
Interrupt – IRQ, 3-3
Intervallo di scansione degli I/O, 3-5
Introduzione dei comandi in testo strutturato, D-13
Introduzione del codice di accesso, 6-3
Introduzione delle istruzioni, 4-83
Introduzione delle istruzioni, 4-105

L

L'interfaccia DDE, 3-38
L'interfaccia PC (PCIF = PC Interface), 3-16
La finestra di dialogo Define Board, 3-6
La finestra di dialogo System Configuration, 3-5
Limitazioni dei controlli ActiveX, 6-33
Linguaggi di programmazione IEC 1131-3, 2-3
Livelli di accesso, 2-7
Livelli di accesso, 6-2
Localizzazione dei simboli secondo IEC, 2-29

M

Menu, 2-10
Mezzi di controllo del flusso del programma, 4-53
Microinterruttori, C-6
Miglioramenti del software PC CONTROL allo standard RS-274D, E-3
Modi attivazione e edit, 6-4
Modulo di interfaccia con Personal Computer per Genius I/O, A-1

Monitoraggio del flusso di corrente, 5-5
Monitoraggio e prova dei programmi applicativi e dei simboli, 4-102
Monitoraggio ed esecuzione di programmi applicativi Motion Control, E-24
Motion Control, E-1

N

Navigazione all'interno dell'utilità di configurazione, 3-5
Nomi dei file, 2-34
Note sui puntatori, F-3
Numeric Display, 6-29

O

Operatori, 4-84
Operatori, 4-108
Operatori del linguaggio ST, 4-97
Operatori per i puntatori, 4-85
Operatori riservati ai puntatori, F-2
Operazioni relative agli schermi, 6-7
Operazioni relative ai controlli, 6-9
Operazioni su file, 4-117
Operazioni sui passi, 4-63
Operazioni sui passi macro, 4-67
Operazioni sui simboli, 6-13
Operazioni sulle azioni, 4-68
Operazioni sulle transizioni, 4-66

P

Pannello di stato monoasse, E-32
Pannello di stato multiasse, E-33
Pannello jog, E-31

Indice alfabetico

Pannello Selected Program
Status, 6-30
Parole riservate, 4-22
Partenza veloce, 2-4
Ponticelli, C-3
Porte I/O, 3-3
Prestazioni, 3-20
Prima scansione con la
configurazione
attiva, 2-32
Progetti e applicazioni,
2-20
Program Editor, 2-7
Programmazione con il
linguaggio IL (Lista
di istruzioni), 4-105
Programmazione con il
linguaggio ST (Testo
strutturato), 4-82
Programmazione Motion
Control, E-3
Programmazione RLL,
4-26
Programmazione SFC,
4-41
Programmi applicativi,
2-22
Programmi IL, 4-118
Programmi RLL, 4-117
Programmi RLL attivi, 5-5
Programmi SFC, 4-117
Programmi SFC attivi, 5-5
Programmi ST, 4-118
Puntatori, 4-12
Puntatori ad array, F-5

R

REAL, 4-9
Registrazione di controlli
ActiveX, 6-35
Regola generale, 4-116
Regole, 4-116
Ricerca guasti, A-16
Richiamo di funzione, 4-93
Rimozione del PCIM dal
Bus, A-13
Rinominazione di un
progetto, 2-21
Rinominazione di uno
schermo, 6-8
Risposte alle domande più
frequenti, 2-18

S

Salvataggio di un file, 2-23
Salvataggio di un file
interfaccia operatore,
6-6
Scambio dinamico dei dati
(DDE = Dynamic
Data Exchange),
3-38
Schermi dell'interfaccia
con l'operatore, 6-6
Selezione dello schermo da
editare, 6-8
Selezione dello schermo
iniziale, 6-8
Selezione di controlli, 6-10
Sequenza di spegnimento,
2-32
Set di istruzioni del
linguaggio Lista di
istruzioni, G-17
Set di istruzioni del
linguaggio RLL, G-1
Set di istruzioni del
linguaggio SFC, G-7
Set di istruzioni del
linguaggio Testo
strutturato, G-10
Simboli, 4-116
Simboli di Run-Time, 4-21
Simboli di sistema, 4-20
Simboli di sistema
predefiniti, 4-20
Simboli Motion Control
predefiniti, E-11
Sintassi del testo
strutturato, 4-85
Sintassi IL, 4-107
Slide, 6-30
Sommario dei tipi di frasi,
4-98
Sommario del set di
istruzioni, G-1
Sottosistemi di runtime,
5-2
Sottosistemi di runtime del
software PC Control,
2-6
Sottosistemi software, 2-2
Specifiche, A-14
Spostamento dei controlli
davanti/dietro, 6-12

Spostamento di controlli,
6-10
Stampa della Cross-
Reference del
programma, 2-24
Stampa di un programma,
2-23
Standard IEC 1131, 2-3
STRING, 4-9
Stringhe di caratteri
letterali, 4-4
Symbol Manager, 4-12

T

Tasti di scelta rapida
dell'interfaccia
operatore, 2-9
Text, 6-32
TIME, 4-9
Tipi di dati, 4-7
Tipi di dati definiti
dall'utente, 4-11
Tipi di dati generici, 4-11
TOD (TIME_OF_DAY),
4-10
Trascinamento, 4-19
Trasferimento di dati ad
Excel, 3-38
Trasferimento di dati al
sistema di comando,
3-39
Trasferimento di valori al
sistema di comando
su richiesta, 3-39
Trasmissione dei
datagrammi con il
driver PCIM, 3-15

U

UINT (Intero senza segno),
4-10
Uso dei passi macro, 4-58
Uso dei puntatori nel
linguaggio ST, F-1
Uso dei simboli, 4-19
Uso dei simboli di sistema
del passo, 4-43
Uso del richiamo macro
G65 con Motion
Control, E-22

Uso della barra del menu e
degli strumenti SFC,
4-62
Uso della documentazione
in linea, 1-7
Uso delle divergenze
simultanee, 4-58
Uso delle frasi Motion
Control, E-4
Uso di Pid in un
programma
applicativo, 4-98

V

Variazione dei codici di
accesso, 6-3
Visibilità dei simboli, 4-2
Visualizzazione dei
programmi, 2-25
Visualizzazione dello stato
dei programmi
applicativi, 5-6
WORD, 4-10